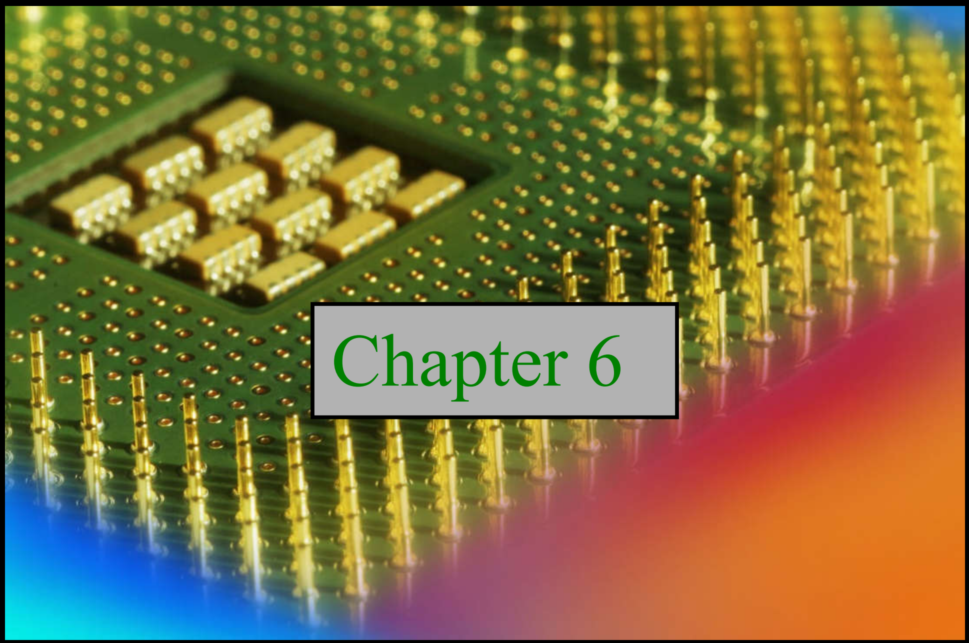
Digital Fundamentals

Tenth Edition

Floyd



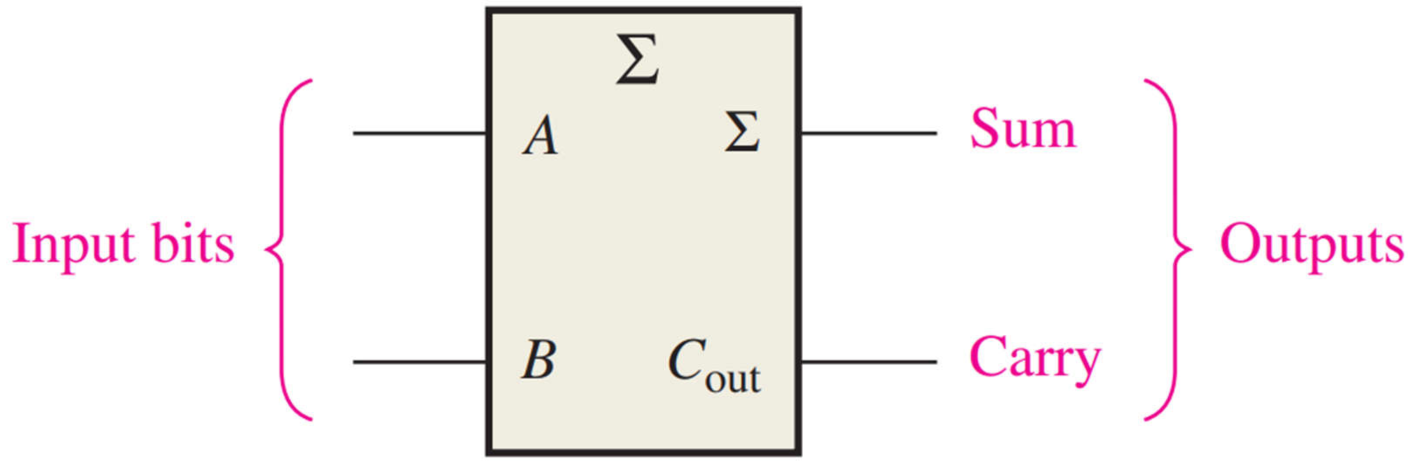
Summary

Half-Adder

Basic rules of binary addition are performed by a **half adder**, which has two binary inputs (A and B) and two binary outputs (Carry out and Sum).

The inputs and outputs can be summarized on a truth table.

Inputs		Outputs	
A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



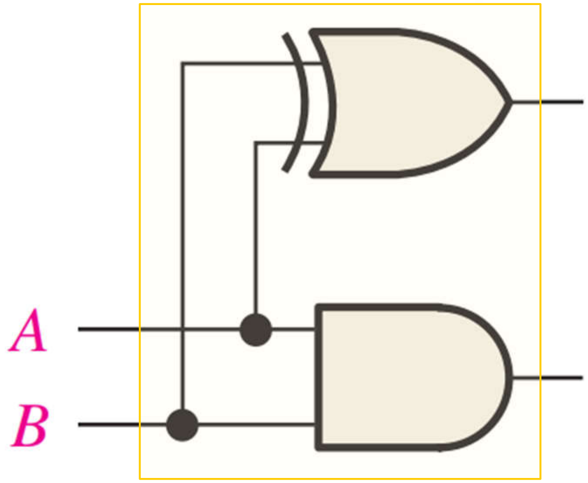
Summary

Half-Adder

The logic symbol and equivalent circuit are:

Half-adder truth table.

A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



$$\Sigma = A \oplus B = A\overline{B} + \overline{A}B$$

$$C_{out} = AB$$

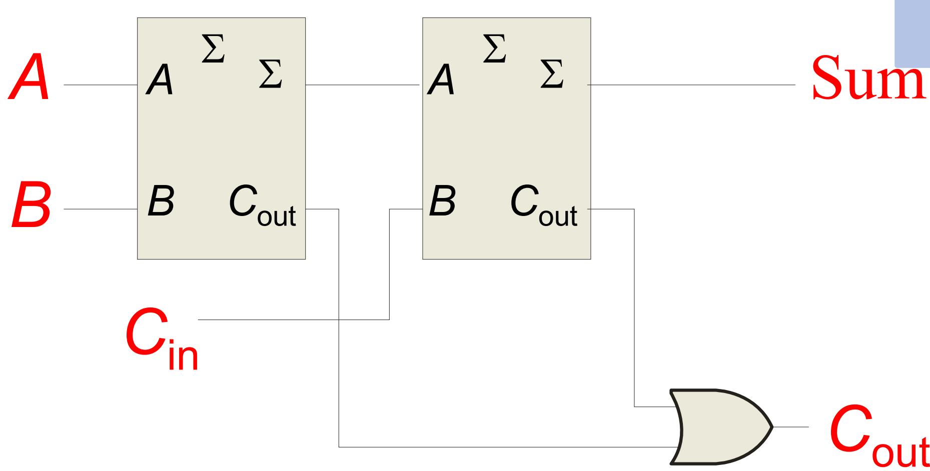
Summary

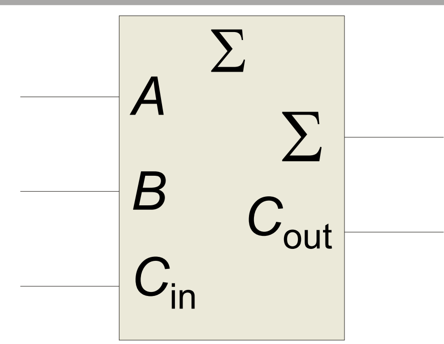
Full-Adder

By contrast, a **full adder** has three binary inputs (*A*, *B,* and Carry in) and two binary outputs (Carry out and Sum). The truth table summarizes the operation.

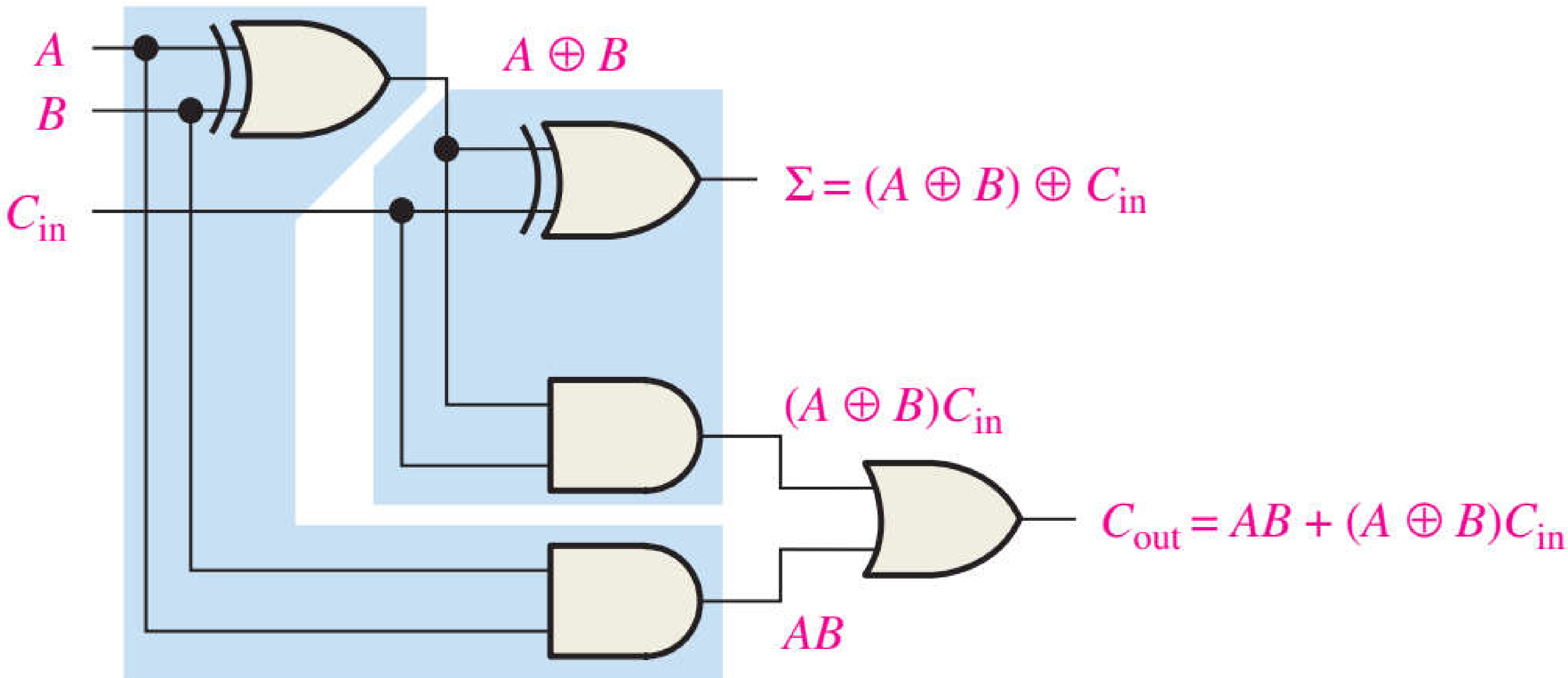
Inputs			Outputs	
A	B	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

A full-adder can be constructed from two half adders as shown:





Full-Adder



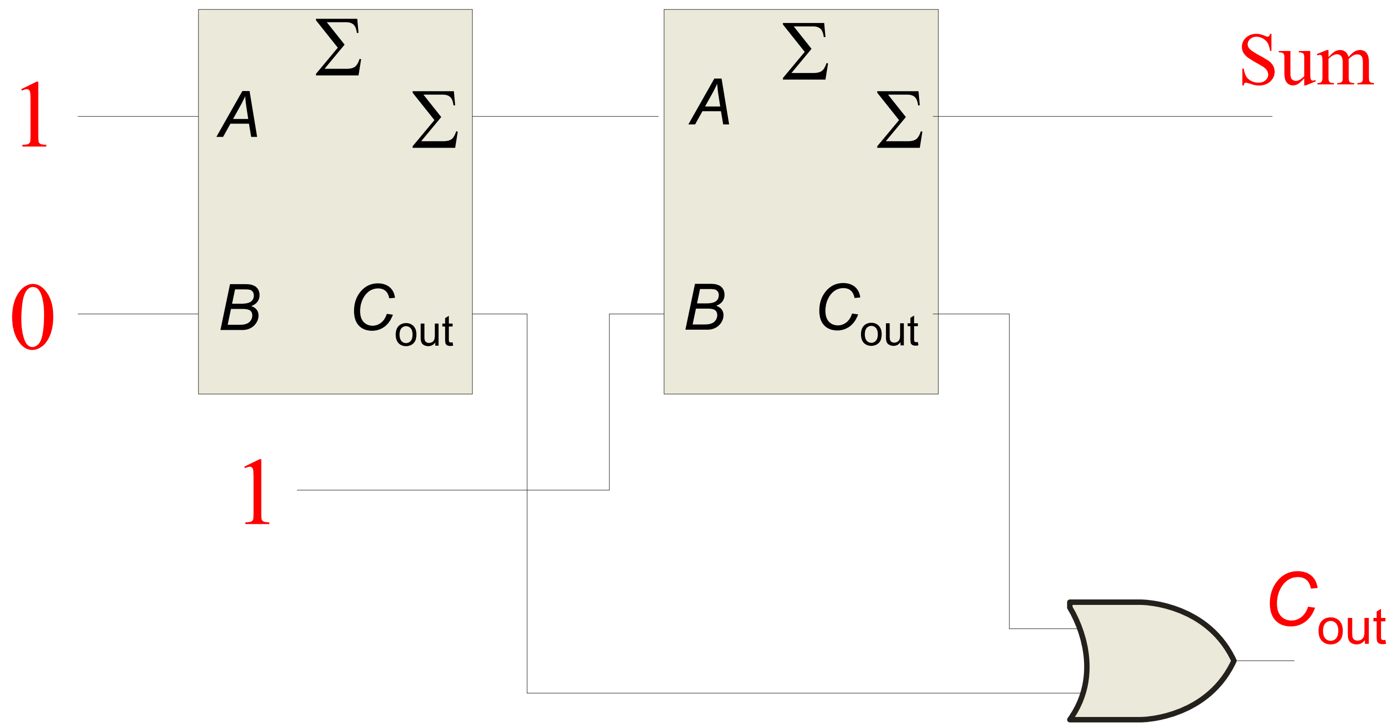
A
B
C_{in}
$A \oplus B$
$\Sigma = (A \oplus B) \oplus C_{in}$
$(A \oplus B)C_{in}$
$C_{out} = AB + (A \oplus B)C_{in}$
AB

Summary

Full-Adder

Example

For the given inputs, determine the intermediate and final outputs of the full adder.

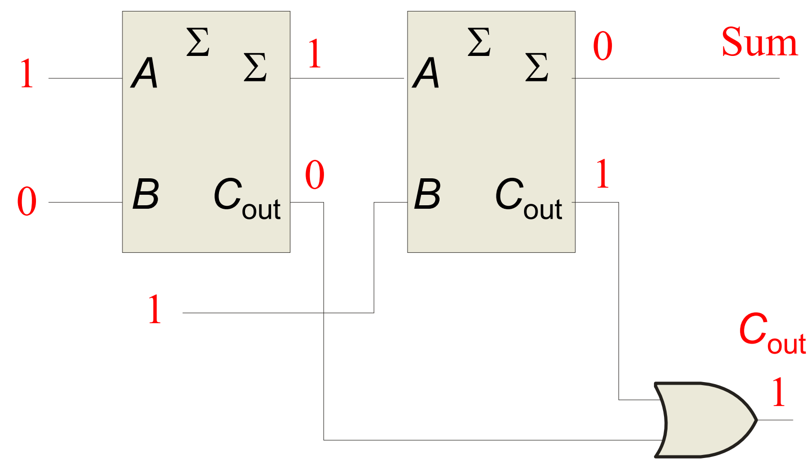


Summary

Full-Adder

Notice that the result from the previous example can be read directly on the truth table for a full adder.

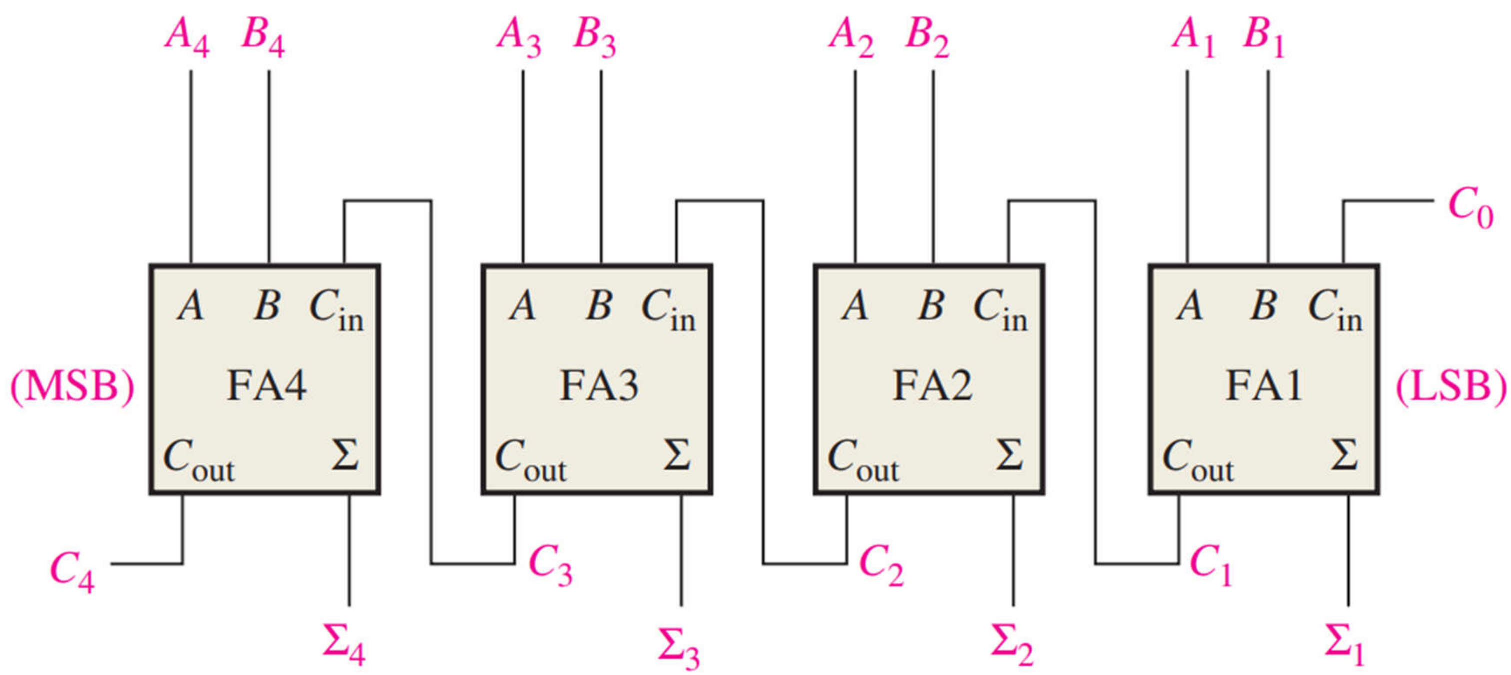
Inputs			Outputs	
A	B	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Summary

Parallel Adders

Full adders are combined into parallel adders that can add binary numbers with multiple bits. A 4-bit adder is shown.

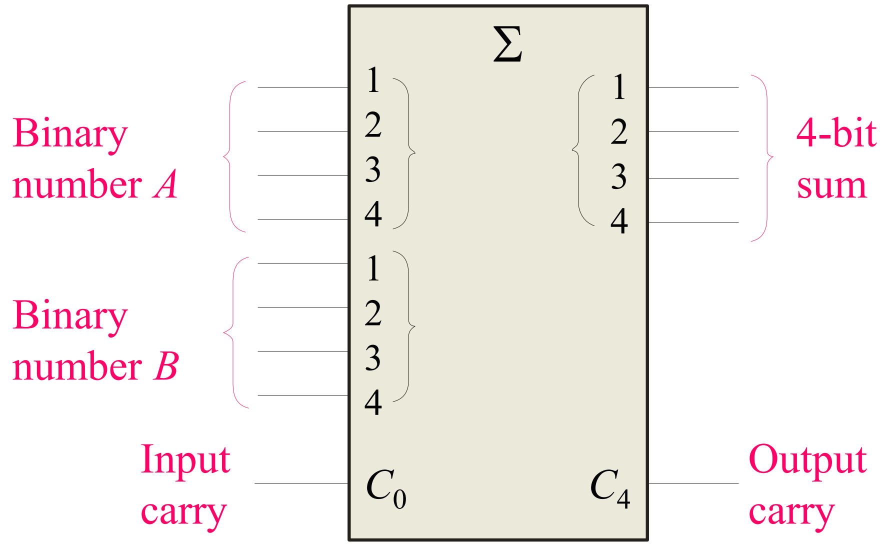


The output carry (C_4) is not ready until it propagates through all of the full adders. This is called *ripple carry*, delaying the addition process.

Summary

Parallel Adders

The logic symbol for a 4-bit parallel adder is shown. This 4-bit adder includes a carry in (labeled (C_0) and a Carry out (labeled C_4).

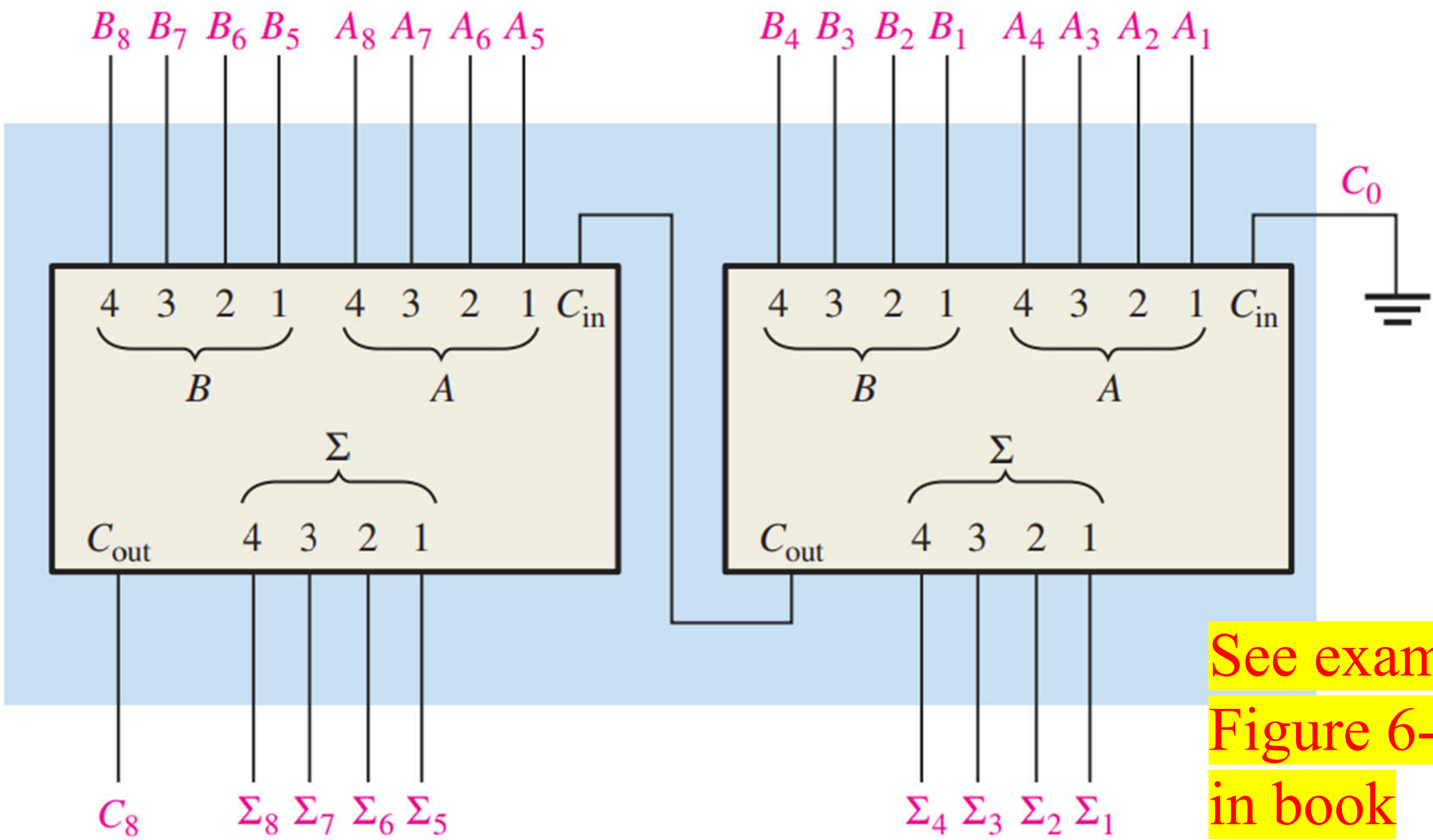


The 74LS283 is an example. It features *look-ahead carry*, which adds logic to minimize the output carry delay. For the 74LS283, the maximum delay to the output carry is 17 ns.

Summary

Adder Expansion

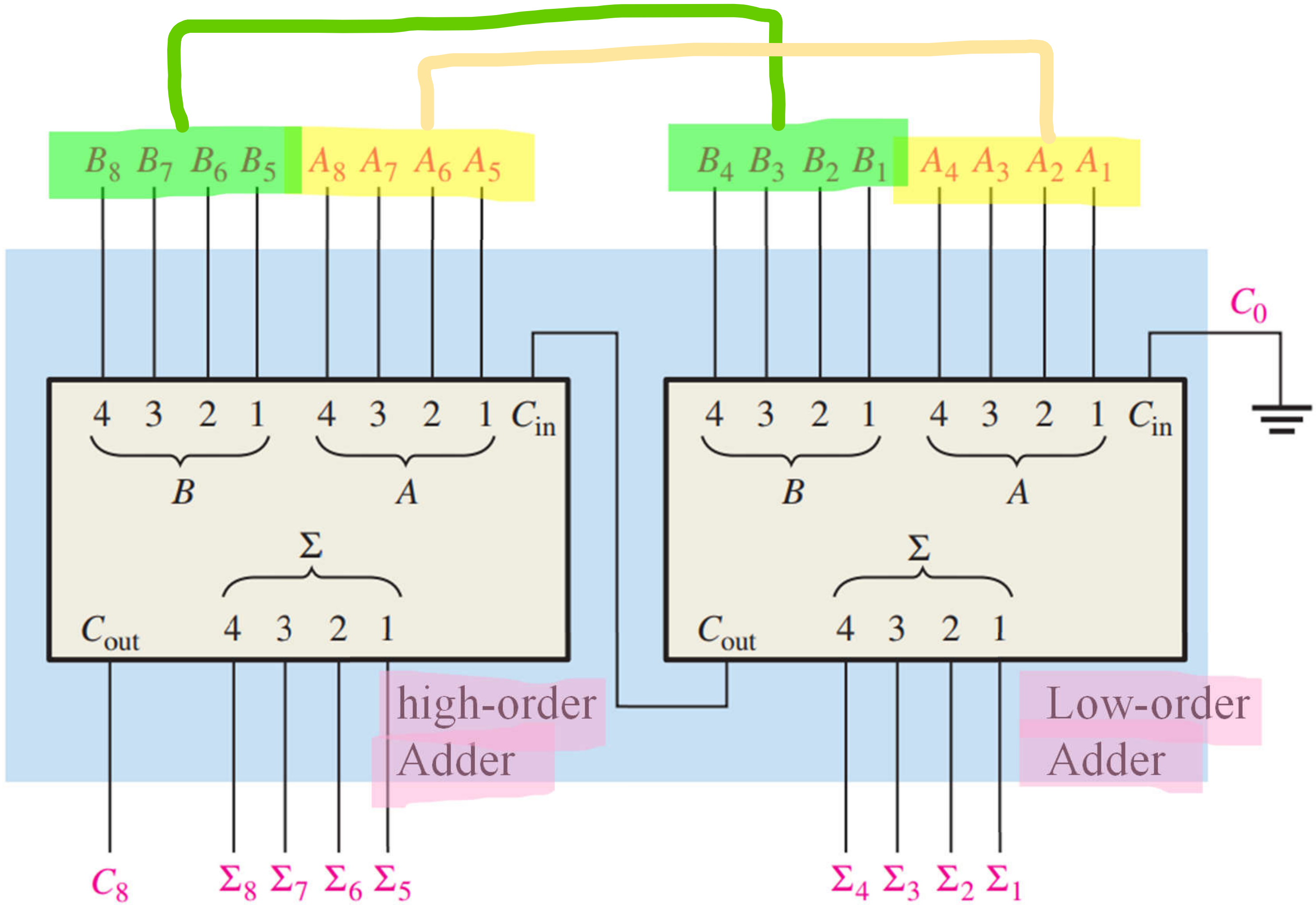
Adders can be further extended by the process of **Cascading**.



See example 6-4
Figure 6-13
in book

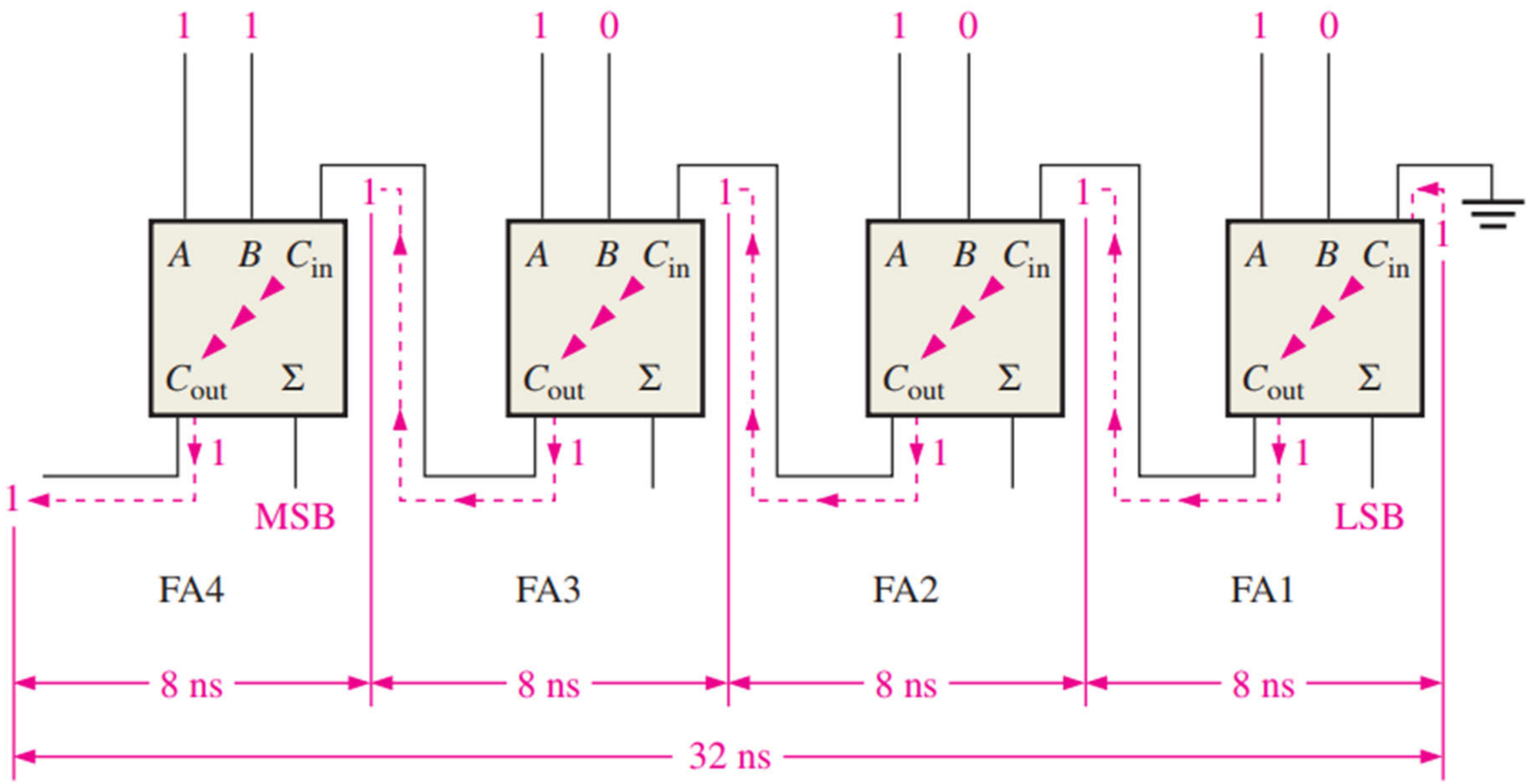
Summary

This is a two 8-bit adder.



The Ripple Carry Adder

A **ripple carry** adder is one in which the carry output of each full-adder is connected to the carry input of the next higher-order stage (a stage is one full-adder). The sum and the output carry of any stage cannot be produced until the input carry occurs



The Look-Ahead Carry Adder

One method of speeding up the addition process by eliminating this ripple carry delay is called **look-ahead carry** addition. The look-ahead carry adder anticipates the output carry of each stage, and based on the inputs, produces the output carry by either carry generation or carry propagation.

Carry generation occurs when an output carry is produced (generated) internally by the full-adder. A carry is generated only when both input bits are 1s. The generated carry, C_g, is expressed as the AND function of the two input bits, A and B.

$$C_g = AB \qquad \text{Equation 6–5}$$

Carry propagation occurs when the input carry is rippled to become the output carry. An input carry may be propagated by the full-adder when either or both of the input bits are 1s. The propagated carry, C_p, is expressed as the OR function of the input bits.

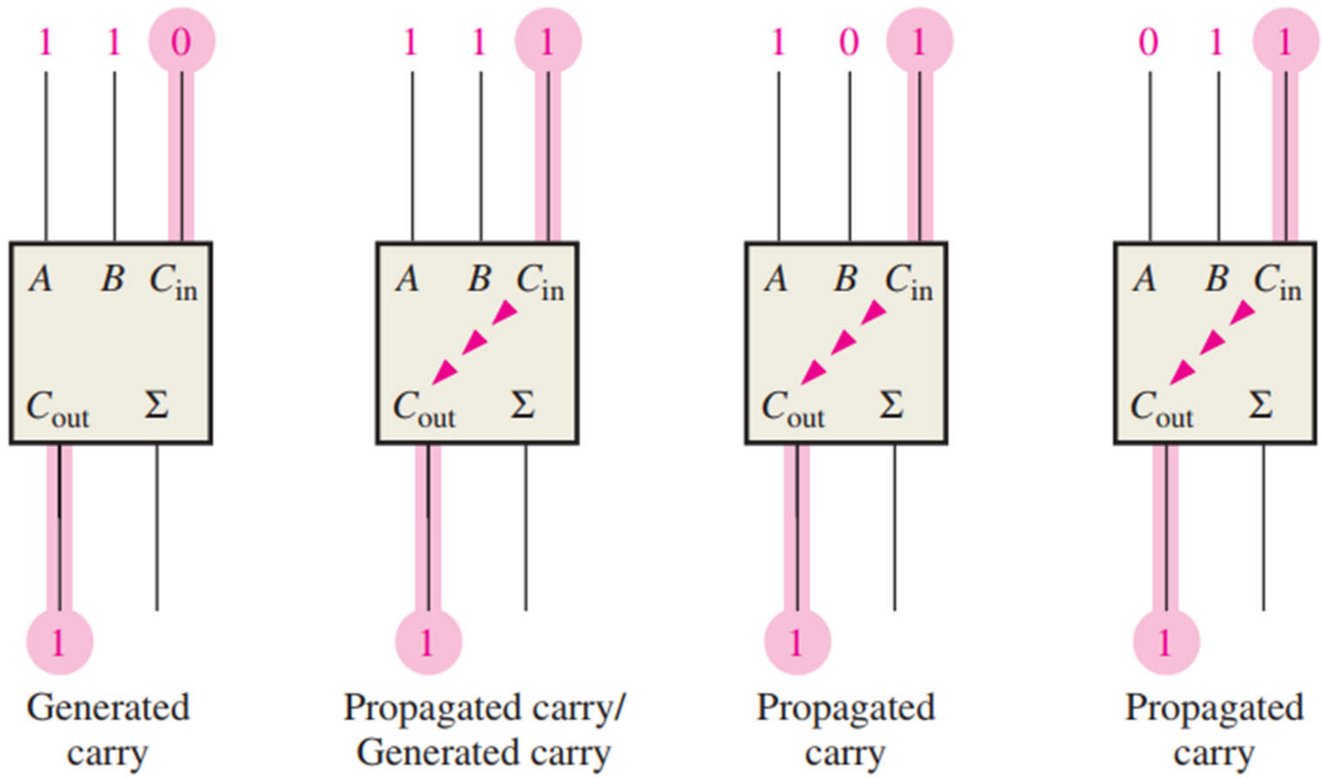
$$C_p = A + B \qquad \text{Equation 6–6}$$

The Look-Ahead Carry Adder

$$C_g = AB$$

$$C_p = A + B$$

Example conditions for the generated and propagation carries

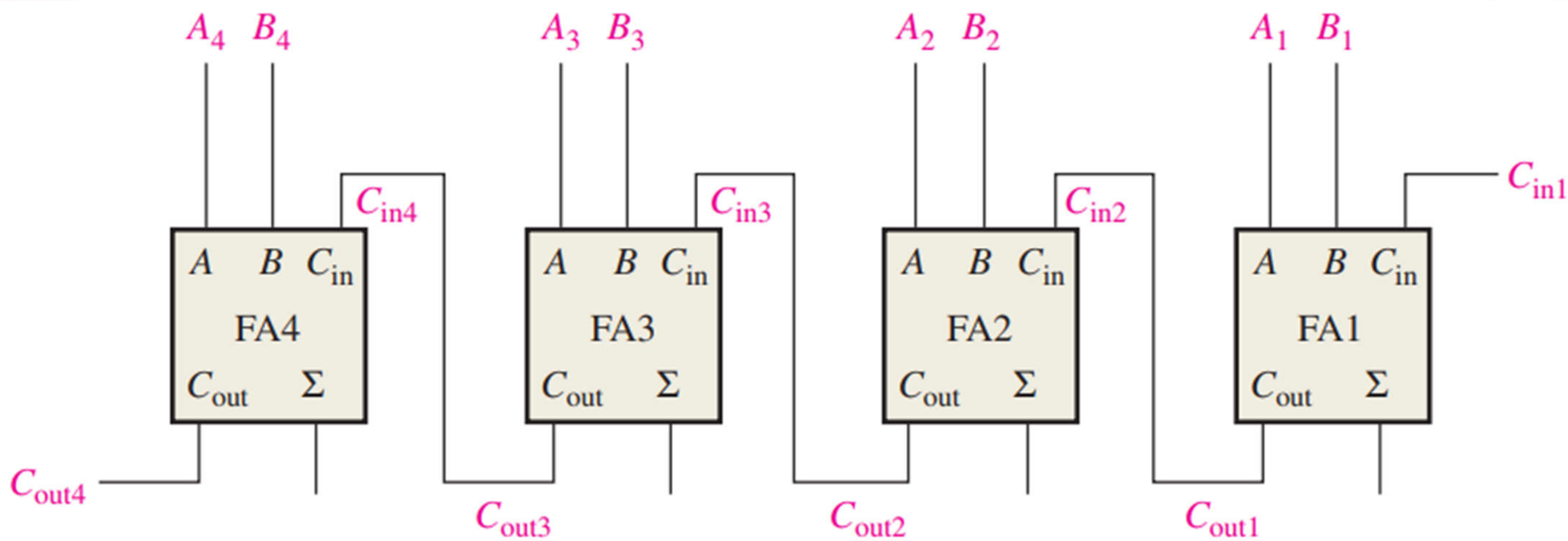


$$C_{out} = C_g + C_pC_{in}$$

The Look-Ahead Carry Adder

$C_g = AB$ | $C_p = A + B$ | $C_{out} = C_g + C_pC_{in}$

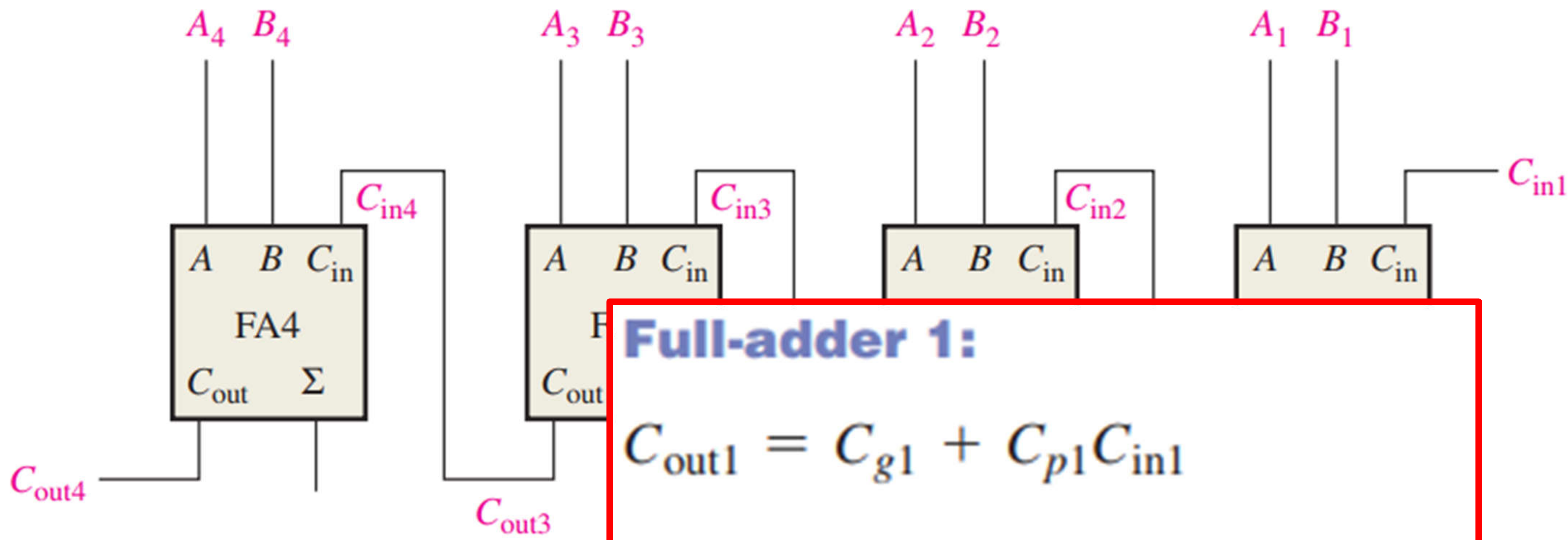
Let us create a Look-Ahead logic circuit for the following example.



Full-adder 4	Full-adder 3	Full-adder 2	Full-adder 1
$C_{g4} = A_4B_4$	$C_{g3} = A_3B_3$	$C_{g2} = A_2B_2$	$C_{g1} = A_1B_1$
$C_{p4} = A_4 + B_4$	$C_{p3} = A_3 + B_3$	$C_{p2} = A_2 + B_2$	$C_{p1} = A_1 + B_1$

The Look-Ahead Carry Adder

$C_g = AB$ $C_p = A + B$ $C_{out} = C_g + C_pC_{in}$



Full-adder 1:

$$C_{out1} = C_{g1} + C_{p1}C_{in1}$$

Full-adder 4

$C_{g4} = A_4B_4$
$C_{p4} = A_4 + B_4$

Full-adder 3

$C_{g3} = A_3B_3$
$C_{p3} = A_3 + B_3$

Full-adder 2

$C_{g2} = A_2B_2$
$C_{p2} = A_2 + B_2$

Full-adder 1

$C_{g1} = A_1B_1$
$C_{p1} = A_1 + B_1$

The Look-Ahead Carry Adder

$$C_g = AB$$

$$C_p = A + B$$

$$C_{out} = C_g + C_pC_{in}$$

Full-adder 2:

$$C_{in2} = C_{out1}$$

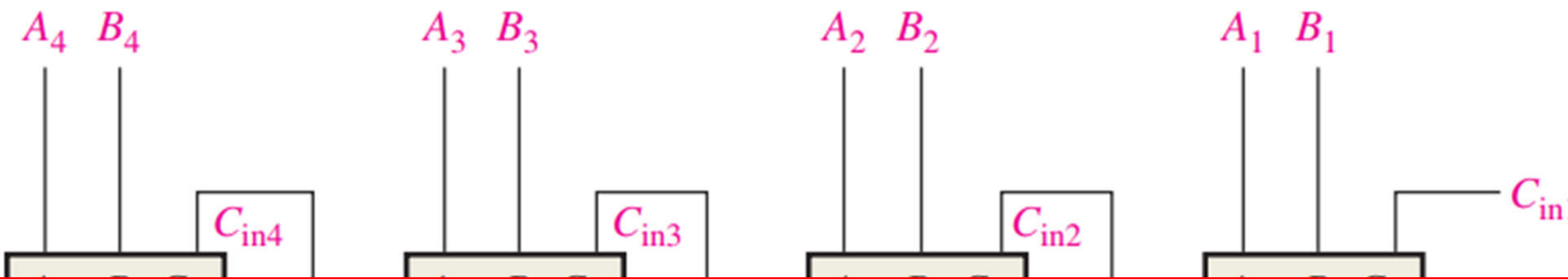
$$C_{out2} = C_{g2} + C_{p2}C_{in2} = C_{g2} + C_{p2}C_{out1} = C_{g2} + C_{p2}(C_{g1} + C_{p1}C_{in1})$$

$$= C_{g2} + C_{p2}C_{g1} + C_{p2}C_{p1}C_{in1}$$

Full-adder 4	Full-adder 3	Full-adder 2	Full-adder 1
$C_{g4} = A_4B_4$	$C_{g3} = A_3B_3$	$C_{g2} = A_2B_2$	$C_{g1} = A_1B_1$
$C_{p4} = A_4 + B_4$	$C_{p3} = A_3 + B_3$	$C_{p2} = A_2 + B_2$	$C_{p1} = A_1 + B_1$

The Look-Ahead Carry Adder

$$C_g = AB \qquad C_p = A + B \qquad C_{out} = C_g + C_p C_{in}$$



Full-adder 3:

$$C_{in3} = C_{out2}$$

$$C_{out3} = C_{g3} + C_{p3}C_{in3} = C_{g3} + C_{p3}C_{out2} = C_{g3} + C_{p3}(C_{g2} + C_{p2}C_{g1} + C_{p2}C_{p1}C_{in1})$$

$$= C_{g3} + C_{p3}C_{g2} + C_{p3}C_{p2}C_{g1} + C_{p3}C_{p2}C_{p1}C_{in1}$$

Full-adder 4	Full-adder 3	Full-adder 2	Full-adder 1
$C_{g4} = A_4 B_4$	$C_{g3} = A_3 B_3$	$C_{g2} = A_2 B_2$	$C_{g1} = A_1 B_1$
$C_{p4} = A_4 + B_4$	$C_{p3} = A_3 + B_3$	$C_{p2} = A_2 + B_2$	$C_{p1} = A_1 + B_1$

The Look-Ahead Carry Adder

$\boxed{C_g = AB}$ $\boxed{C_p = A + B}$ $\boxed{C_{out} = C_g + C_pC_{in}}$

A_4 B_4 A_3 B_3 A_2 B_2 A_1 B_1

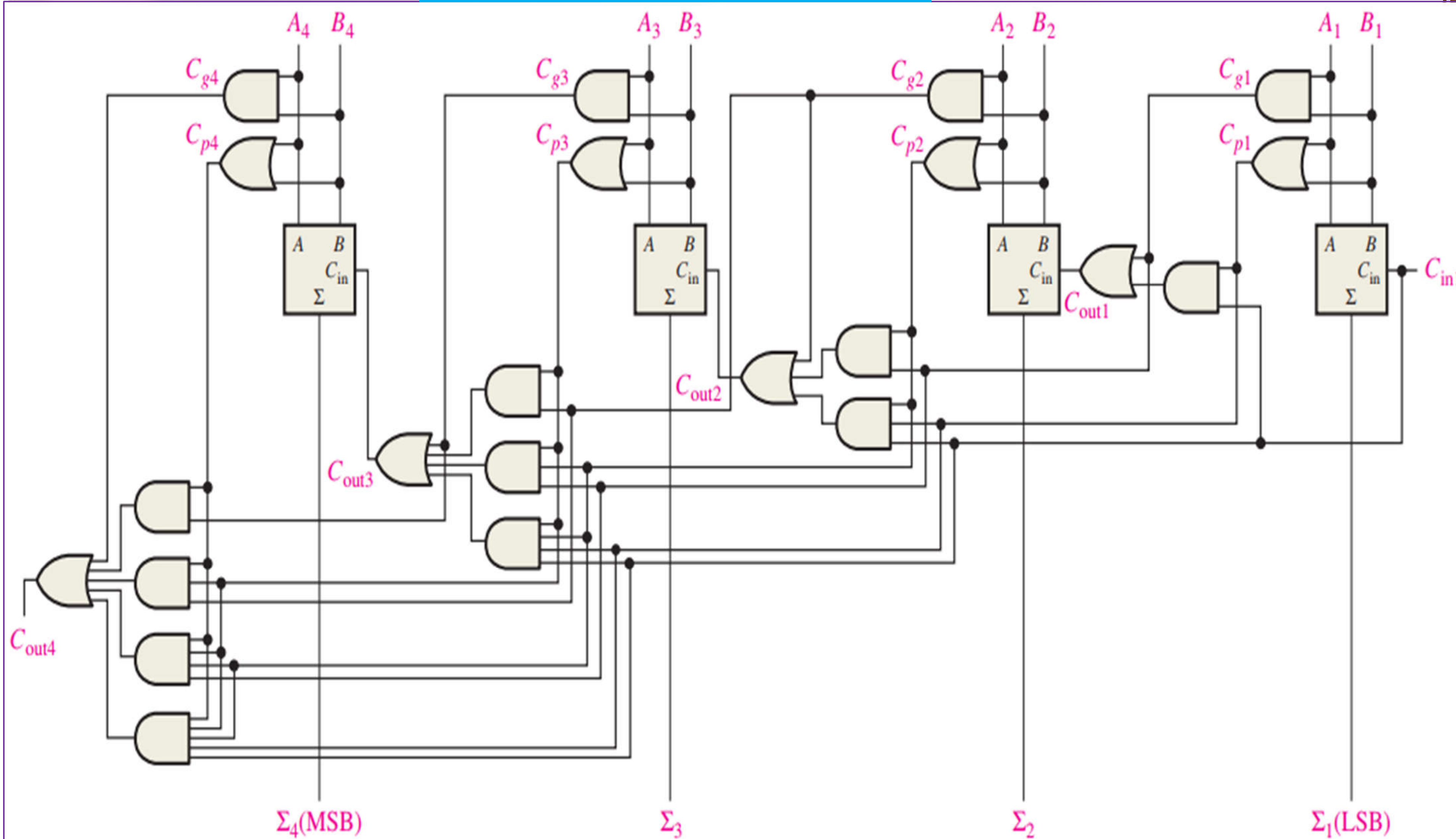
Full-adder 4:

$$C_{in4} = C_{out3}$$

$$\begin{aligned}
C_{out4} &= C_{g4} + C_{p4}C_{in4} = C_{g4} + C_{p4}C_{out3} \\
&= C_{g4} + C_{p4}(C_{g3} + C_{p3}C_{g2} + C_{p3}C_{p2}C_{g1} + C_{p3}C_{p2}C_{p1}C_{in1}) \\
&= C_{g4} + C_{p4}C_{g3} + C_{p4}C_{p3}C_{g2} + C_{p4}C_{p3}C_{p2}C_{g1} + C_{p4}C_{p3}C_{p2}C_{p1}C_{in1}
\end{aligned}$$

Full-adder 4	Full-adder 3	Full-adder 2	Full-adder 1
$C_{g4} = A_4B_4$	$C_{g3} = A_3B_3$	$C_{g2} = A_2B_2$	$C_{g1} = A_1B_1$
$C_{p4} = A_4 + B_4$	$C_{p3} = A_3 + B_3$	$C_{p2} = A_2 + B_2$	$C_{p1} = A_1 + B_1$

4-bit Look-Ahead Full Adder



$$= C_{g4} + C_{p4}C_{g3} + C_{p4}C_{p3}C_{g2} + C_{p4}C_{p3}C_{p2}C_{g1} + C_{p4}C_{p3}C_{p2}C_{p1}C_{in1}$$