SNS COLLEGE OF TECHNOLOGY

Coimbatore-35.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 19ITT202 – COMPUTER ORGANIZATION AND ARCHITECTURE

II YEAR/ III SEMESTER

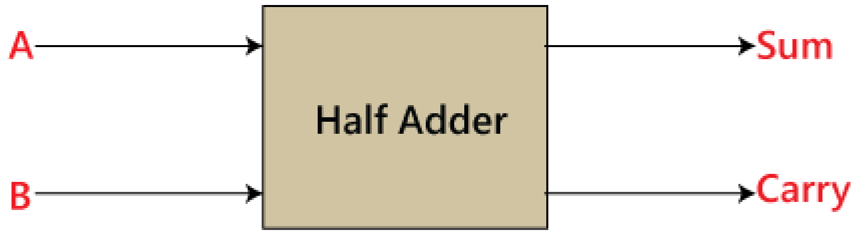
UNIT – II Arithmetic Operations

Topic: Design of Fast Adders

Mrs. G.Vanitha
Assistant Professor
Department of Computer Science and Engineering

HALF ADDER

- A half adder is a combinational circuit that performs addition / subtraction operation for two input values and gives the output SUM and CARRY.
- The carry of previous operation is not carried for next operation.

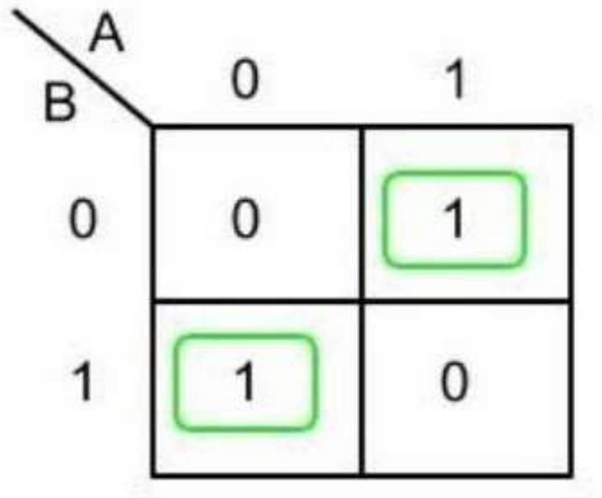


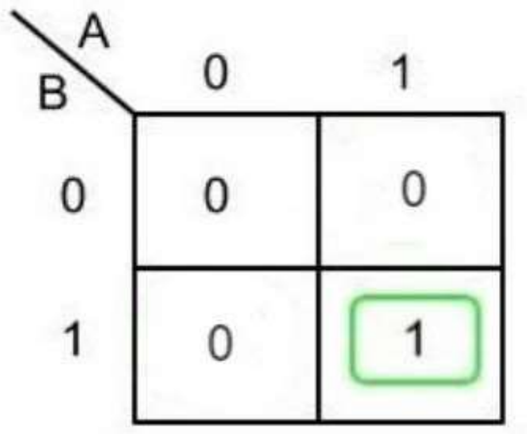
- The addition of 2 bits is done using a combination circuit called a Half adder.
- The input variables are augend and addend bits and output variables are sum & carry bits. A and B are the two input bits.

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = A \oplus B$

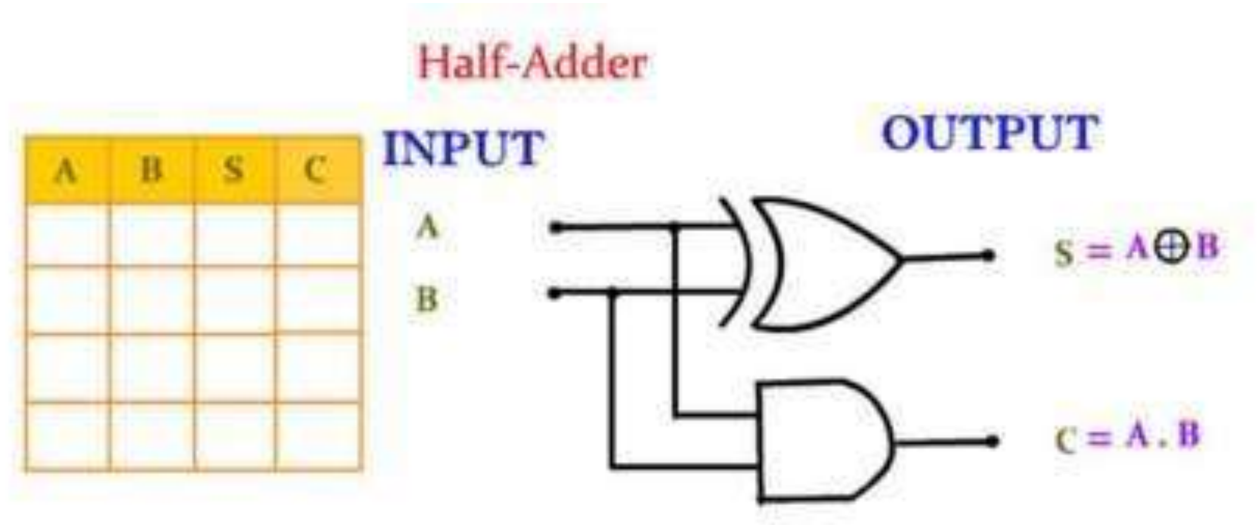
$C = A . B$

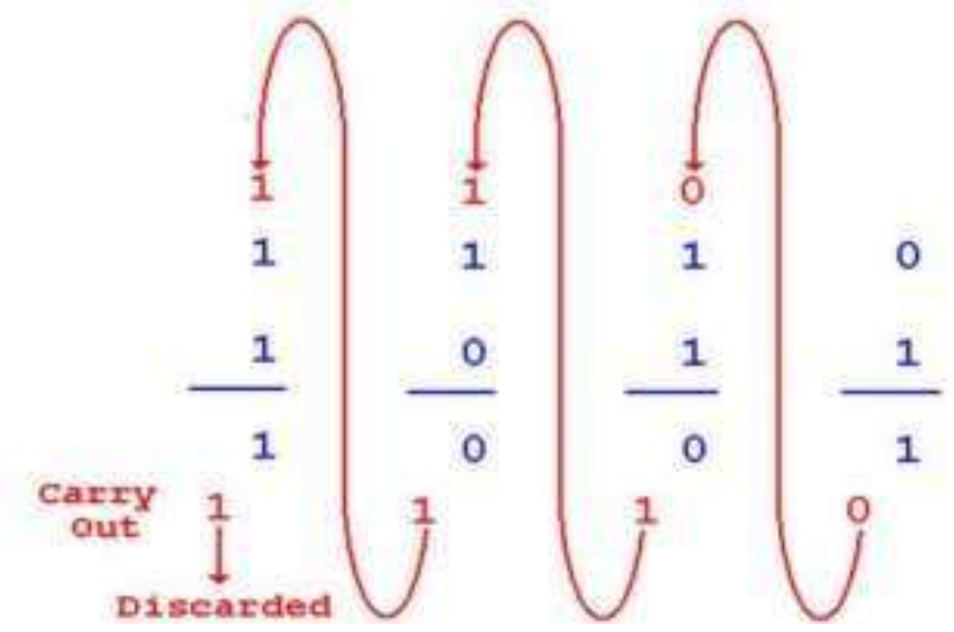




Carry = A AND B

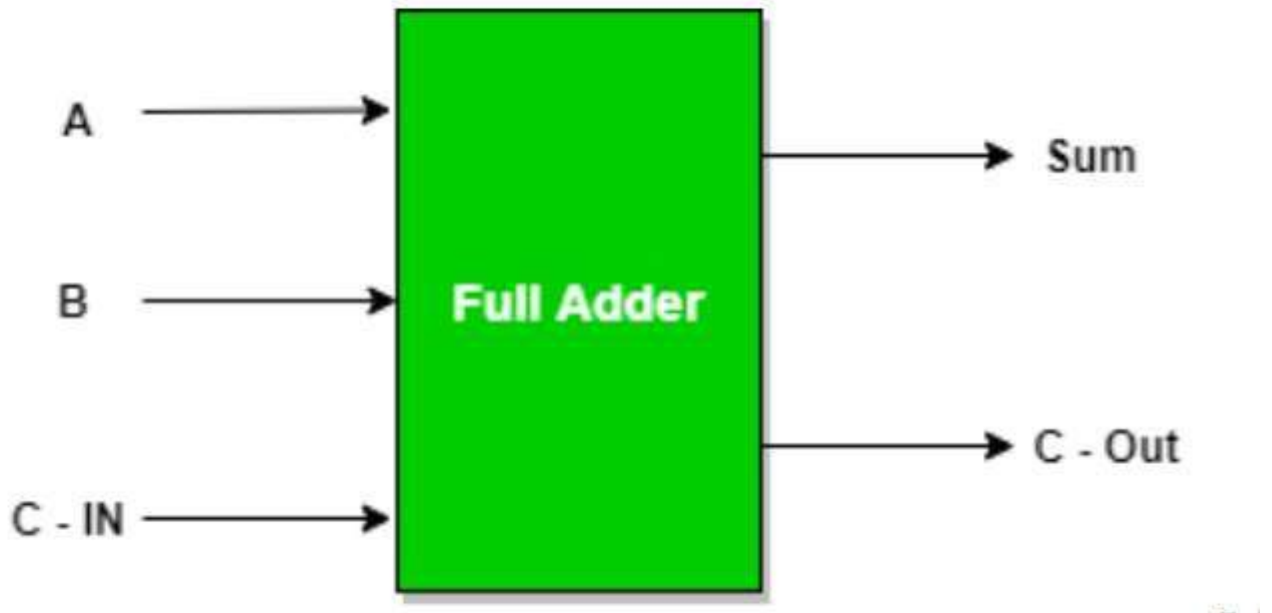
Sum = A XOR B

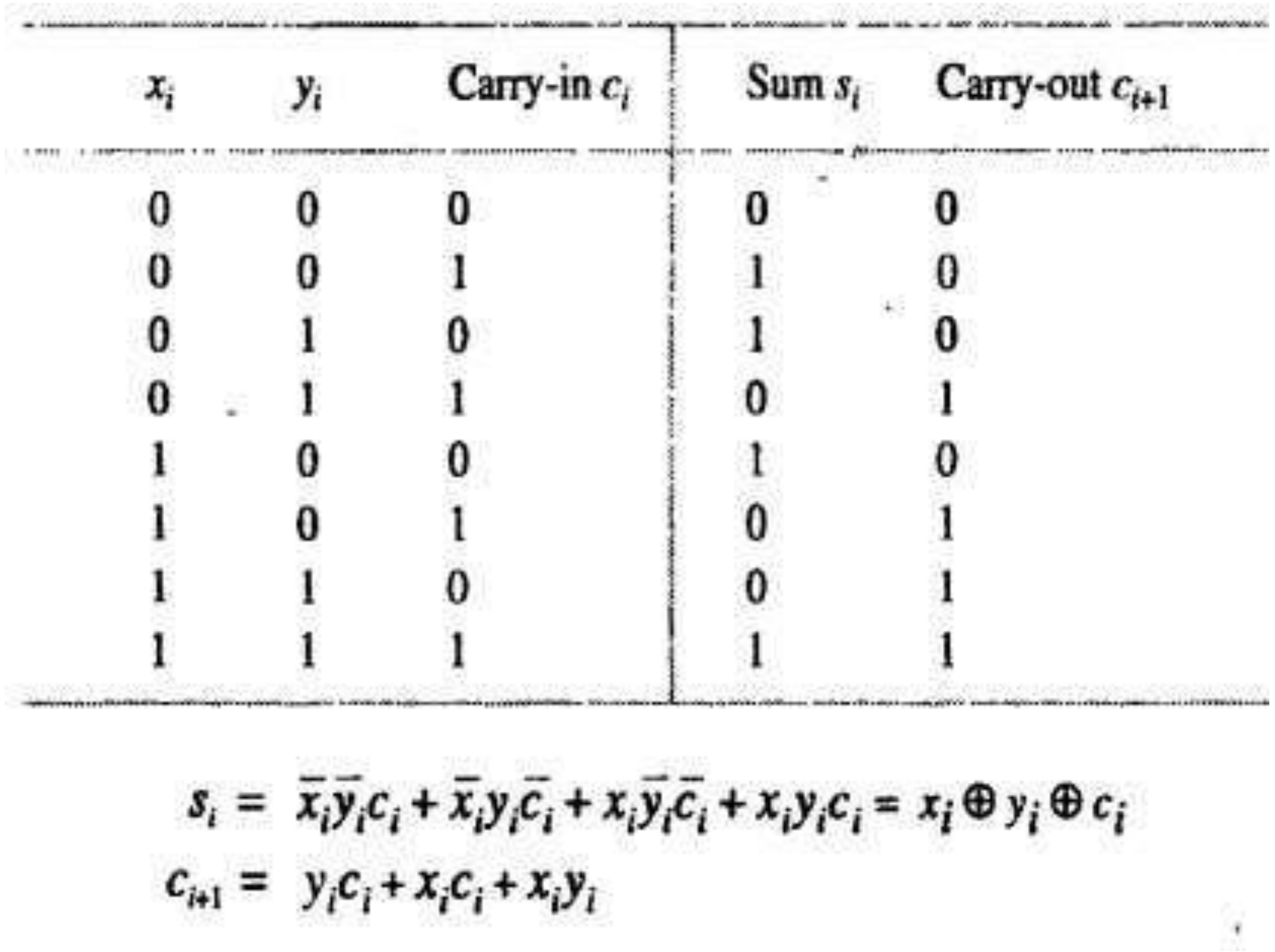

Half-Adder
A B S C
INPUT
OUTPUT
A
B
S = A⊕B
C = A . B


1 1 0
1 1 1 0
1 0 1 1
1 0 0 1
Carry out 1 1 1 0
Discarded

FULL ADDER

- A complete circuit to perform a single stage of addition is called as a full adder (FA).
- It is used to add 3 values.

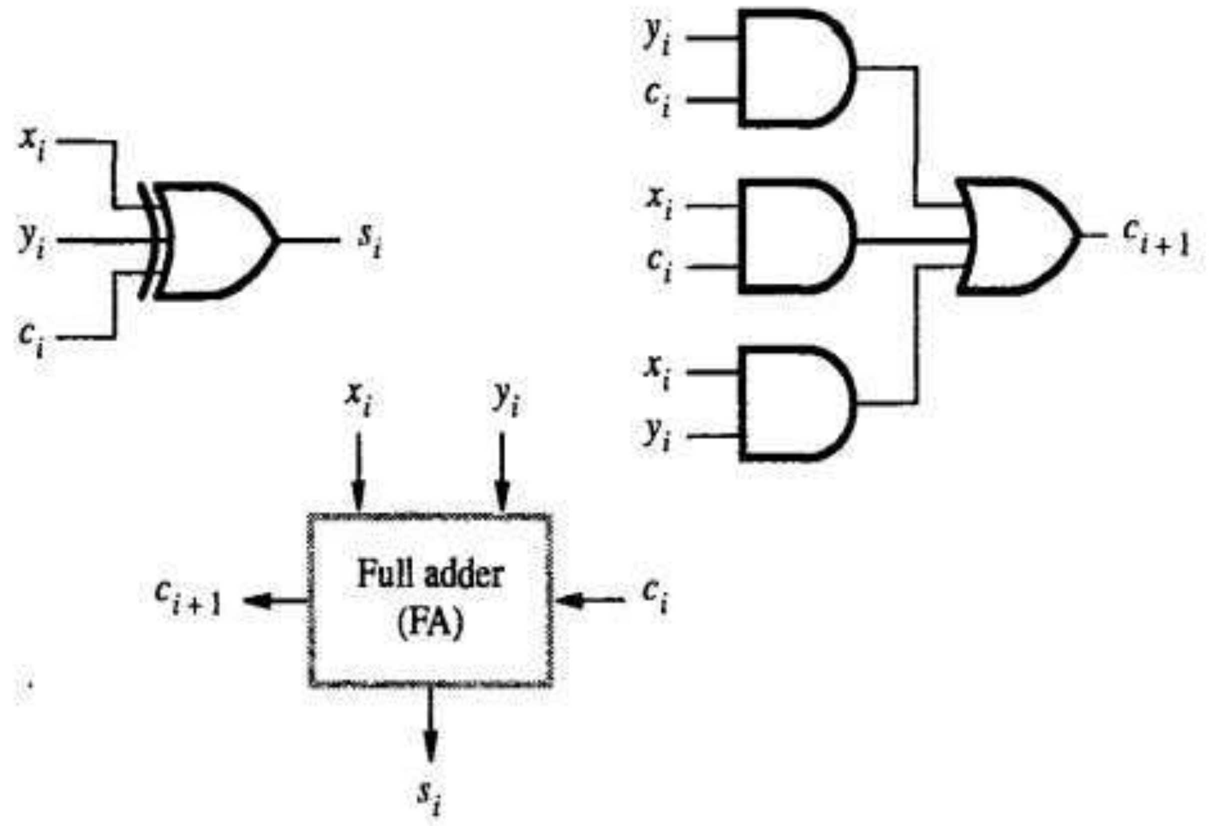




x_i	y_i	Carry-in c_i	Sum s_i	Carry-out c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$s_i = \bar{x}_i\bar{y}_ic_i + \bar{x}_iy_i\bar{c}_i + x_i\bar{y}_i\bar{c}_i + x_iy_ic_i = x_i \oplus y_i \oplus c_i$$

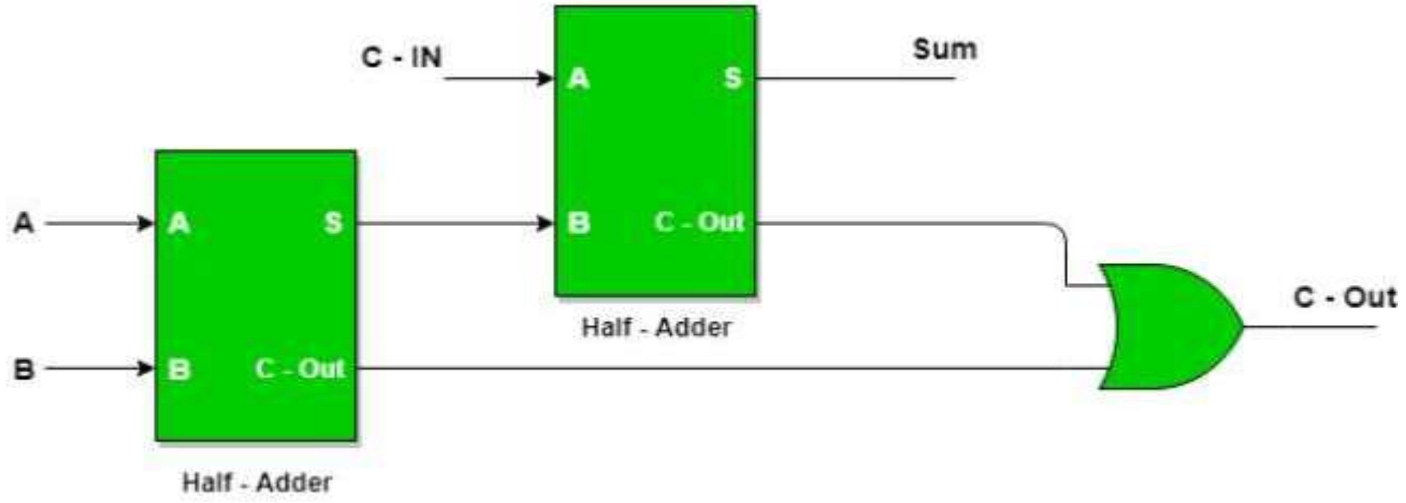
$$c_{i+1} = y_ic_i + x_ic_i + x_iy_i$$

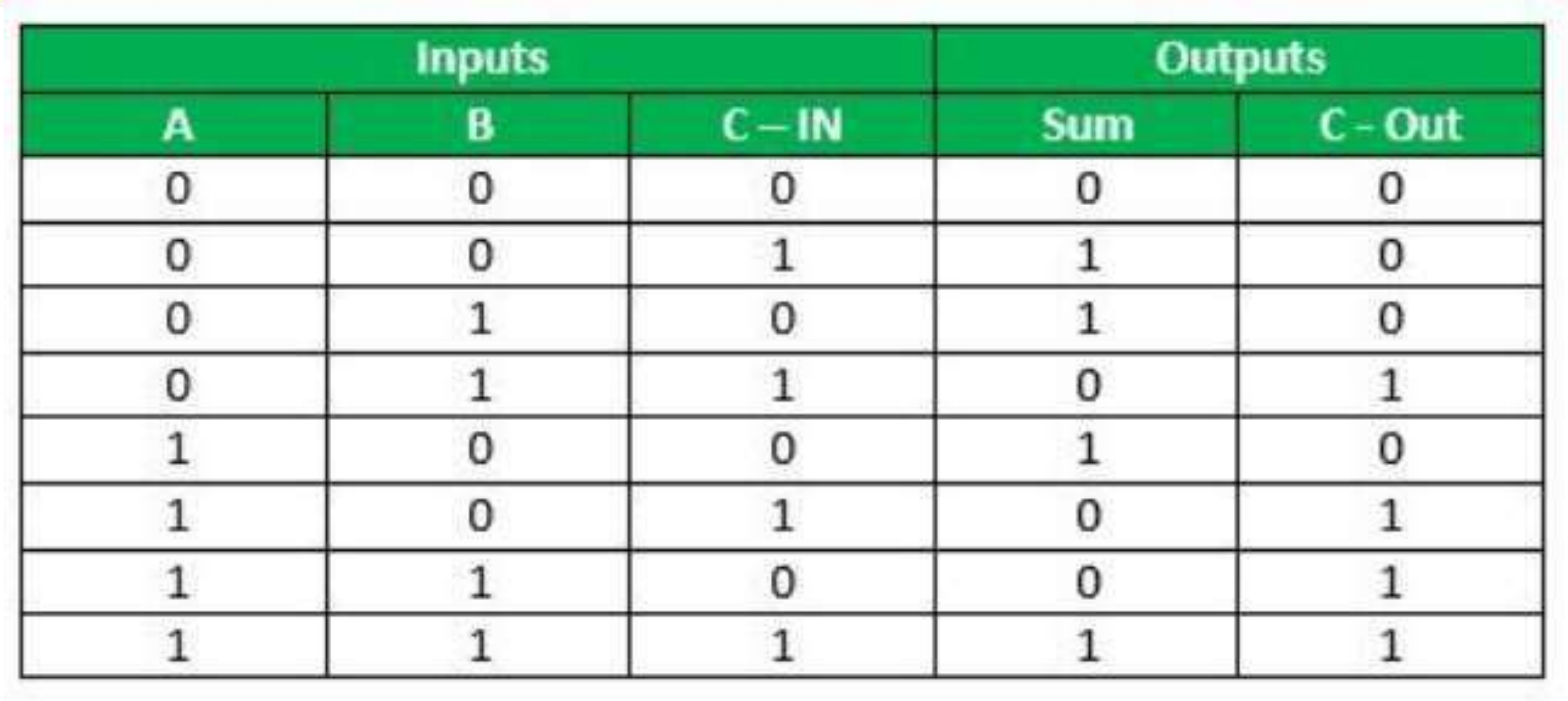


(a) Logic for a single stage

Construction of Full Adder from 2 Half Adder

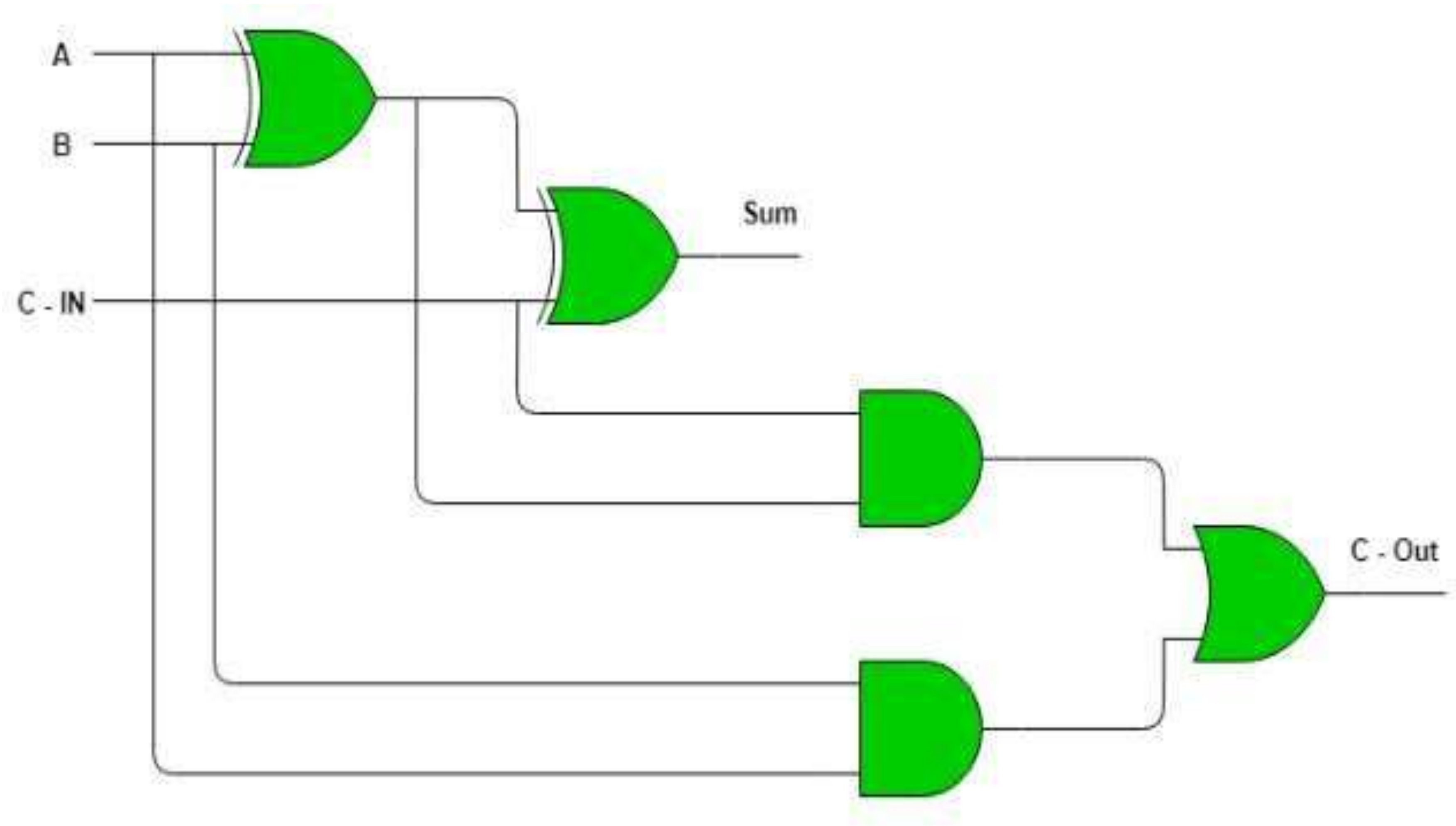
- A full adder can be constructed with the help of two half adder (HA)
- The difference is that the carry of the previous sum can be given as input for the next addition/operation in full adder.



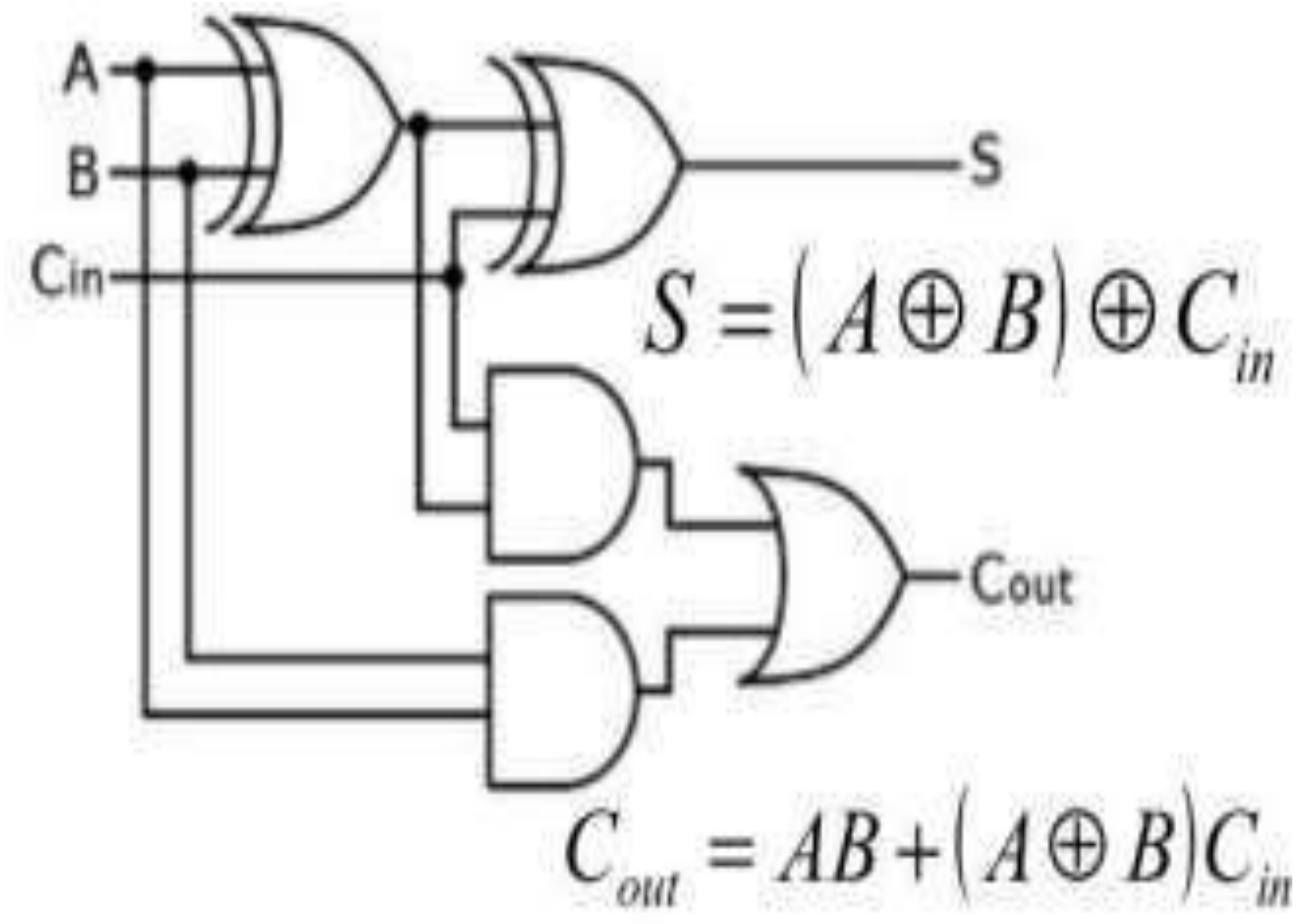


Inputs			Outputs	
A	B	C – IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = (A \oplus B) \oplus C_{in} \qquad C_{out} = AB + (A \oplus B)C_{in}$$



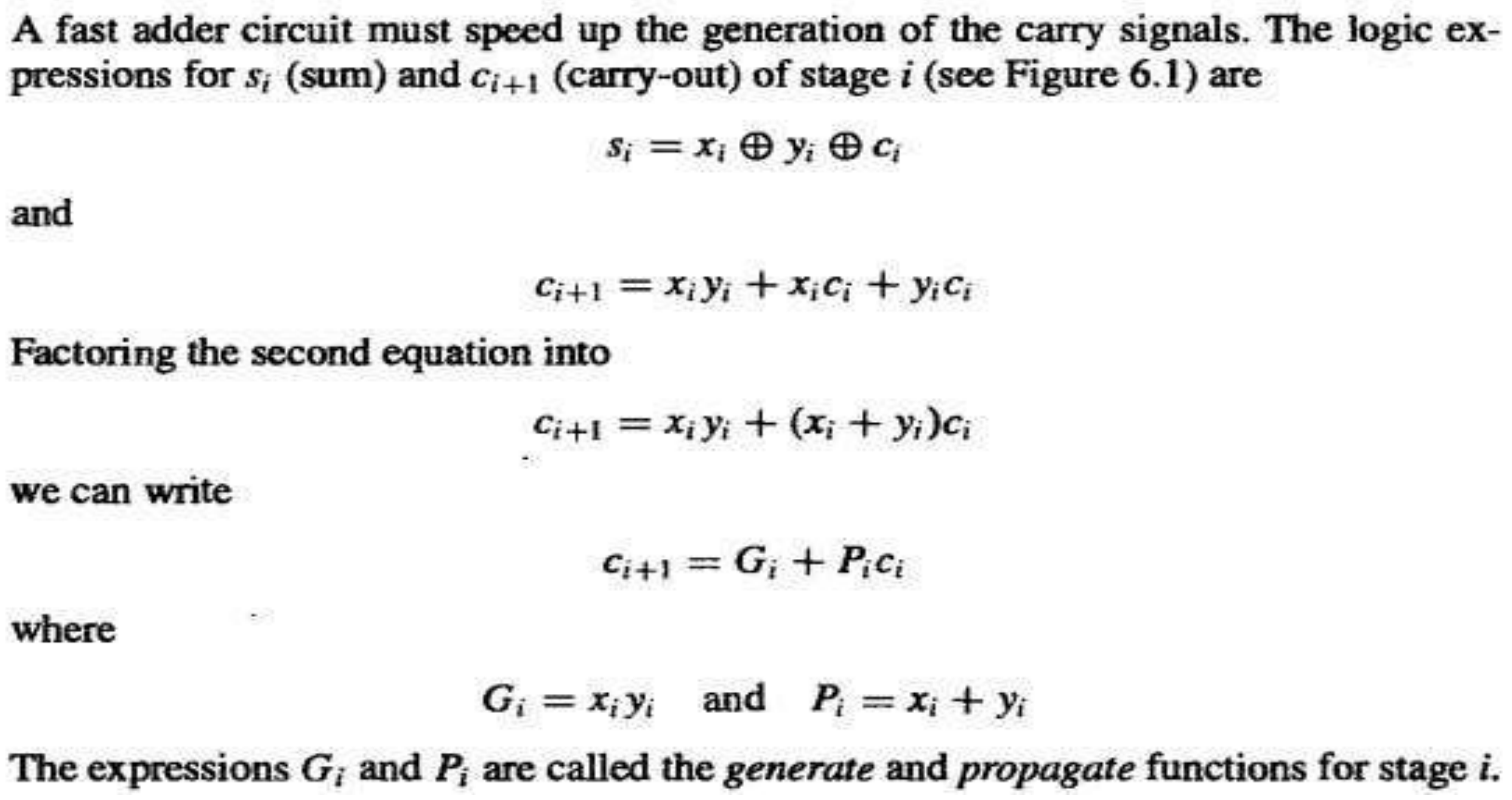
Full Adder logic circuit.



$$S = (A \oplus B) \oplus C_{in}$$

$$C_{out} = AB + (A \oplus B)C_{in}$$

CARRY-LOOK AHEAD ADD



A fast adder circuit must speed up the generation of the carry signals. The logic expressions for s_i (sum) and c_{i+1} (carry-out) of stage i (see Figure 6.1) are

$$s_i = x_i \oplus y_i \oplus c_i$$

and

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

Factoring the second equation into

$$c_{i+1} = x_i y_i + (x_i + y_i) c_i$$

we can write

$$c_{i+1} = G_i + P_i c_i$$

where

$$G_i = x_i y_i \quad \text{and} \quad P_i = x_i + y_i$$

The expressions G_i and P_i are called the *generate* and *propagate* functions for stage i.

Expanding c_i in terms of $i-1$ subscripted variables and substituting into the c_{i+1} expression, we obtain

$$c_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} c_{i-1}$$

Continuing this type of expansion, the final expression for any carry variable is

$$c_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \cdots + P_i P_{i-1} \cdots P_1 G_0 + P_i P_{i-1} \cdots P_0 c_0 \qquad [6.1]$$

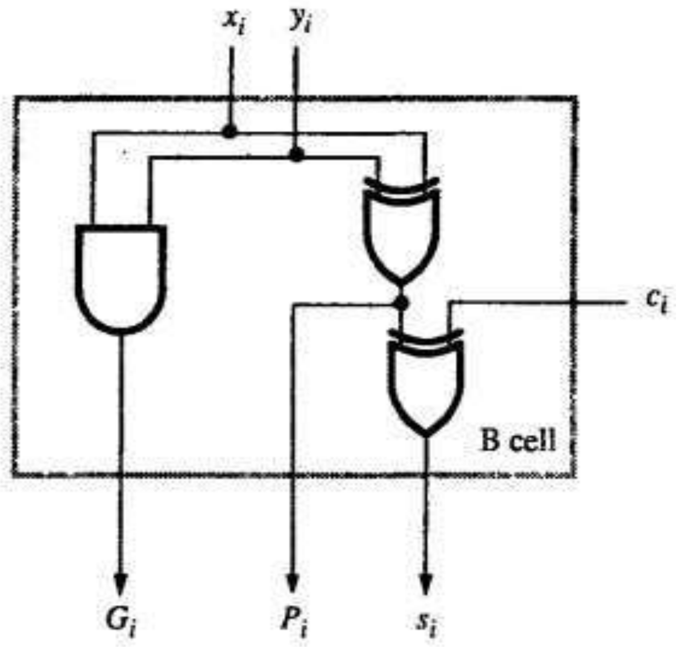
Let us consider the design of a 4-bit adder. The carries can be implemented as

$$c_1 = G_0 + P_0 c_0$$

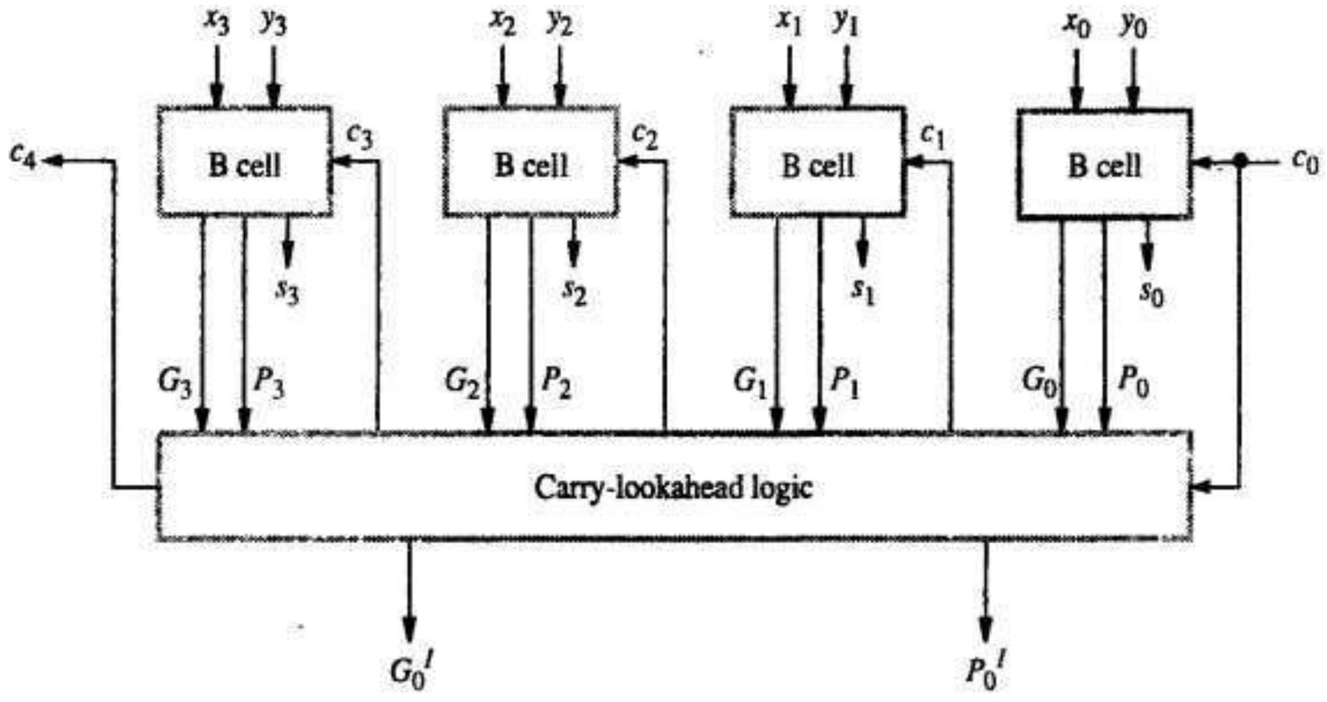
$$c_2 = G_1 + P_1 G_0 + P_1 P_0 c_0$$

$$c_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_0$$

$$c_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0$$



(a) Bit-stage cell



(b) 4-bit adder

Figure 6.4 4-bit carry-lookahead adder.

The complete 4-bit adder is shown in Figure 6.4*b*. The carries are implemented in the block labeled carry-lookahead logic. An adder implemented in this form is called a ***carry-lookahead adder***.

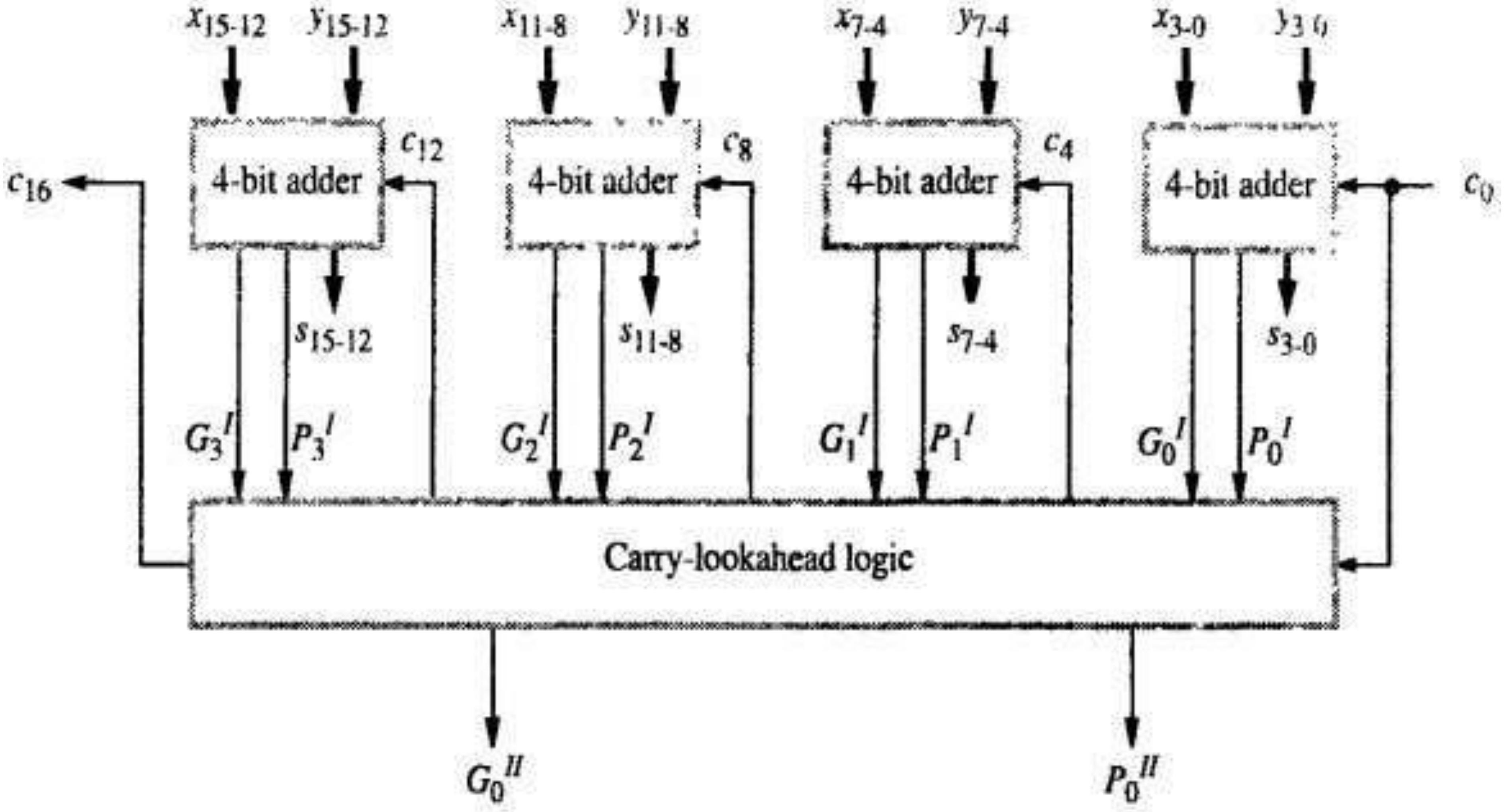


Figure 6.5 16-bit carry-lookahead adder built from 4-bit adders (see Figure 6.4*b*).

THANK
YOU!