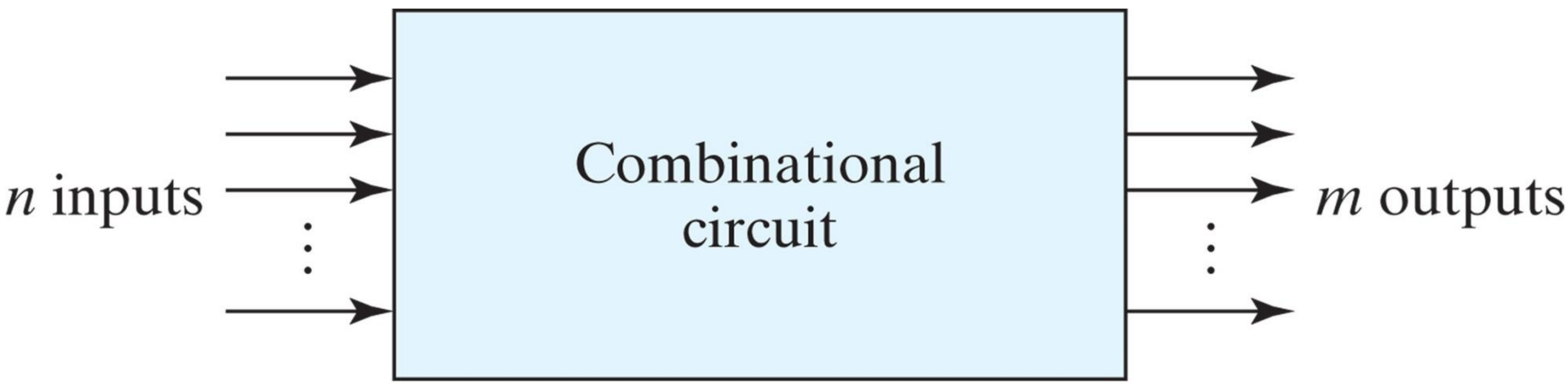
Chapter Four

Combinational Logic

- Two basic types of logic circuits
  - Combinational
    - Output(s) depend solely on the inputs
  - Sequential
    - Output(s) depend on inputs and a feedback mechanism.
    - More on sequential circuits in Chapter 5


Combinational circuit
n inputs
m outputs

For n inputs

2^n possible combinations

For each combination of inputs, there is

one possible value for each output variable.

Therefore, a combinational circuit can be described by a

Truth Table.

Common Combinational Circuits

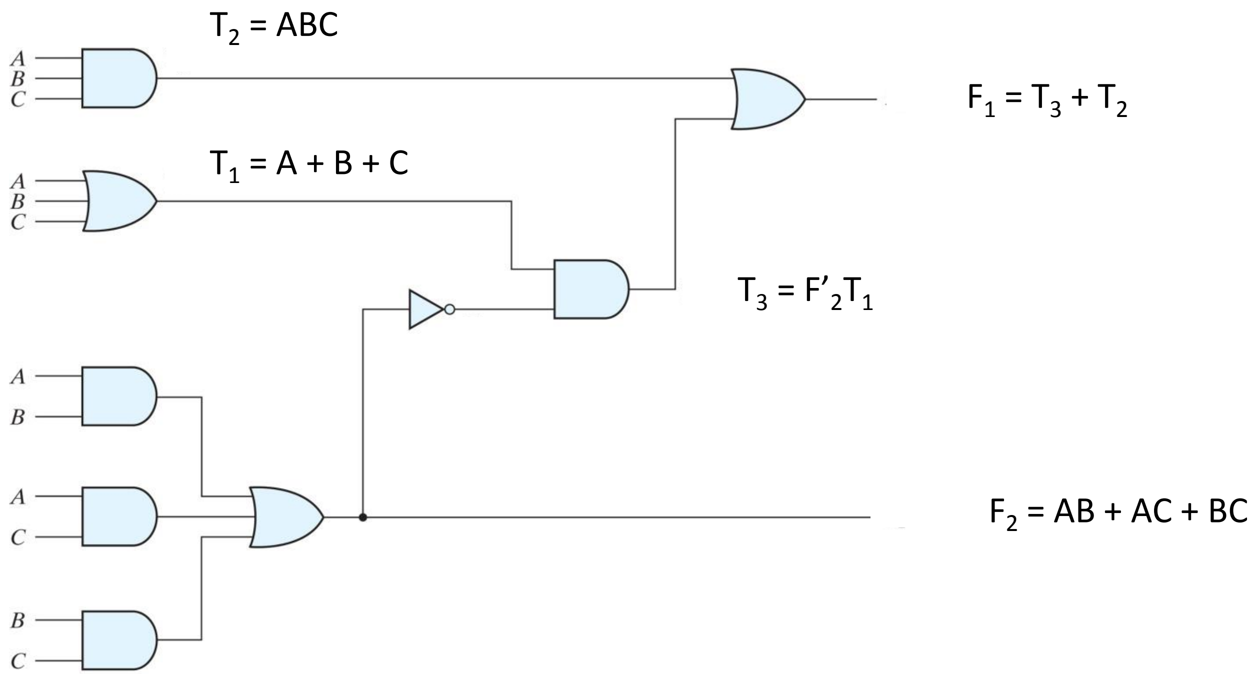
- Adders/subtractors
- Comparators
- Decoders
- Encoders
- Multiplexers.

Analysis of a given circuit

1. Make sure the circuit is combinational and not sequential.
   - No Feedback paths
   - No Memory elements
2. Obtain the Boolean functions or truth table applicable to the circuit.

Obtaining the Boolean Functions

1. Label all gate outputs that are a function of the inputs.
   - Labels are arbitrary, BUT
   - Names should be meaningful
2. Label the gates that are a function of input variables and/or previously labeled gates.
3. Repeat Step 2
4. Substitution



$T_2 = ABC$
$T_1 = A + B + C$
$T_3 = F'_2T_1$
$F_1 = T_3 + T_2$
$F_2 = AB + AC + BC$
A
B
C
A
B
C
A
B
A
C
B
C

Substitution

$F_1 = T_3 + T_2$

$F_1 = T_3 + T_2$

$T_3 = F'_2 T_1$

$T_2 = ABC$

$T_1 = A + B + C$

$F_2 = AB + AC + BC$

And now derive the truth table

Table 4.1
Truth Table for the Logic Diagram of Fig. 4.2

A	B	C	F_2	F'_2	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	1	0	1

Design

- Starts with a specification.

1. Determine the number of required inputs.
2. Derive the truth table.
3. Obtain the simplified Boolean Functions.
4. Implement the circuit.

Example Circuit

Need a circuit that takes 2 inputs and has 2 outputs.

When the inputs are both 0, the outputs are both zero.

When the inputs are both 1, the outputs are 1 and 0, respectively.

When the inputs are different, the outputs are 0 and 1 respectively.

- How many inputs are needed?
  - 2, let’s call them x and y.
- How many outputs are needed?
  - 2, let’s call them C and S
- Build the table.

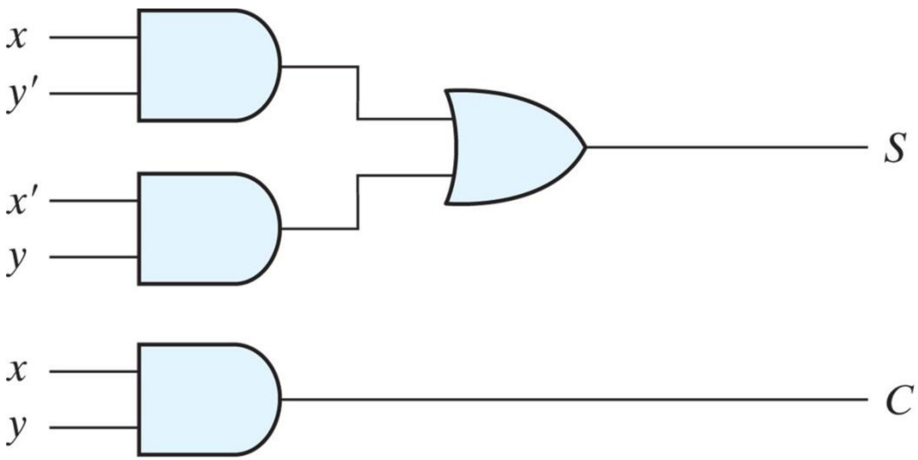
x	y	C	S
0	0		
0	1		
1	0		
1	1		

- How many inputs are needed?
  - 2, let’s call them x and y.
- How many outputs are needed?
  - 2, let’s call them C and S
- Build the table.

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

$C = xy$

$S = xy' + x'y$

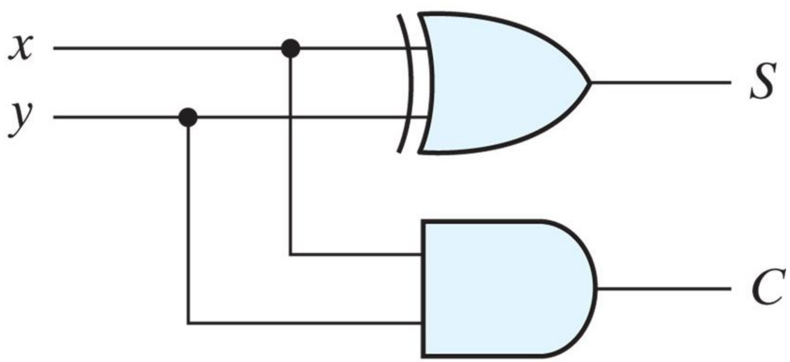


x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$C = xy$

$S = xy' + x'y$

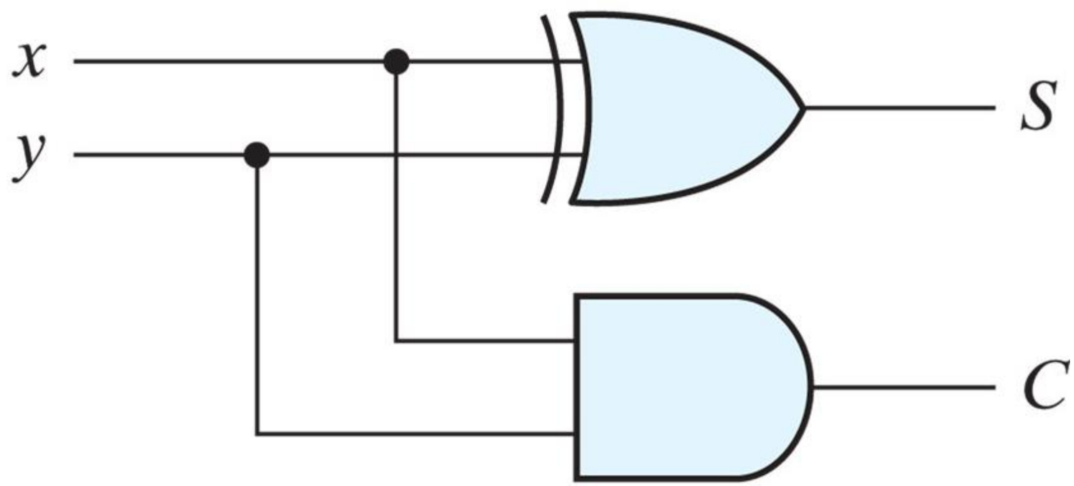
$= x \oplus y$



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

Half-Adder

A combinational circuit that performs the addition of two bits.



Full-Adder

Full Adder

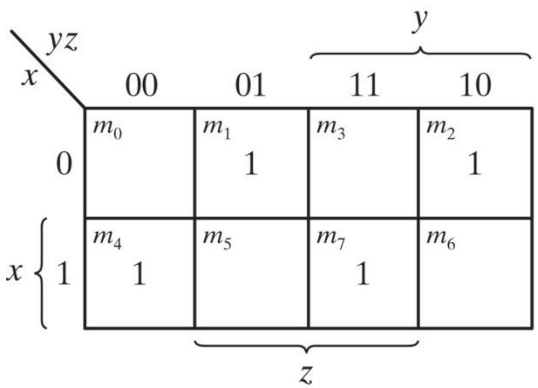
x	*y*	*z*	*C*	*S*
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Full-Adder

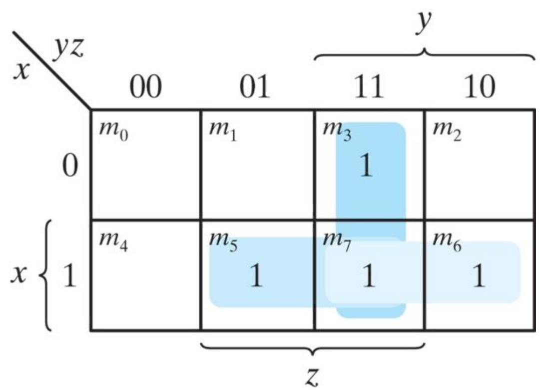
Full Adder

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

Find the Boolean Functions



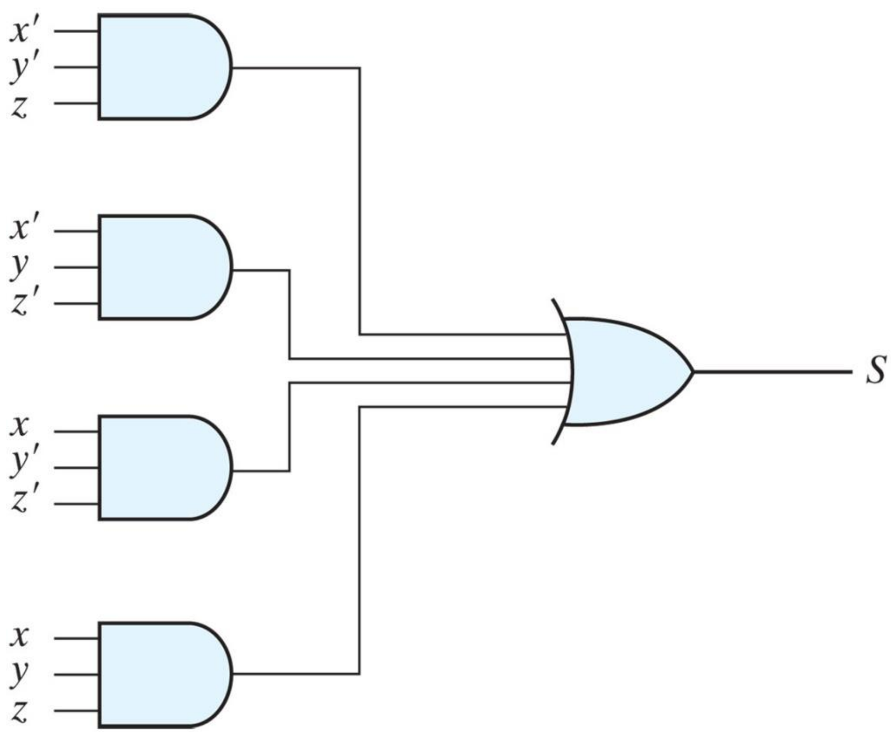
(a) $S = x'y'z + x'yz' + xy'z' + xyz$



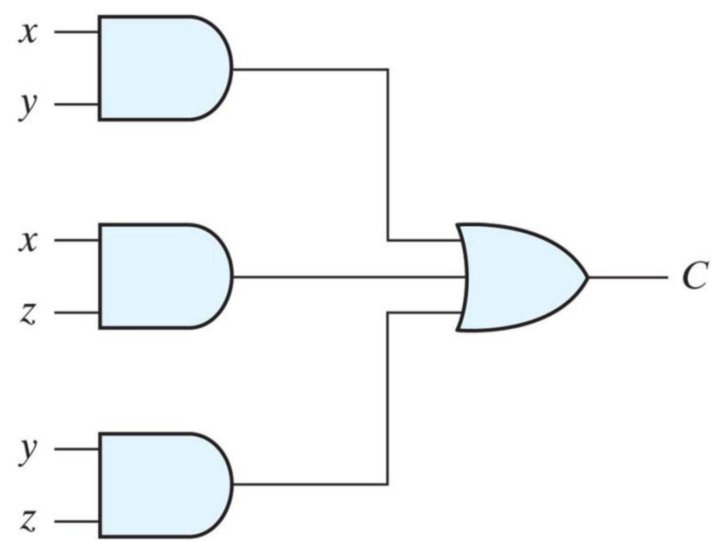
(b) $C = xy + xz + yz$

Full Adder

x	*y*	*z*	*C*	*S*
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



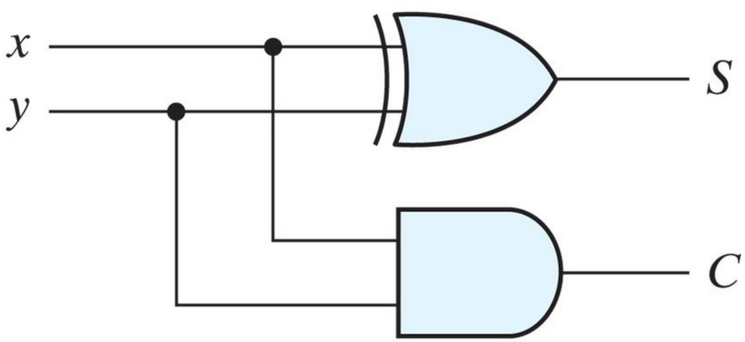
x′
y′
z
x′
y
z′
x
y′
z′
x
y
z
S



x
y
x
z
y
z
C

Further Examination

Further Examination

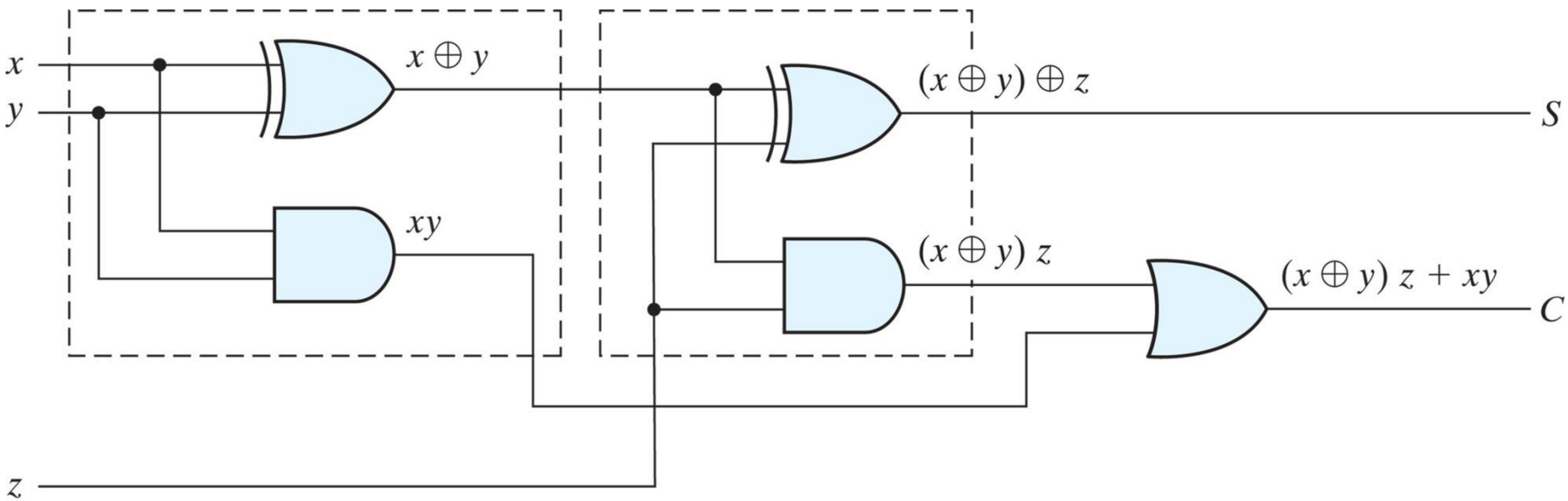


Full Adder

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

Another implementation

Full Adder



Binary Adder

- Provides the arithmetic sum of two binary numbers.
- n full adders cascaded
- OR 1 half-adder and n-1 full adders cascaded

Full-Adders Cascaded