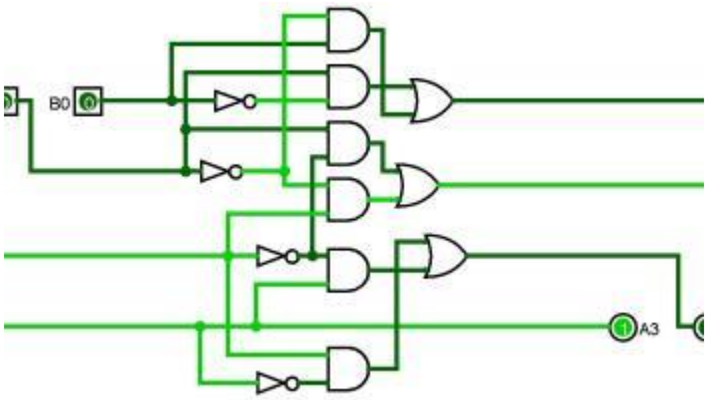
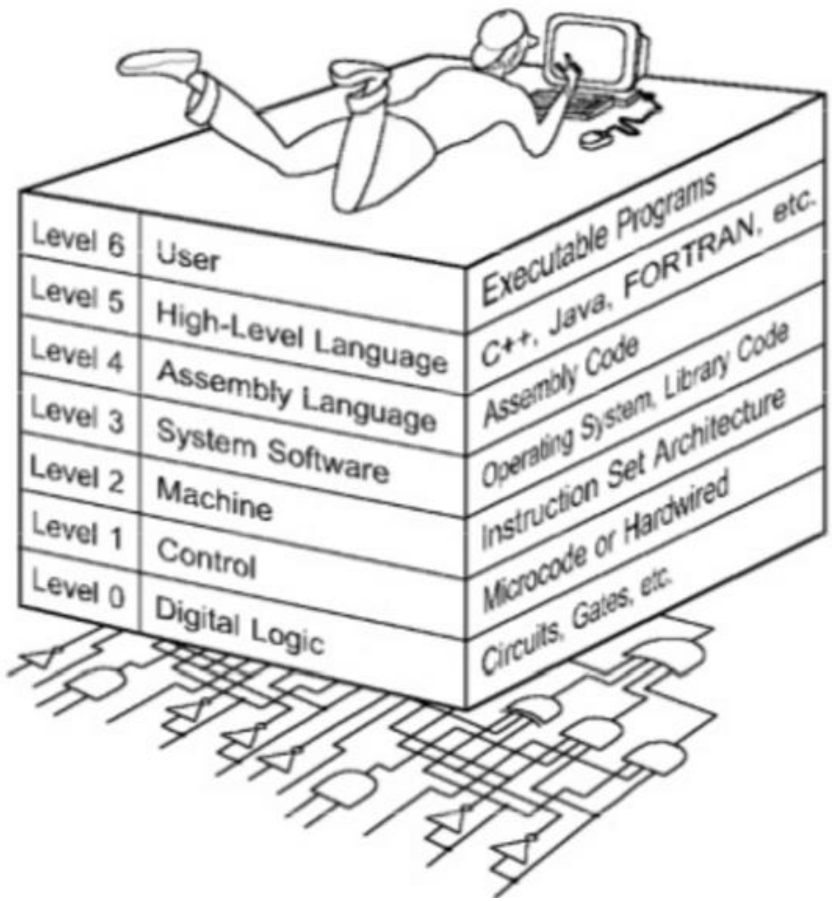


4. Digital Logic

Chapter 11
Combinational Logic Circuits

Computer Level Hierarchy



http://www.slideshare.net/jsrao78/introduction-to-computer-architecuure

Sections 11.3

- Reading: pp.370-380
- Good Problems to Work: 11.4, 11.5, 11.6, 11.8

COMBINATIONAL LOGIC CIRCUITS

Combinational Logic Circuit

- When designing a combinational logical circuit, the designer works from two sets of known values
  1. the input states that the circuit can take
  2. the output states for each input

Example

- A combinational circuit has two inputs, X and Y, and a single output Z. The relationship between the inputs and the outputs is:
    1. When X and Y are 0, Z is 0
    2. When X and Y are 1, Z is 1
    3. When X is 0 and Y is 1, Z is 0
    4. When X is 1 and Y is 0, Z is 1

Example

- The following truth table represents the circuit on the previous slide

X	**Y**	**Z**
0	0	0
0	1	0
1	0	1
1	1	1

Inputs: X, Y
Output: Z

SUM OF PRODUCTS
PRODUCT OF SUMS

Boolean Expressions

- Certain types of Boolean expressions lead to circuits that are more desirable from an implementation viewpoint.
- **product term** – a single variable or the logical product of several variables. The variables may or may not be complemented.
- **sum term** – a single variable or the logical sum of several variables. The variables may or may not be complemented.

Boolean Expressions

- **X** is both a sum term and a product term
  X+Y is a sum term
  XY is a product term
  X+YX is neither
- **Sum of products (SOP)** - a product term or several product terms logically added
- **Product of sums (POS)** - a sum term or several sum terms logically multiplied

Example: Sum of Products

- Add a column of product terms to the truth table of the previous example

X	Y	Z	Product Term
0	0	0	$\bar{X}\bar{Y}$
0	1	0	$\bar{X}Y$
1	0	1	$X\bar{Y}$
1	1	1	XY

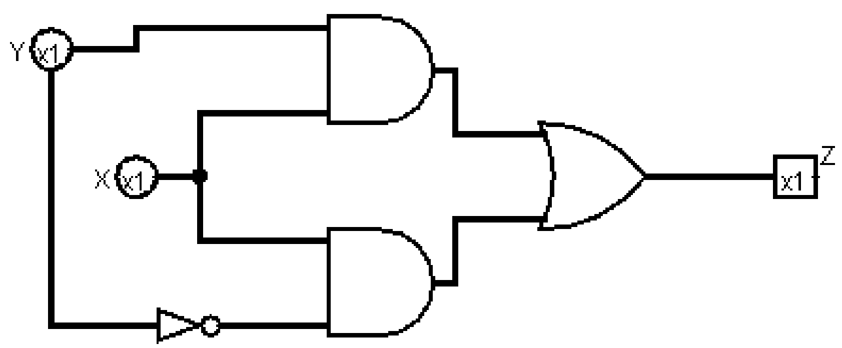
Inputs: X, Y
Output: Z

Example: Sum of Products

- When Z equals 1, the product term is used in the sum-of-products
- A product term containing all input variables is called **canonic**.
- A canonical expansion is the logical sum of the product terms where the output (Z in this case) is a 1.
- The canonical expansion for the logical network is therefore: $Z = X\bar{Y} + XY$

Combinational Logical Circuit Realized

- Draw the circuit diagram for: $Z = X\bar{Y} + XY$ using OR gates, AND gates, and inverters



ALGEBRAIC SIMPLIFICATION

Simplifying

- We are interested in finding the least expensive combinational logic circuit; therefore, we might use some Boolean algebra to simplify the canonical expansion.

- How may we simplify the circuit on the previous slide?

Problem: Sum of Products

- Write the Boolean expression (in sum-of-products form) for a logic network that will have an output F=1 when: A=1, B=0, C=0 and A=1, B=1, C=0. All other combinations of input will produce an output of 0.

- Derive the expression.
- Simplify the derived expression.
- Draw the logic diagram for the simplified expression.

Product of Sums

- How to do this?
  - Construct a truth table of input and output values
  - Identify the rows that output a 0
  - For those rows, write out the sum of the inputs. If the input is 0, then write the input; if the input is 1, then write the input's complement
  - Multiply all the sums

Problem

- For the following truth table:
  - Give a sum-of-products expression
  - Give a product-of-sums expression
  - Draw the logical circuitry that realizes each expression

Inputs		**Output**
X	**Y**	**F**
0	0	1
0	1	0
1	0	0
1	1	1

NAND Gate Design

- NAND gates are widely used in the design of modern computers. In general, NAND gates take less resistors than AND and NOR gates take less resistors than OR.
- To design a NAND-to-NAND two level gate network, we produce the sum-of-products expression.
- A sum-of-products expression can be drawn using AND gates at the first level and an OR gate at the second level. Finally, we can replace all AND and OR gates with NAND gates

Problem

- Develop the sum-of-products that describe the function of the following truth table (X,Y, and Z are inputs, A is output)
- Draw the logical circuitry using only AND and OR logic. Then design the logical circuitry using only NAND gates

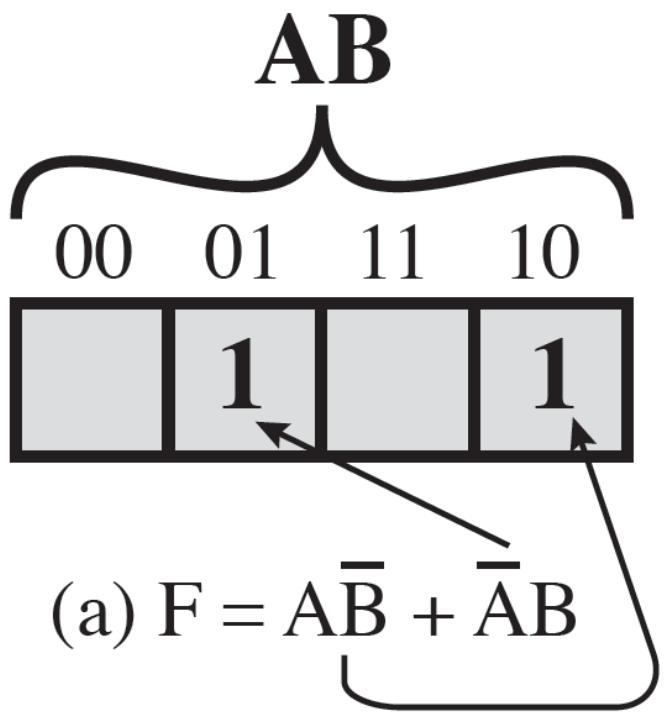
X	**Y**	**Z**	**A**
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

SIMPLIFICATION USING KARNAUGH MAP

Karnaugh Map (K-map)

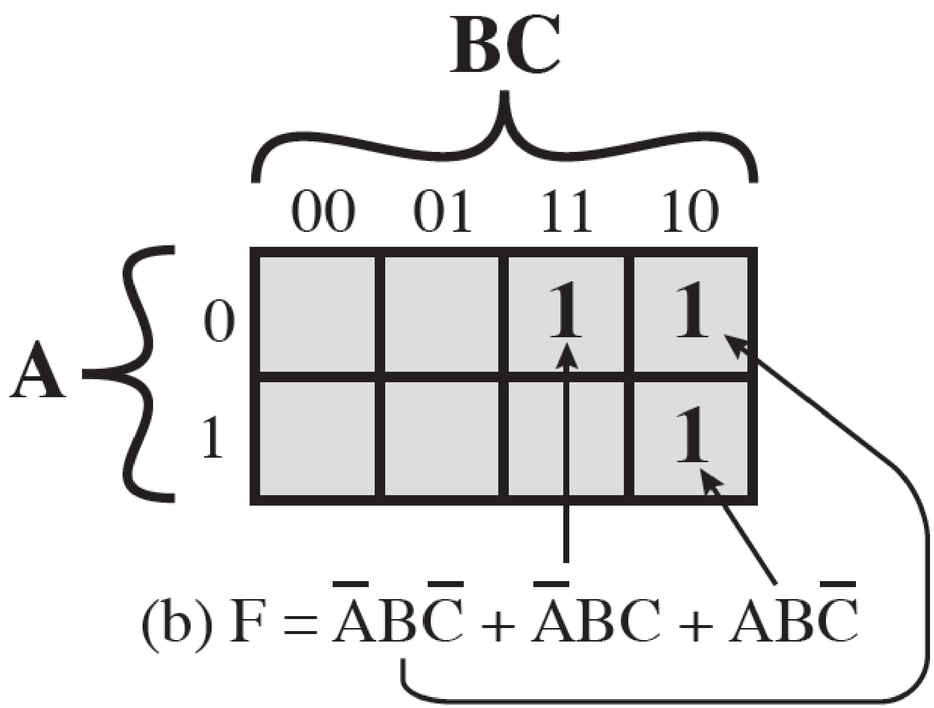
- A Karnaugh Map is a way of representing and simplifying a Boolean function for up to 4 variables.
- The map represents all possible combinations of n binary variables and thus contains 2^n possible squares (where n is the number of variables).

2 Variable K-map

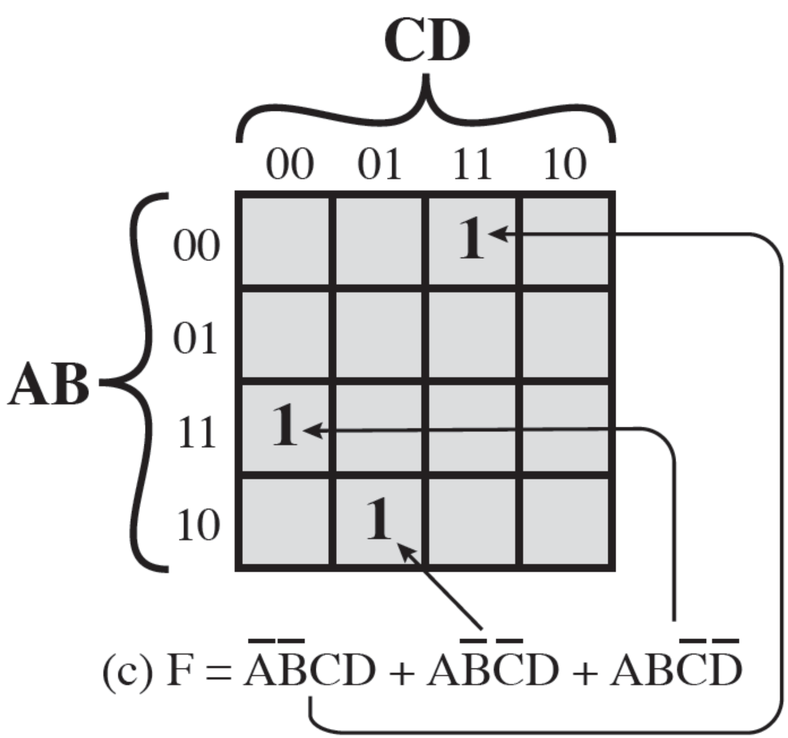


(a) $F = A\overline{B} + \overline{A}B$

3 Variable K-map



4 Variable K-map

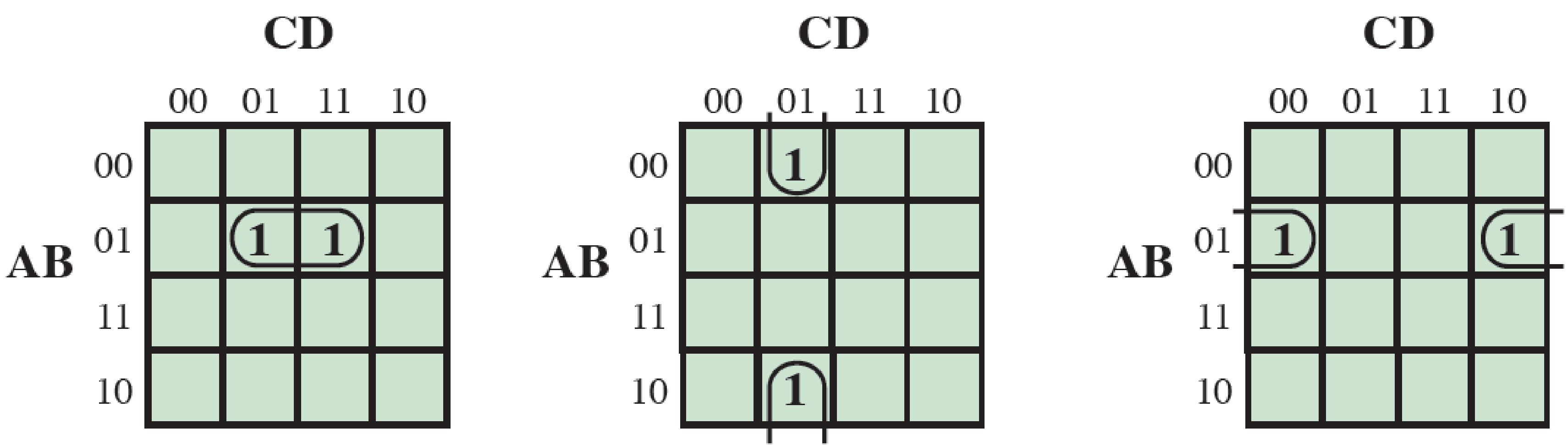


(c) $F = \overline{A}\,\overline{B}CD + A\overline{B}\,\overline{C}D + AB\overline{C}\,\overline{D}$

Simplification using K-maps

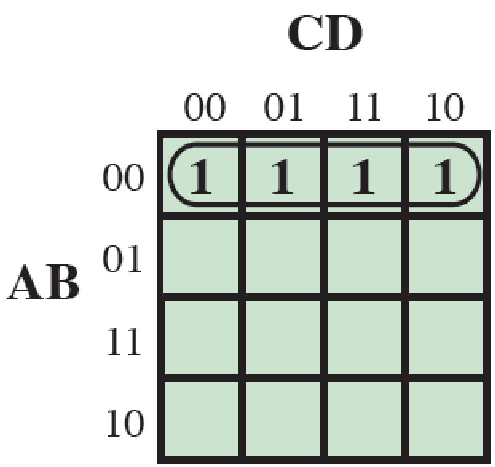
- Among the squares with a 1, find those that belong to the largest "block" of 1, 2, 4, or 8. Circle those blocks.
- Select additional blocks so that every 1 in the square is a member of some block. 1's can be included in multiple blocks.
- A nice explanation of K-maps can be found at: http://www.facstaff.bucknell.edu/mastascu/elessonshtml/Logic/Logic3.html

Simplification using K-maps

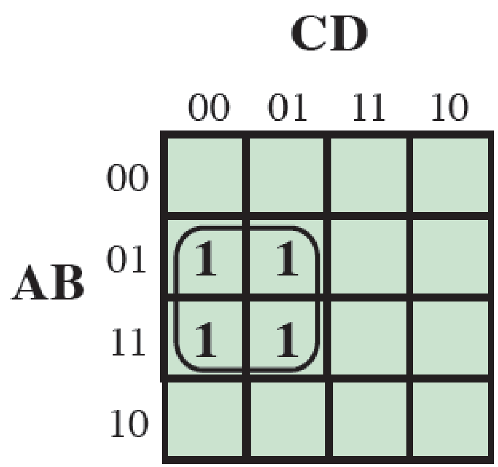


(a) $\overline{A}BD$ (b) $\overline{B}\overline{C}D$ (c) $\overline{A}B\overline{D}$

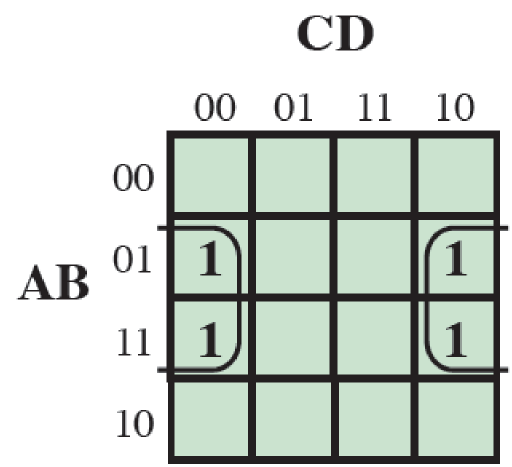
Simplification using K-maps



(d) $\overline{A}\overline{B}$

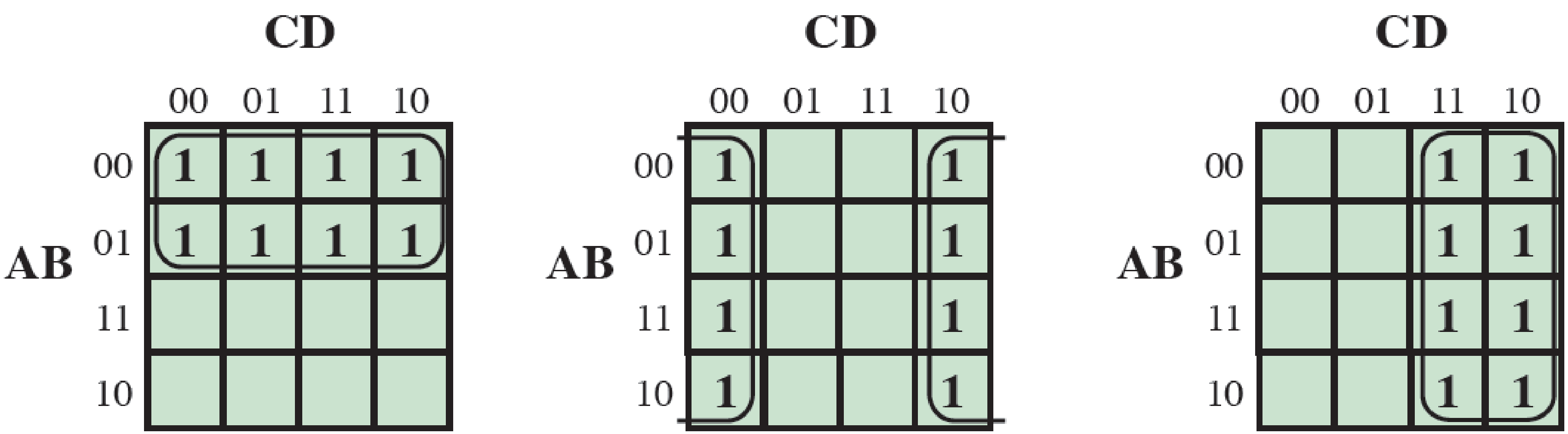


(e) $B\overline{C}$



(f) $B\overline{D}$

Simplification using K-maps



(g) $\overline{A}$ (h) $\overline{D}$ (i) C

Problem

- Draw the K-map (A, B, C are inputs; F is output)
- Give the Boolean function for the simplest sum-of-products form
- Draw the circuit using all NANDs

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0