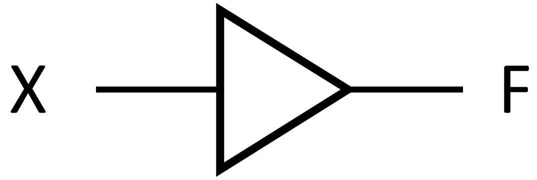
Digital Logic Design

Lecture 13

Buffer

- A buffer is a gate with the function F = X:



- In terms of Boolean function, a buffer is the same as a connection!
- So why use it?
  - A buffer is an electronic amplifier used to improve circuit voltage levels and increase the speed of circuit operation.

Design a circuit that accepts 3-bit number and generates a 6 bit binary number equal to square of the given number

A	B	C	S5	S4	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

K-map for S0:

A \ BC			
0	**1**	**1**	**0**
0	**1**	**1**	**0**

S0 = C

K-map for S2:

A \ BC

0	0	0	1
0	0	0	1

S2 = BC’

K-map for S3:

A \ BC

0	0	1	0
0	1	0	0

S3 = AB’C + A’BC

K-map for S4:

A \ BC

0	0	0	0
1	1	1	0

S4 = AB’ + AC

K-map for S5:

A \ BC

0	0	0	0
0	0	1	1

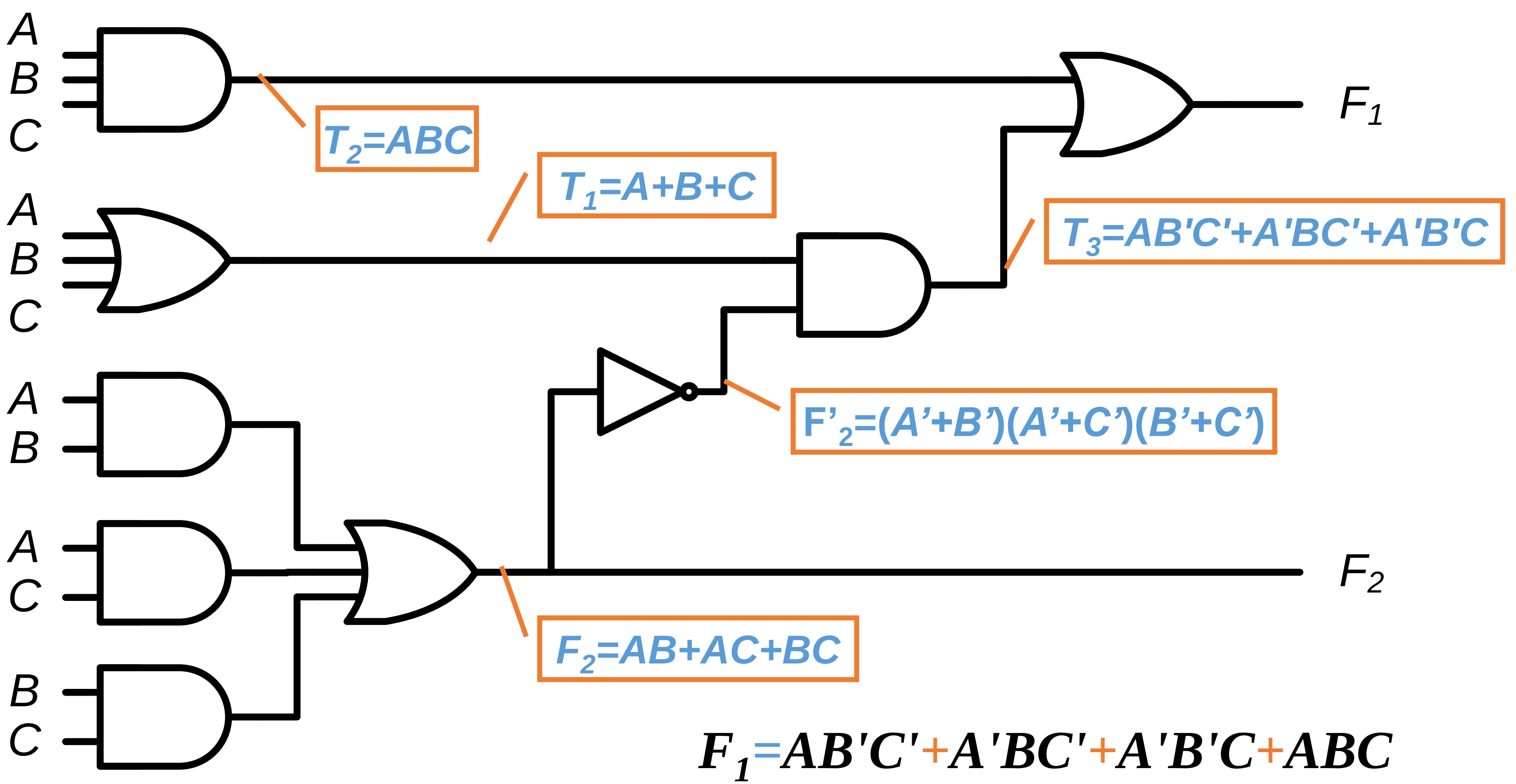
S5 = AB

Analysis Procedure

- The design of a combinational circuit starts from the verbal specifications of a required function and ends with a set of output Boolean functions or a logic diagram.
- The analysis of a combinational circuit is somewhat the reverse process.
- It starts with a given logic diagram and ends with a set of Boolean functions, a truth table or a verbal explanation of the circuit operation.

Analysis Procedure

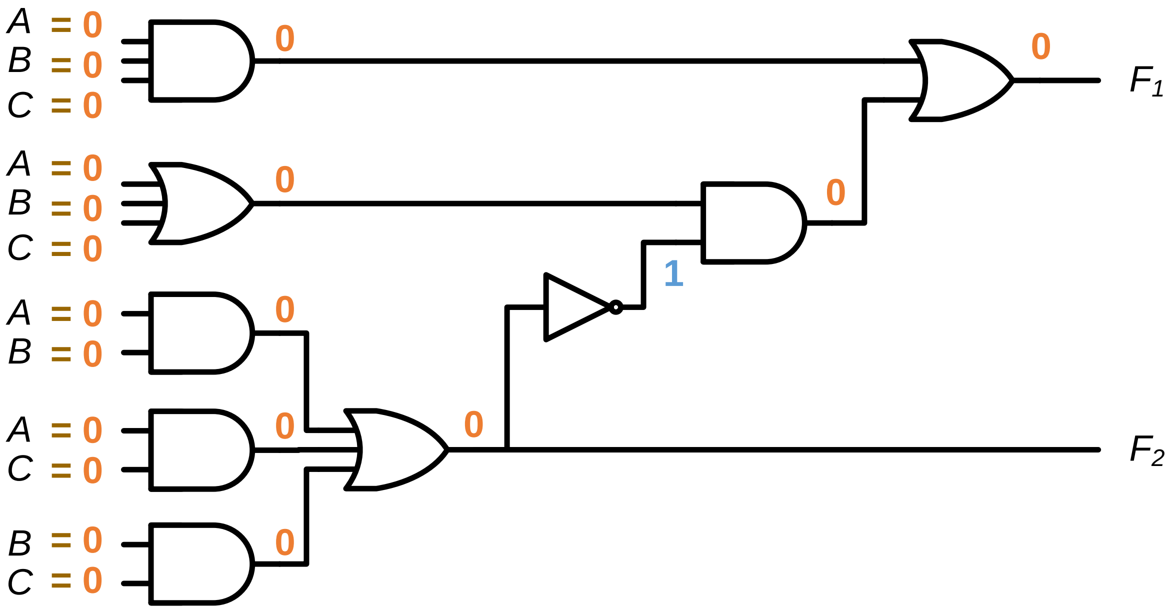
- **Boolean Expression Approach**



$$F_1=AB'C'+A'BC'+A'B'C+ABC$$
$$F_2=AB+AC+BC$$

Analysis Procedure

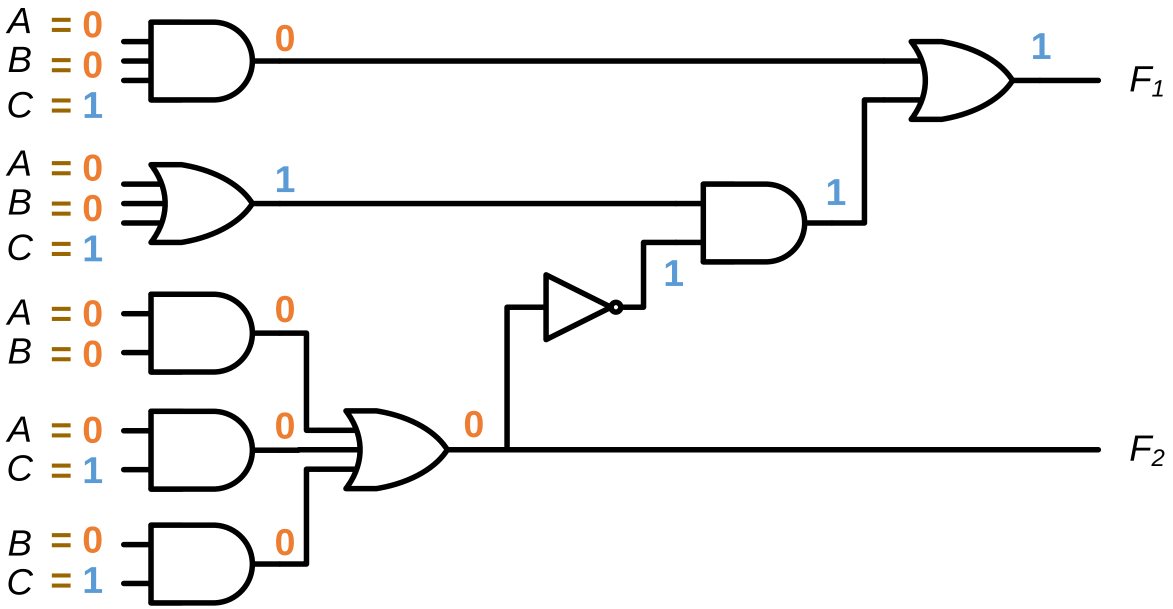
- **Truth Table Approach**



A B C	F_1	F_2
0 0 0	0	0

Analysis Procedure

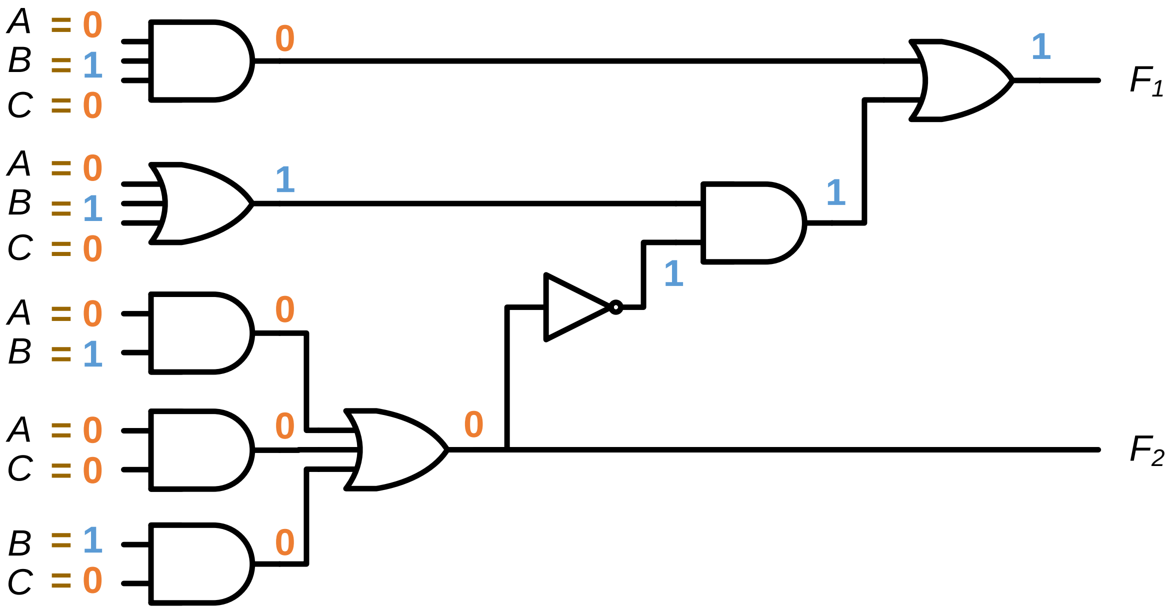
- **Truth Table Approach**



A B C	F_1	F_2
0 0 0	0	0
0 0 1	1	0

Analysis Procedure

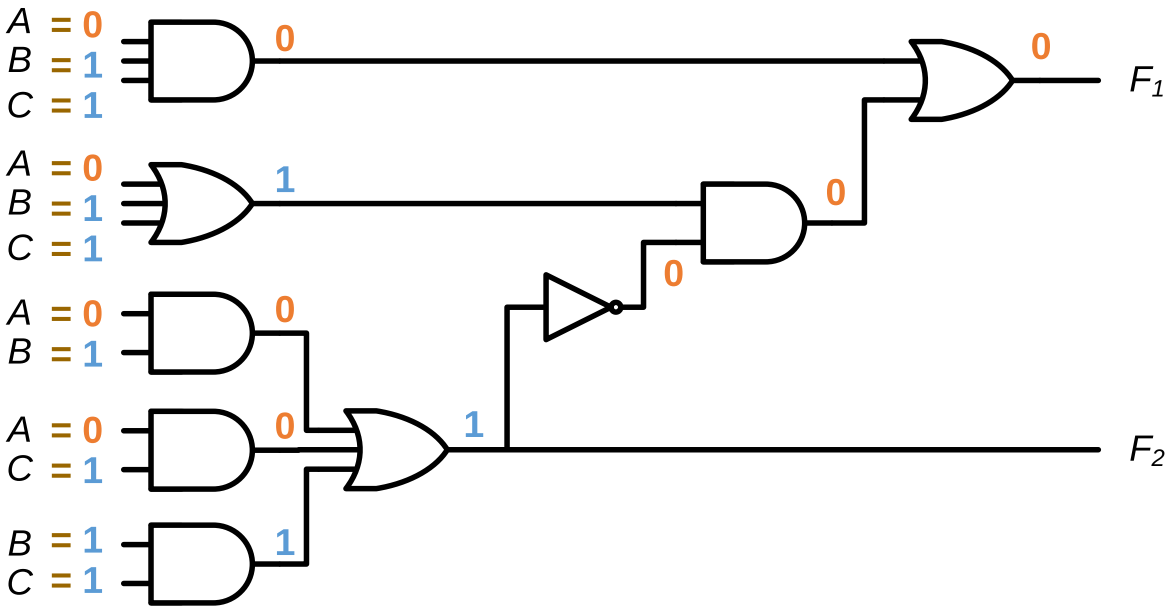
- **Truth Table Approach**



A B C	F_1	F_2
0 0 0	0	0
0 0 1	1	0
0 1 0	1	0

Analysis Procedure

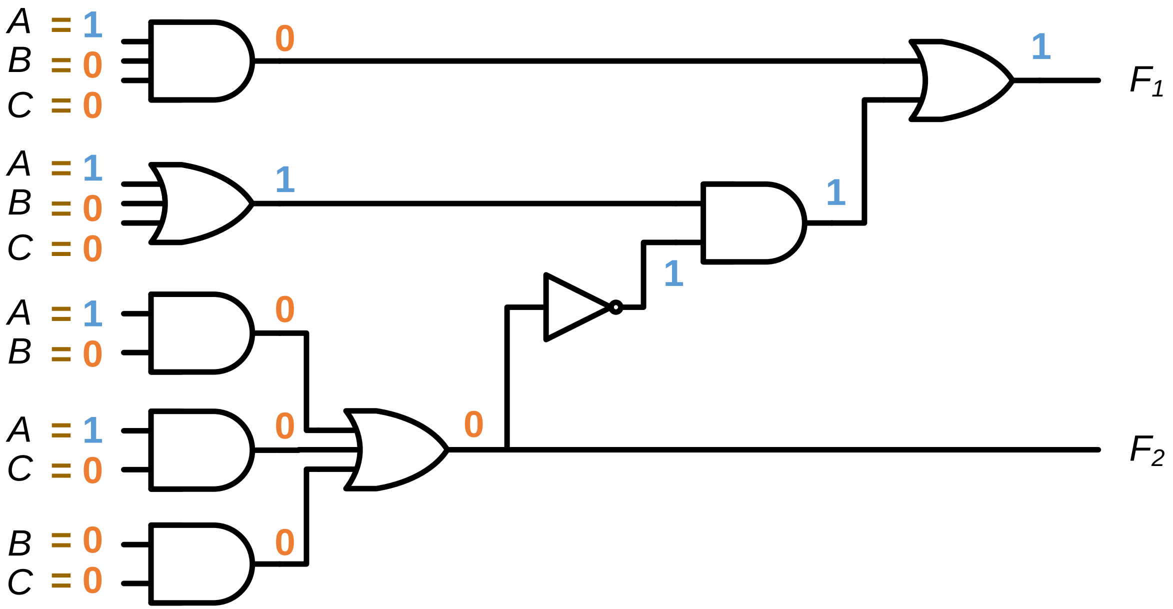
- **Truth Table Approach**



A B C	F_1	F_2
0 0 0	0	0
0 0 1	1	0
0 1 0	1	0
0 1 1	0	1

Analysis Procedure

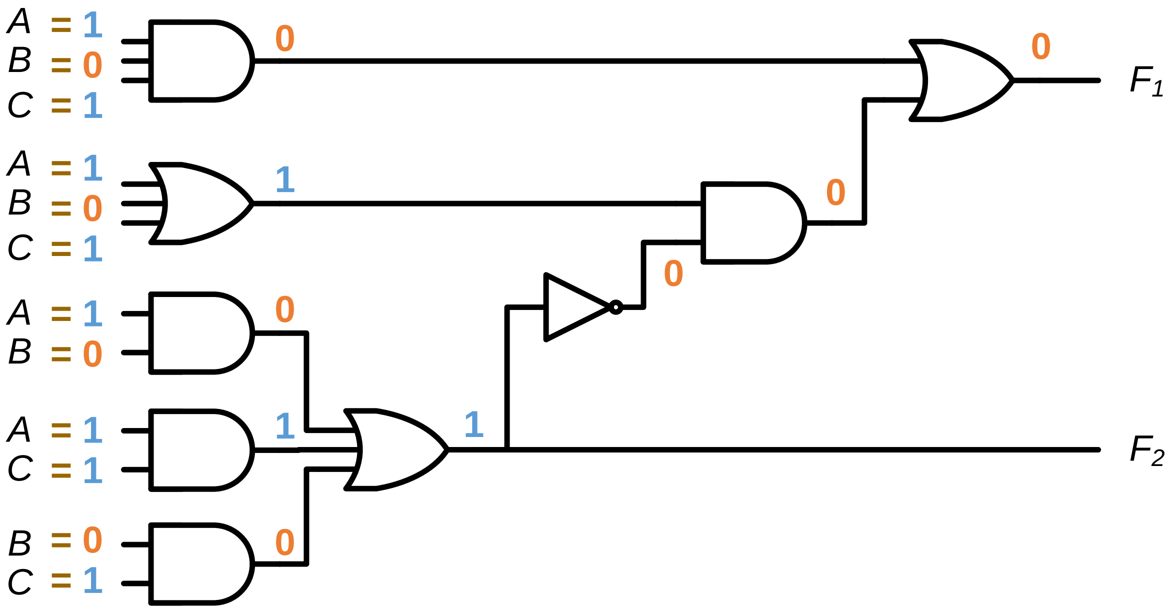
- **Truth Table Approach**



A B C	F_1	F_2
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

Analysis Procedure

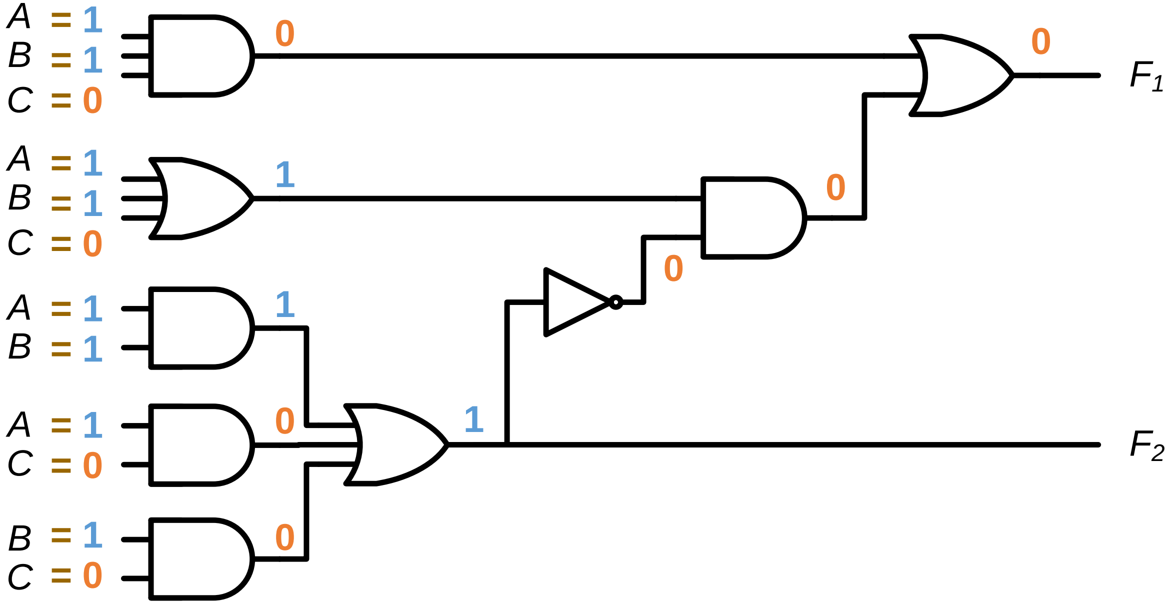
- **Truth Table Approach**



A B C	F_1	F_2
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

Analysis Procedure

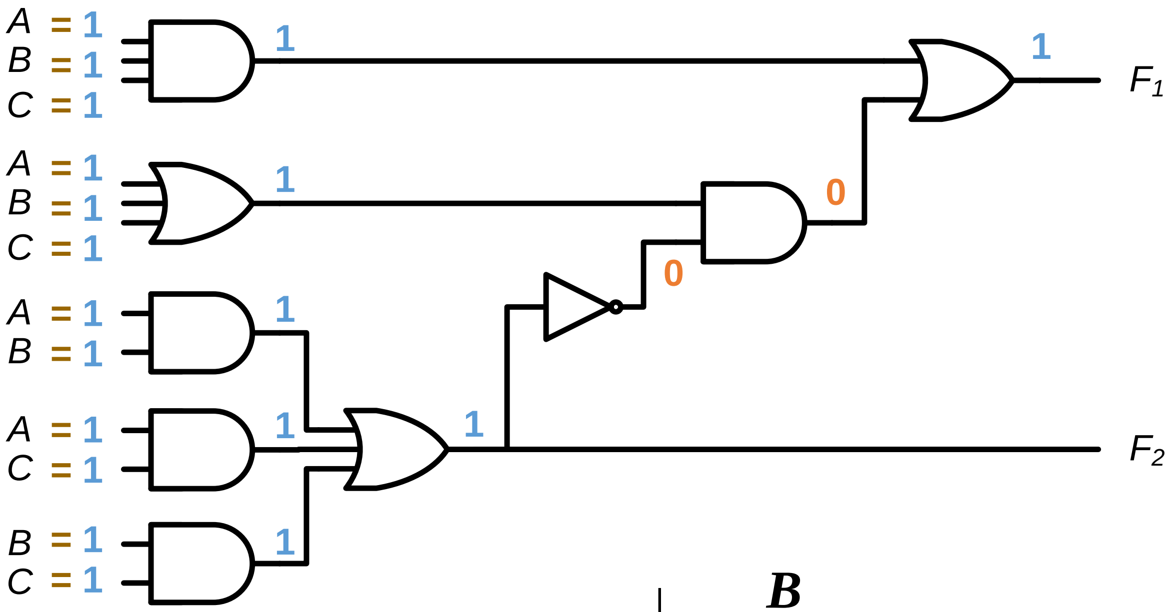
- **Truth Table Approach**



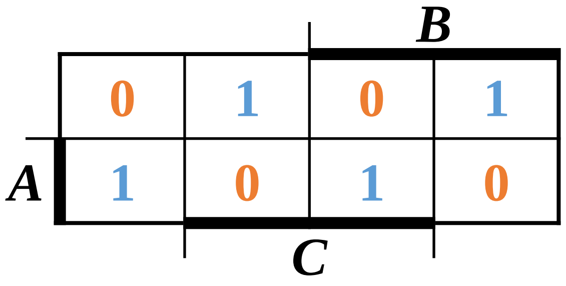
A B C	F_1	F_2
0 0 0	0	0
0 0 1	1	0
0 1 0	1	0
0 1 1	0	1
1 0 0	1	0
1 0 1	0	1
1 1 0	0	1

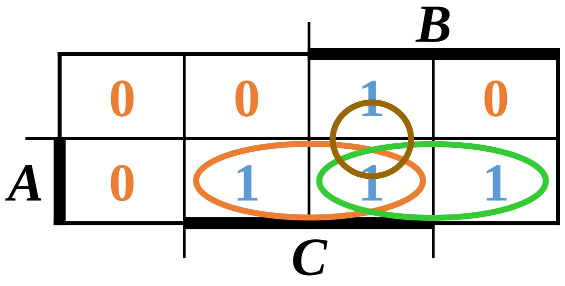
Analysis Procedure

- **Truth Table Approach**



A B C	F_1	F_2
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





$F_1 = AB'C' + A'BC' + A'B'C + ABC$

$F_2 = AB + AC + BC$

Code Converter

- The availability of a large variety of codes for the same discrete elements of information results in the use of different codes by different digital systems.
- It is sometimes necessary to use the output of one system as the input to another.
- A code conversion circuit must be inserted between the two systems if each uses different codes for the same information.
- Thus, a code converter is a circuit that makes the two systems compatible even though each uses a different binary code.

Contd.

- To convert from binary code A to binary code B, the input lines must supply the bit combination of elements as specified by code A and the output line must generate the corresponding bit combination of code B.
- A combinational circuit perform this transformation by means of logic gates.
- The design procedure of code converters will be illustrated by means of a specific example of conversion from the BCD to the excess-3 code.

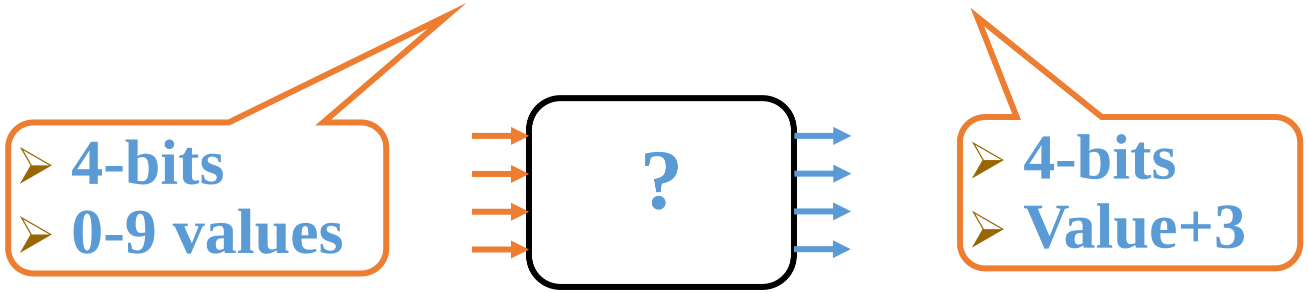
Design Procedure:

Given a problem statement:

- **Determine the number of *inputs* and *outputs***
- **Derive the truth table**
- **Simplify the Boolean expression for each output**
- **Produce the required circuit**

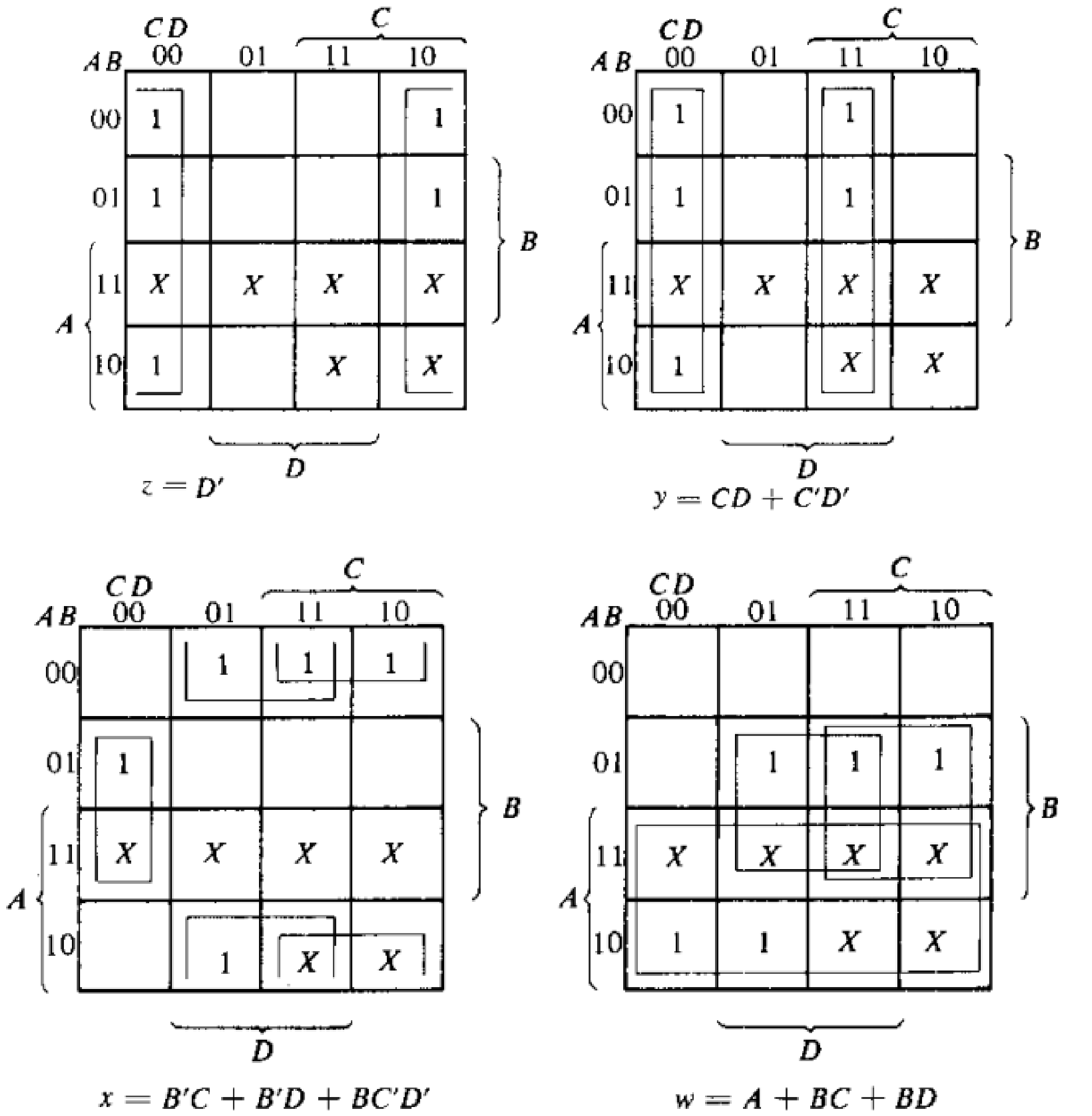
Example:

Design a circuit to convert a "BCD" code to "Excess 3" code



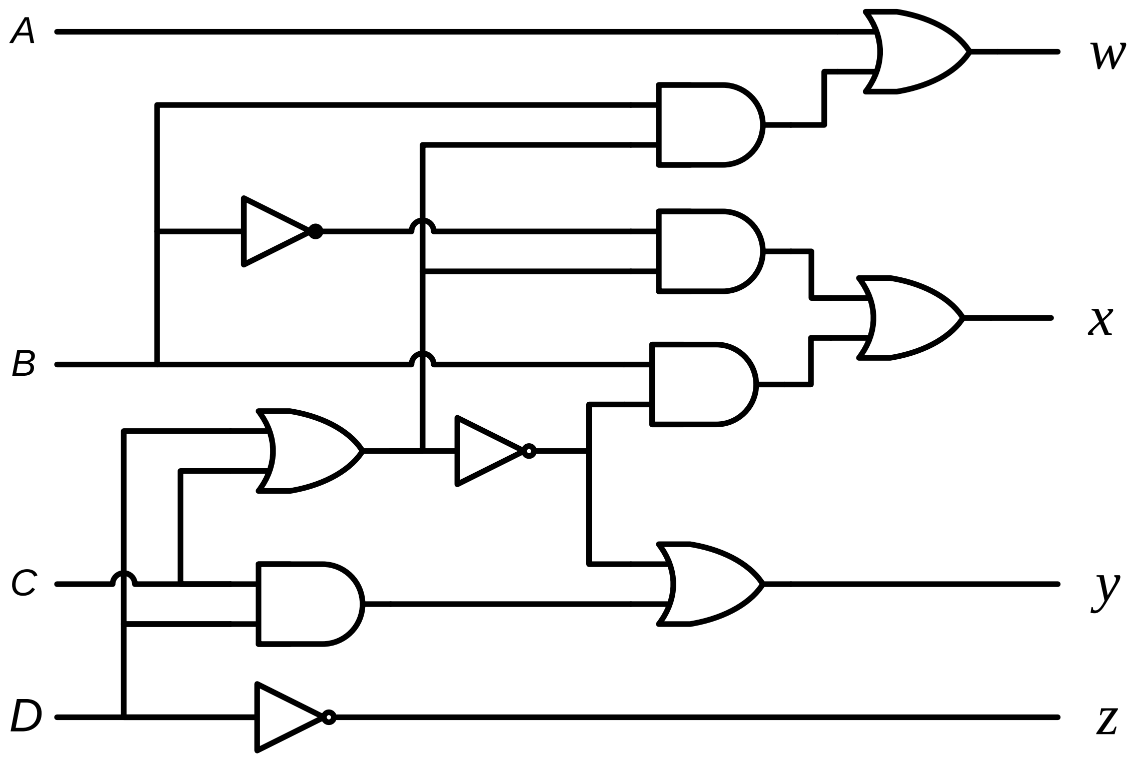
BCD-to-Excess 3 Converter

A B C D	w x y z
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0
1 0 1 0	x x x x
1 0 1 1	x x x x
1 1 0 0	x x x x
1 1 0 1	x x x x
1 1 1 0	x x x x
1 1 1 1	x x x x



BCD-to-Excess 3 Converter

A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x



$w = A + B(C+D)$

$x = B'(C+D) + B(C+D)'$

$y = (C+D)' + CD$

$z = D'$

Gray to Binary Code

- Design a circuit to convert a 3-bit Gray code to a binary code
- The formulation gives the truth table on the right
- It is obvious from this table that X = A and the Y and Z are more complex

Gray A B C	Binary X Y Z
0 0 0	0 0 0
0 0 1	0 0 1
0 1 1	0 1 0
0 1 0	0 1 1
1 1 0	1 0 0
1 1 1	1 0 1
1 0 1	1 1 0
1 0 0	1 1 1

Seven-Segment Decoder

- **A digital or binary decoder is a digital combinational logic circuit which can convert one form of digital code into another form.**
- **BCD to 7-segment display decoder is a special decoder which can convert binary coded decimals into another form which can be easily displayed through a 7-segment display.**

BCD

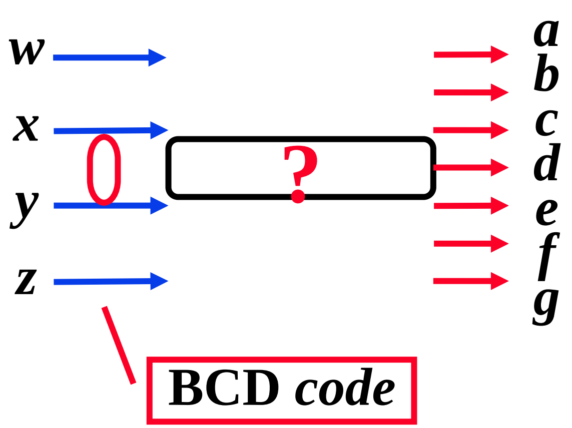
- **BCD stands for binary coded decimal.**
- **It is a digital numbering system in which we can represent each decimal number using 4 bits of binary numbers.**
- **There are 10 digits in the decimal system. To represent all 10 digits we need 10 combinations of 4 binary bits.**

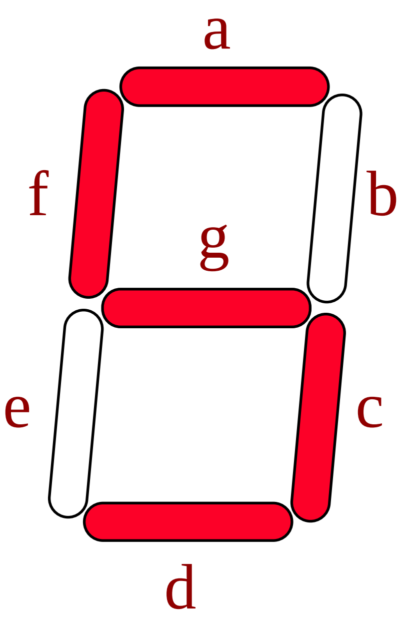
7-Segment Display

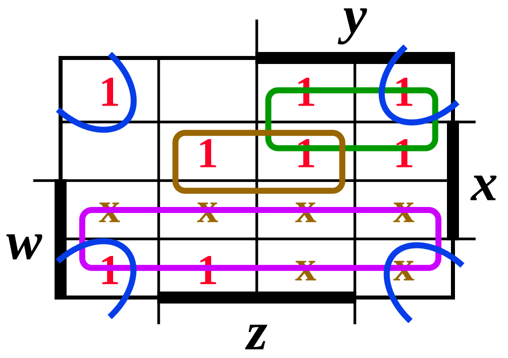
- **It is a digital device that can be used for displaying decimal number, alphabets, and characters.**
- **7-Segment display contains 7 LED segments arranged in a shape given in figure.**
- **Generally, there are 8 input pins in a 7-Segment display.**
- **7 input pins for each of the 7 LEDs and one pin for the common terminal.**

Seven-Segment Decoder

w x y z	a b c d e f g
0 0 0 0	1 1 1 1 1 1 0
0 0 0 1	0 1 1 0 0 0 0
0 0 1 0	1 1 0 1 1 0 1
0 0 1 1	1 1 1 1 0 0 1
0 1 0 0	0 1 1 0 0 1 1
0 1 0 1	1 0 1 1 0 1 1
0 1 1 0	1 0 1 1 1 1 1
0 1 1 1	1 1 1 0 0 0 0
1 0 0 0	1 1 1 1 1 1 1
1 0 0 1	1 1 1 1 0 1 1
1 0 1 0	x x x x x x x
1 0 1 1	x x x x x x x
1 1 0 0	x x x x x x x
1 1 0 1	x x x x x x x
1 1 1 0	x x x x x x x
1 1 1 1	x x x x x x x







$a = w + y + xz + x'z'$

$b = . . .$

$c = . . .$

$d = . . .$

Why we need this decoder?

- **A digital system like a computer can understand and easily read a large number in binary format.**
- **However, a human cannot read large binary numbers.**
- **To solve this problem we need to display it as a decimal digit using 7-segment display.**