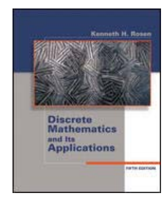
Chapter 10 (Part 2): Boolean Algebra

- Logic Gates (10.3) (cont.)
- Minimization of Circuits (10.4)



Logic Gates (10.3) (cont.)

- Adders
  - Question: How can we carry out operations such as additions of two positive integers (in binary expansion) using logic circuits?
  - We first build a circuit that can be used to determine x + y where x and y are two bits

Input		Output	
x	y	s	c
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Logic Gates (10.3) (cont.)

– The output consists of two bits s and c which are the sum bit and the carry bit respectively

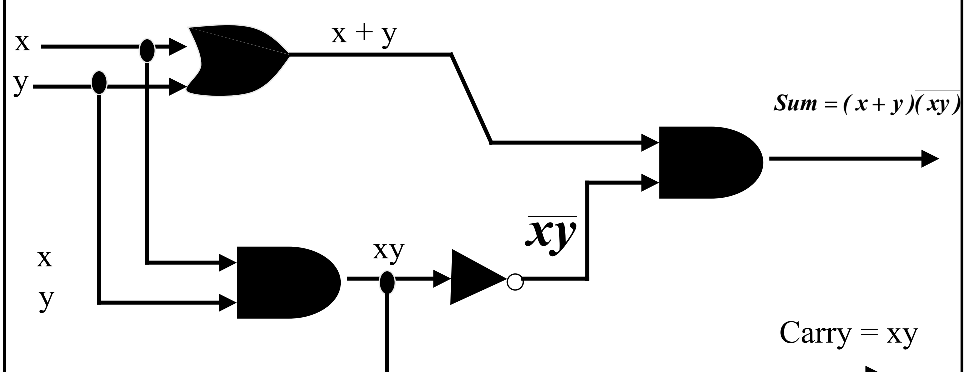
– From the previous table, we can write:

i. $s = x\bar{y} + \bar{x}y = (x+y)\overline{(xy)}$

ii. c = xy

Logic Gates (10.3) (cont.)

– The following circuit called half adder illustrates the sum of two bits with carry



- We need to compute the sum bit and the carry bit when two bits and a carry are added
- The inputs to this full adder are the bits x and y and the carry c_i
- The outputs are the sum bit s and the new carry c_{i+1}.

Input			Output	
x	y	c_i	s	C_{i+1}
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

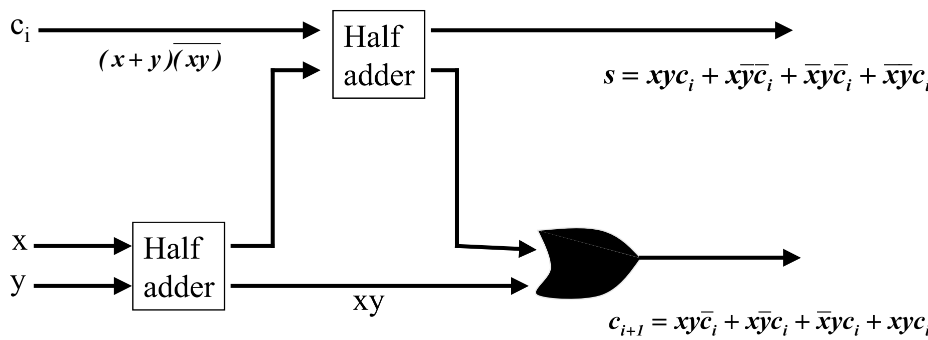
Logic Gates (10.3) (cont.)

- The two outputs of the full adder, the sum bit s and the carry c_{i+1} are given by the sum-of-products expansion

  $xyc_i + x\overline{y}\overline{c_i} + \overline{x}y\overline{c_i}$ and $xy\overline{c_i} + x\overline{y}c_i + \overline{x}yc_i$, respectively

- The full adder circuit using half adders is depicted in the following picture

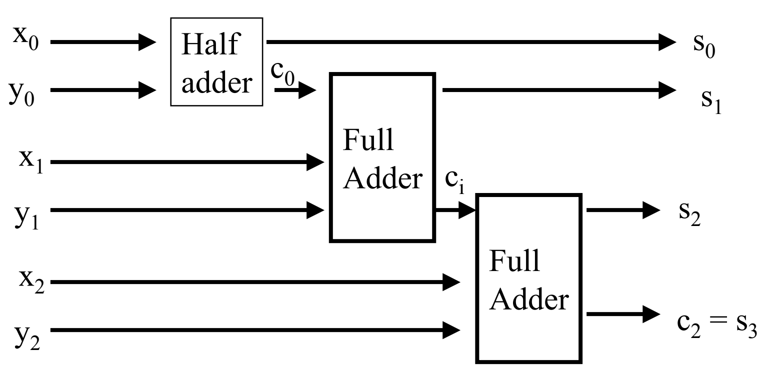
Logic Gates (10.3) (cont.)



A full adder

Logic Gates (10.3) (cont.)

– Finally, the following figure shows how the half and the full adders can be combined to add the two three-bit integers $(x_2x_1x_0)_2$ and $(y_2y_1y_0)_2$ to produce the sum $(s_3s_2s_1s_0)_2$



Minimization of Circuits (10.4)

- Introduction

  – Goal: Representing circuits with Boolean expressions having fewer operations

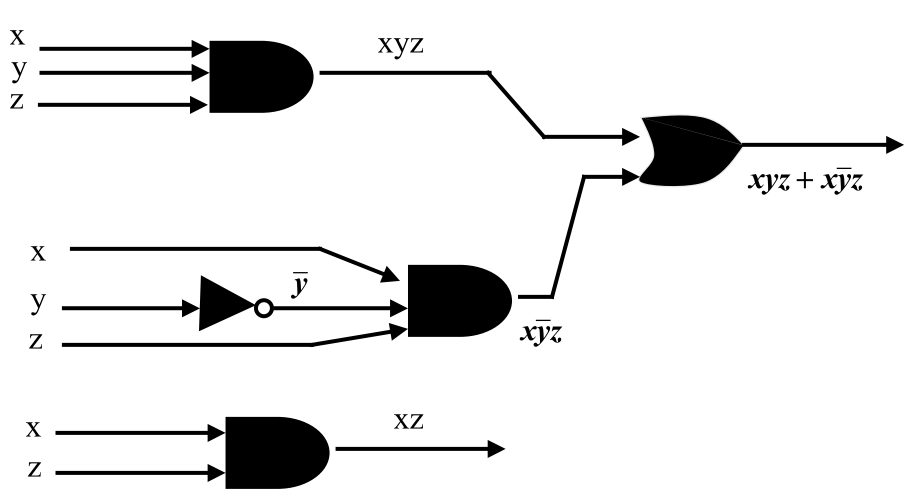
  – Example: a circuit has output 1 if and only if: x = y = z = 1 or x = z = 1 and y = 0. The sum-of-product of this circuit is:

$$xyz + x\bar{y}z = (y + \bar{y})(xz) = 1.(xz) = xz.$$

  – xz is a Boolean expression with fewer operators that represents the circuit

  – Therefore, there are 2 different implementations of this circuit as depicted in the following picture

Minimization of circuits (10.4) (cont.)



Two circuits with the same output

Minimization of circuits (10.4) (cont.)

- Karnaugh maps

  - Goal: reduce the number of terms in a Boolean expression representing a circuit
  - Question: How can we find terms to combine for Boolean functions for a relatively small number of variables?
  - Answer: There is a method introduced by Maurice Karnaugh in 1953 for fewer variables.

Minimization of circuits (10.4) (cont.)

- K-map was designed in 1950 to help minimize circuits by hand
- K-maps are useful in the minimization of circuits with up to 6 variables only
- K-maps are inapplicable when dealing with 50, 100, or 1000 variables
- The minimization problem with respect to the number of variables is NP-complete

Minimization of Circuits (10.4) (cont.)

– Principle

- K-maps for two variables:
  - There are 4 possible minterms in the sum-of-product expansion of a Boolean function in 2 variables x and y
  - Adjacent cells are those whose minterms differ in exactly one litteral as shown in the following figure.
  - A k-map for a Boolean function in two variables consists of 4 cells, where a 1 is placed in the cell representing a minterm if this minterm is present in the expansion

	y	$\overline{y}$
x	xy	$x\overline{y}$
$\overline{x}$	$\overline{x}y$	$\overline{x}\,\overline{y}$

Minimization of circuits (10.4) (cont.)

- Example: Find the K-maps for:
  a) $xy + \overline{x}y$
  b) $x\overline{y} + \overline{x}y$
  c) $x\overline{y} + \overline{x}y + \overline{x}\,\overline{y}$

Solution: We include a 1 in a cell when the minterm represented by this cell is present in the sum-of-products expansion.

	y	$\overline{y}$
x	1	
$\overline{x}$	1	

(a)

	y	$\overline{y}$
x		1
$\overline{x}$	1	

(b)

	y	$\overline{y}$
x		1
$\overline{x}$	1	1

(c)

Minimization of circuits (10.4) (cont.)

- Whenever there are 1's in two adjacent cells, the minterms are combined into a product involving just one of the variables
- We circle blocks of cells in the K-maps that represent minterms that can be combined and then find the corresponding sum of the products
- The goal is to identify the largest possible blocks, and to cover all the 1s with the fewest blocks using the largest blocks first and always using the largest possible blocks.

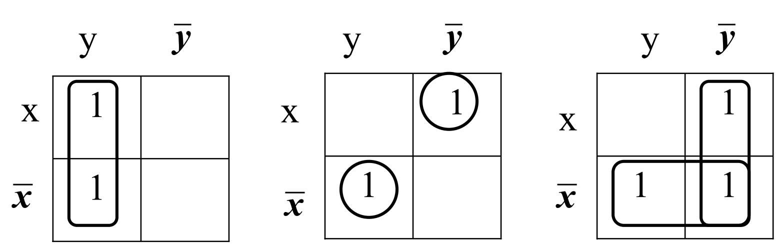
- Example: Simplify the following sum-of-products expansions:

  a) $xy + \overline{x}y$

  b) $x\overline{y} + \overline{x}y$

  c) $x\overline{y} + \overline{x}y + \overline{x}\,\overline{y}$

  Solution: The grouping of minterms is shown in the following figure. Using the k-maps for these expansions Minimal expansions for these sums-of-products are: (a) y; (b) $x\overline{y} + \overline{x}y$; and (c) $\overline{x} + \overline{y}$.



(a) (b) (c)

Minimization of Circuits (10.4) (cont.)

– Conclusion:

- Within a block of adjacent cells, we need to take the common expression
- We need to sum minterms of all blocks.