



# Combinatorial logic functions

## Exercises Digital Design

### Solution vs. Hints:



While not every response provided herein constitutes a comprehensive solution, some serve as helpful hints intended to guide you toward discovering the solution independently. In certain instances, only a portion of the solution is presented.

## 1 | COM - Combinatorial function representations

### 1.1 Truth Table

Six different actions possible:

- no call
- already there
- go down
- go up
- door open elevator stays
- undefined

*com/representation-01*

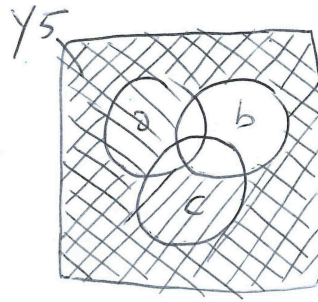
### 1.2 Truth Table from a Chonogram

$$y = a \oplus b \oplus c \quad (1)$$

*com/representation-02*

### 1.3 Representation by Venn diagram

Only the solution for  $y_5$  is given:



*com/representation-03*

## 1.4 Simplification by Venn diagram

ab

(2)

*com/representation-04*



## 2 | COM - Elementary logic functions

### 2.1 Switch Circuits

$ab, a + b$

*com/logic-functions-01*

### 2.2 Truth table of Elementary Functions

Each columns has two '1'.

*com/logic-functions-02*

### 2.3 Elementary Functions in a Truth Table

Not Available

*com/logic-functions-03*

### 2.4 Number Decoding

$$\text{red} = \overline{c_2} + c_2 \overline{c_1} \overline{c_0} \quad (3)$$

$$\text{orange} = \overline{c_2} c_1 + c_2 \overline{c_1} \overline{c_0} \quad (4)$$

$$\text{green} = c_2 (c_1 + c_0) \quad (5)$$

*com/logic-functions-04*



## 3 | COM - Boolean algebra

### 3.1 Proofs

It can be either done with a Venn-Diagram, a Truthtable or Boolean Algebra.

*com/algebra-01*

### 3.2 De Morgan

$$\overline{a + b + \overline{c}d} = \overline{a} \overline{b} c + \overline{a} \overline{b} \overline{d} \quad (6)$$

*com/algebra-02*

### 3.3 Redundancy with the XOR function

$$a = y \oplus b$$

*com/algebra-03*

### 3.4 XOR function

$$\overline{a \oplus b} = ab + \overline{a} \overline{b}$$

*com/algebra-04*

### 3.5 Polynomial Form

$$\overline{\overline{a}b + \overline{b}c + \overline{c}a} = ab + bc + ca$$

*com/algebra-05*



## 4 | COM - Complete operators

### 4.1 Create a function using NAND gates

You need:

- 9 NAND with 2 inputs
- 1 NAND with 4 inputs

*com/operators-01*

### 4.2 Create a function using NAND gates

You need:

- 13 NAND with 2 inputs

*com/operators-02*

### 4.3 Create a function using NAND gates

You need:

- 12 NAND with 2 inputs for a minimal version

*com/operators-03*

### 4.4 NOR-Operator

- Inverter = 1 NOR
- AND = 3 NOR
- OR = 2 NOR

*com/operators-04*

### 4.5 Create a function using NOR gates

You need:

- 11 NOR with 2 inputs
- 1 NOR with 4 inputs

*com/operators-05*

### 4.6 Create a function using inverting gates

You need:

- 9 NAND with 2 inputs
- 1 NAND with 4 inputs

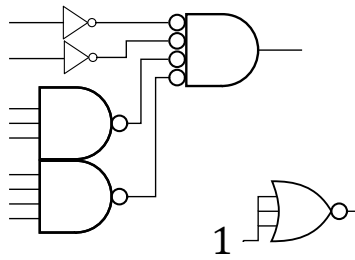
*com/operators-06*

### 4.7 Versatile circuit

NAND, OR, NOR solution not available. Example AND-9:



## AND-9



*com/operators-07*

### 4.8 Creation of an XOR function

You need two 74HC7006 Chips

*com/operators-08*