Digital circuits

Combinational circuit

Output depends only on present inputs

- Half and Full Adders
- Comparator
- Decoder
- Encoder
- Code Converter
- Multiplexer(Data Selector)
- De multiplexers

Sequential circuit

Output depends upon present inputs as well as previous output

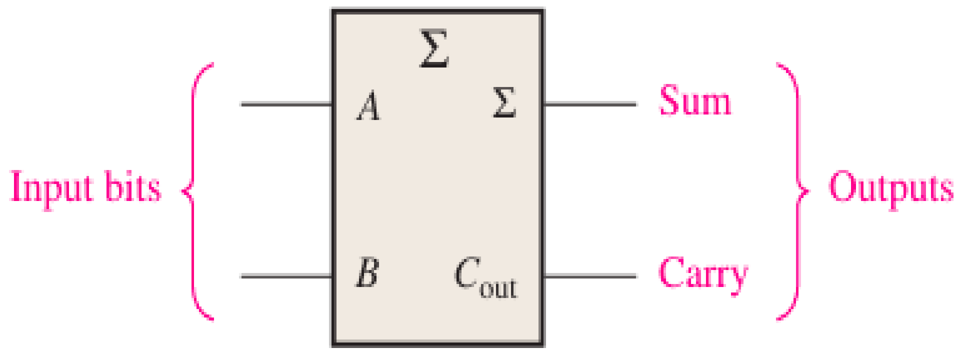
- Latches
- Flip-Flops

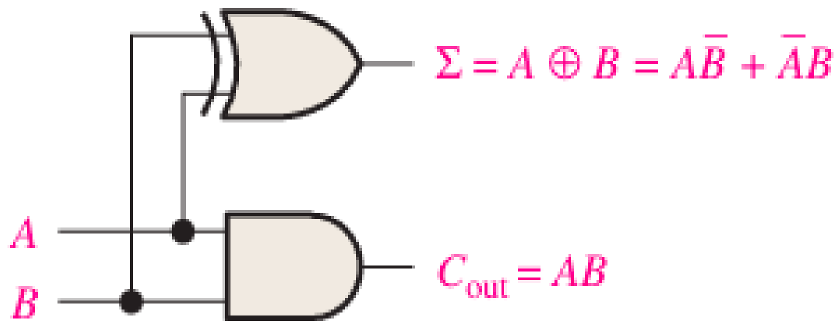
The Half-Adder

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	10

A half-adder adds two bits and produces a sum and an output carry.

The half-adder accepts two binary digits on its inputs and produces two binary digits on its outputs—a sum bit and a carry bit.



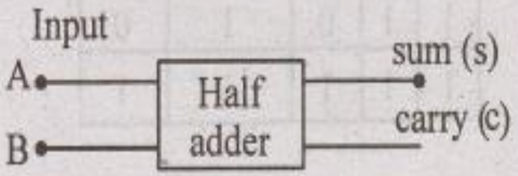


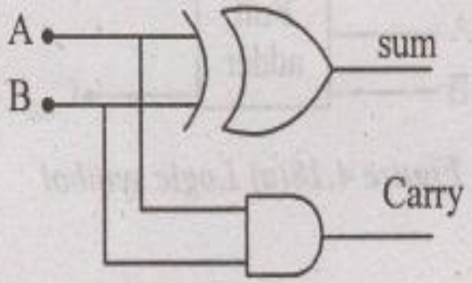
Half Adder

Digital computers and calculators consist of arithmetic and logic circuits. The basic building blocks of the arithmetic unit in a digital computer are adders.

The simplest combinational circuit that performs the arithmetic addition of two binary digits is called a half-adder. A half adder has two inputs and two outputs.

Input		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

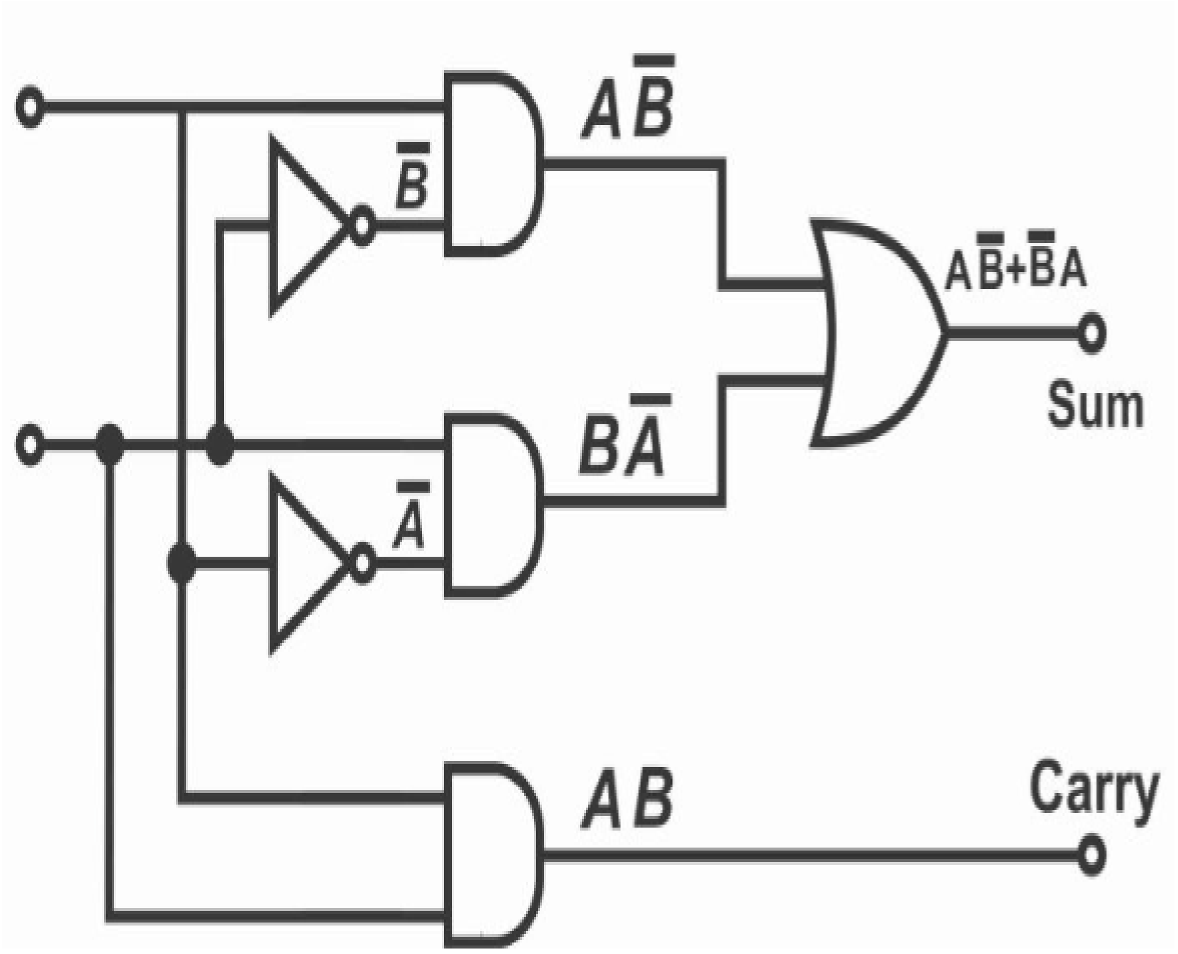




$$S = \bar{A}B + A\bar{B}$$

This expression can be simplified as

$$S = A \oplus B \qquad C = AB$$



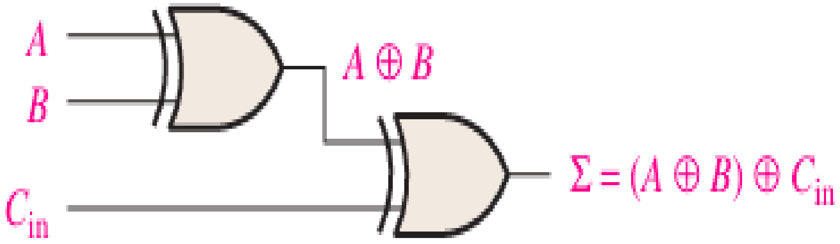
Full Adder

A half-adder has only two inputs, and there is no provision to add a carry coming from the lower-order bits when multi-bit addition is performed. For this purpose, a dull-adder is designed.

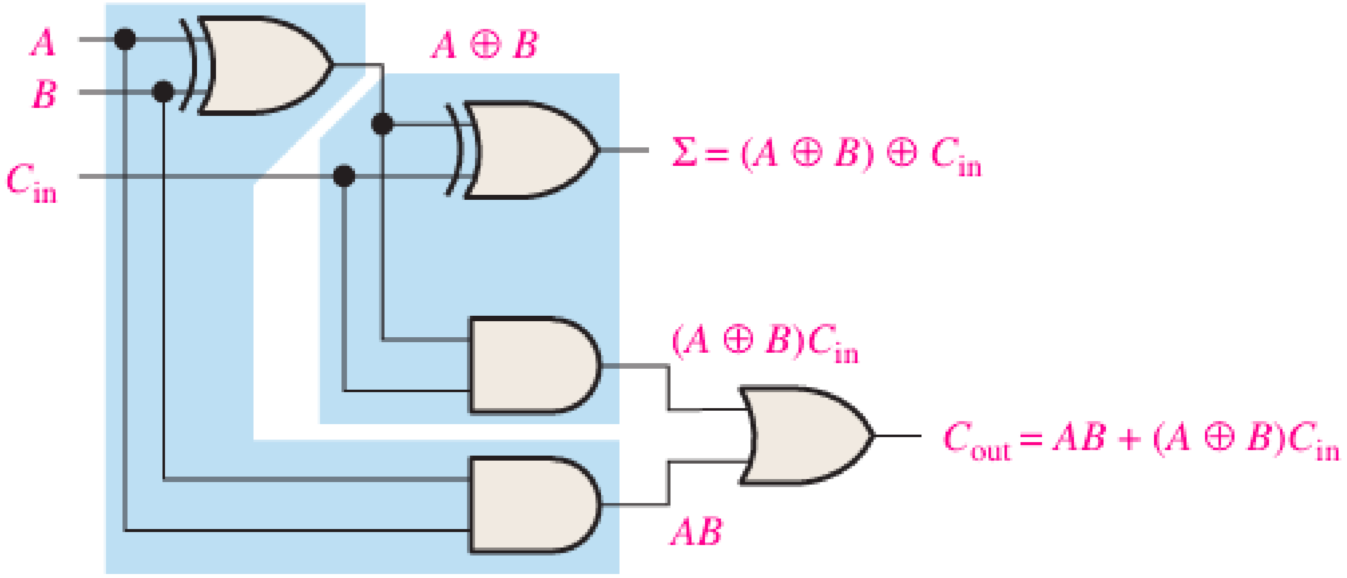
A full adder is a combinational circuit that performs the arithmetic sum of three input bits and produces a sum output and a carry.

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

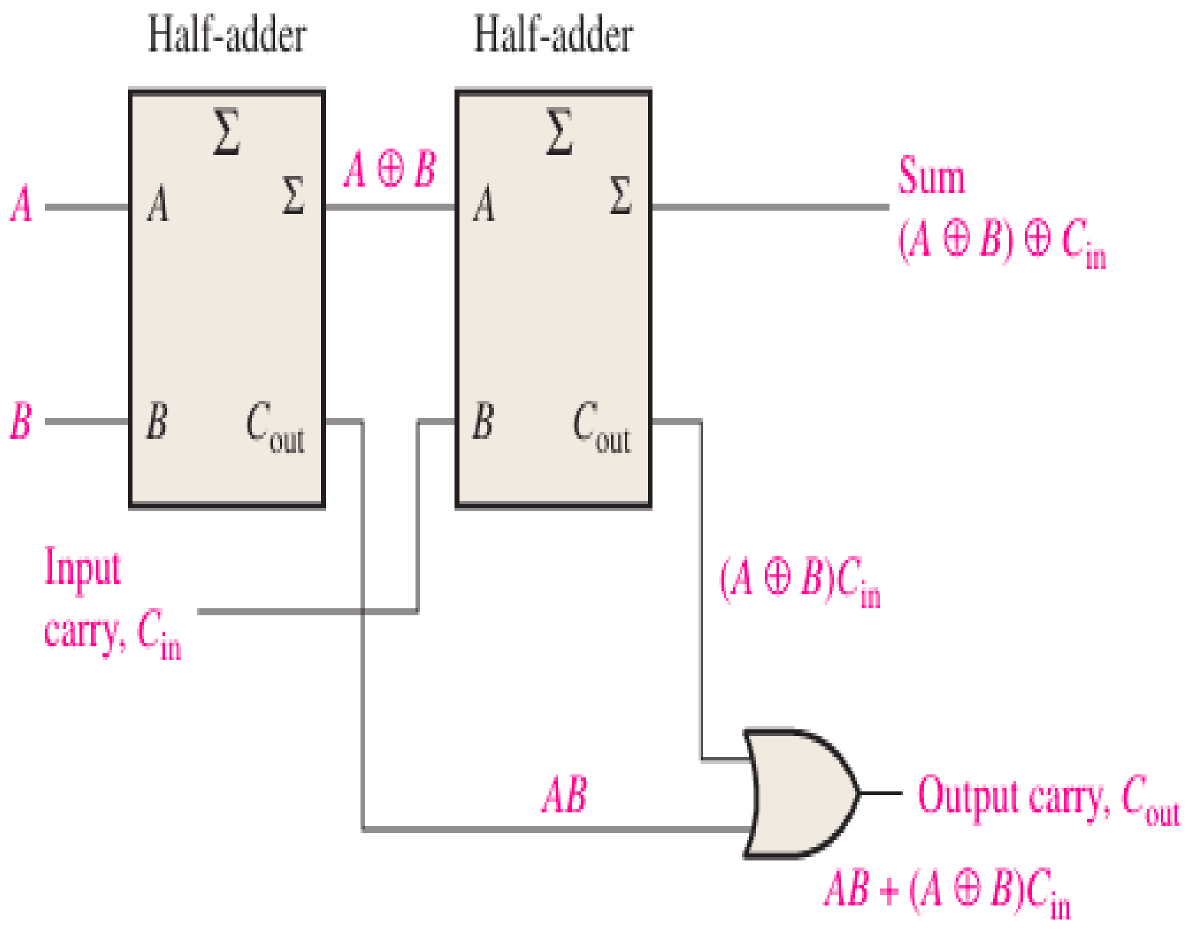
$$C_{out} = AB + (A \oplus B)C_{in}$$



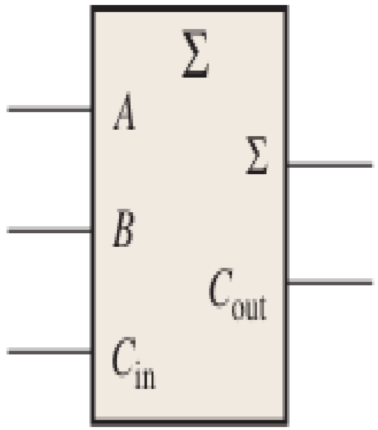
(a) Logic required to form the sum of three bits



(b) Complete logic circuit for a full-adder (each half-adder is enclosed by a shaded area)

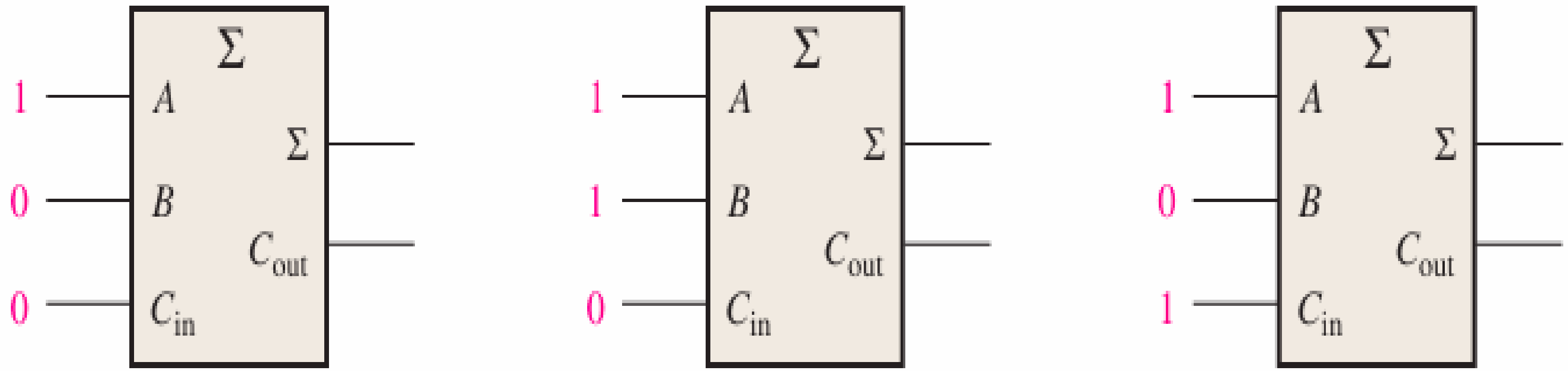


(a) Arrangement of two half-adders to form a full-adder



(b) Full-adder logic symbol

A	B	C_{in}	Carry	Sum
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) (b) (c)

Solution

(a) The input bits are $A = 1$, $B = 0$, and $C_{in} = 0$.

$$1 + 0 + 0 = 1 \text{ with no carry}$$

Therefore, $\Sigma = \mathbf{1}$ and $C_{out} = \mathbf{0}$.

(b) The input bits are $A = 1$, $B = 1$, and $C_{in} = 0$.

$$1 + 1 + 0 = 0 \text{ with a carry of } 1$$

Therefore, $\Sigma = \mathbf{0}$ and $C_{out} = \mathbf{1}$.

(c) The input bits are $A = 1$, $B = 0$, and $C_{in} = 1$.

$$1 + 0 + 1 = 0 \text{ with a carry of } 1$$

Therefore, $\Sigma = \mathbf{0}$ and $C_{out} = \mathbf{1}$.

Parallel Binary Adders

Two or more full-adders are connected to form parallel binary adders.

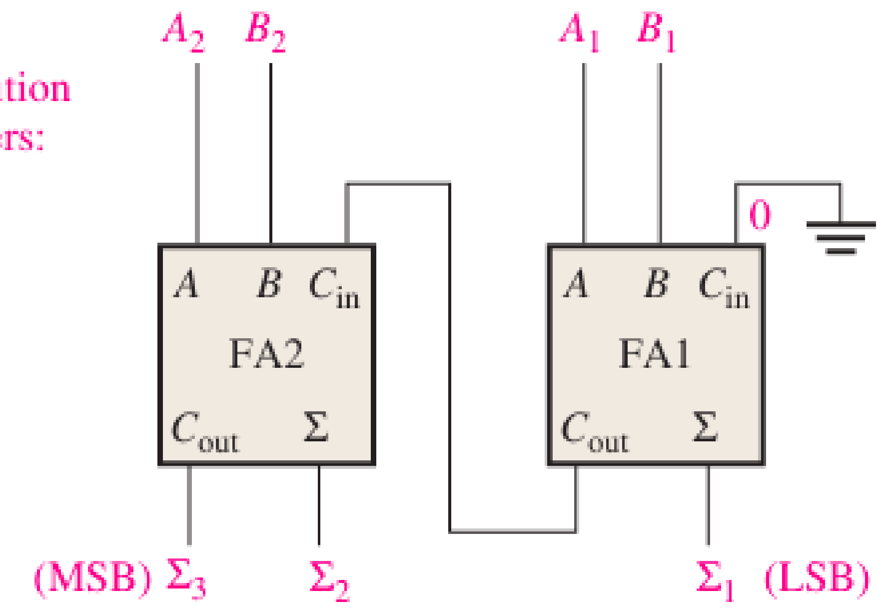
Carry bit from right column

$$\begin{array}{r} {\scriptstyle 1} \\ 11 \\ +\ 01 \\ \hline 100 \end{array}$$

In this case, the carry bit from second column becomes a sum bit.

General format, addition of two 2-bit numbers:

$$\begin{array}{r} A_2A_1 \\ +\ B_2B_1 \\ \hline \Sigma_3\Sigma_2\Sigma_1 \end{array}$$



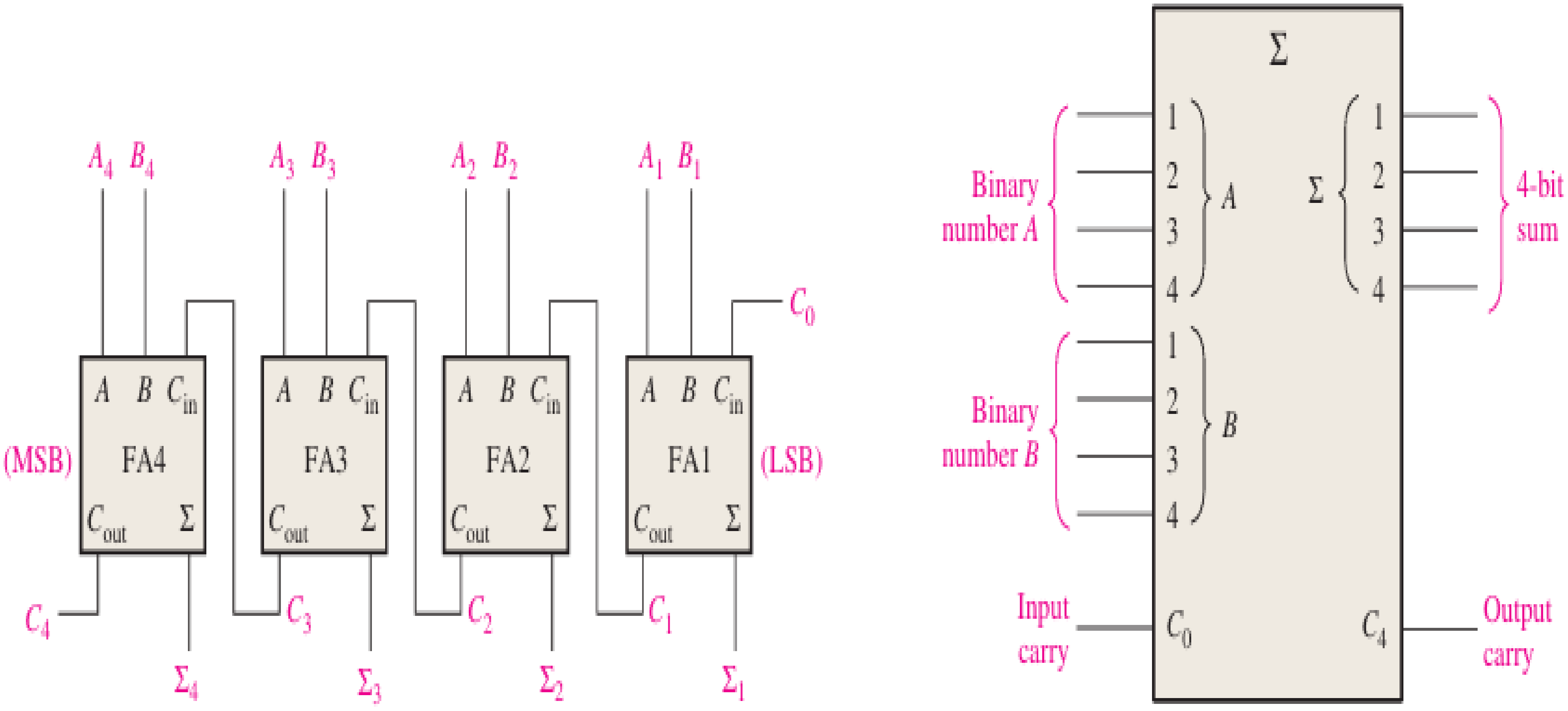
Four-Bit Parallel Adders

A group of four bits is called a nibble.

A basic 4-bit parallel adder is implemented with four full-adder.

Again, the LSBs (A1 and B1) in each number being added go into the right-most full-adder; the higher-order bits are applied to the successively higher-order adders, with the MSBs (A4 and B4) in each number being applied to the left-most full-adder.

The carry output of each adder is connected to the carry input of the next higher-order adder. These are called internal carries.



(a) Block diagram

(b) Logic symbol

Truth Table for a 4-Bit Parallel Adder

Truth table for each stage of a 4-bit parallel adder.

C_{n-1}	A_n	B_n	Σ_n	C_n
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

Use the 4-bit parallel adder truth table to find the sum and output carry for the addition of the following two 4-bit numbers if the input carry (Cn-1) is 0:

$$A_4A_3A_2A_1 = 1100 \quad \text{and} \quad B_4B_3B_2B_1 = 1100$$

Solution

For $n = 1$: $A_1 = 0$, $B_1 = 0$, and $C_{n-1} = 0$. From the 1st row of the table,

$$\Sigma_1 = \mathbf{0} \quad \text{and} \quad C_1 = 0$$

For $n = 2$: $A_2 = 0$, $B_2 = 0$, and $C_{n-1} = 0$. From the 1st row of the table,

$$\Sigma_2 = \mathbf{0} \quad \text{and} \quad C_2 = 0$$

For $n = 3$: $A_3 = 1$, $B_3 = 1$, and $C_{n-1} = 0$. From the 4th row of the table,

$$\Sigma_3 = \mathbf{0} \quad \text{and} \quad C_3 = 1$$

For $n = 4$: $A_4 = 1$, $B_4 = 1$, and $C_{n-1} = 1$. From the last row of the table,

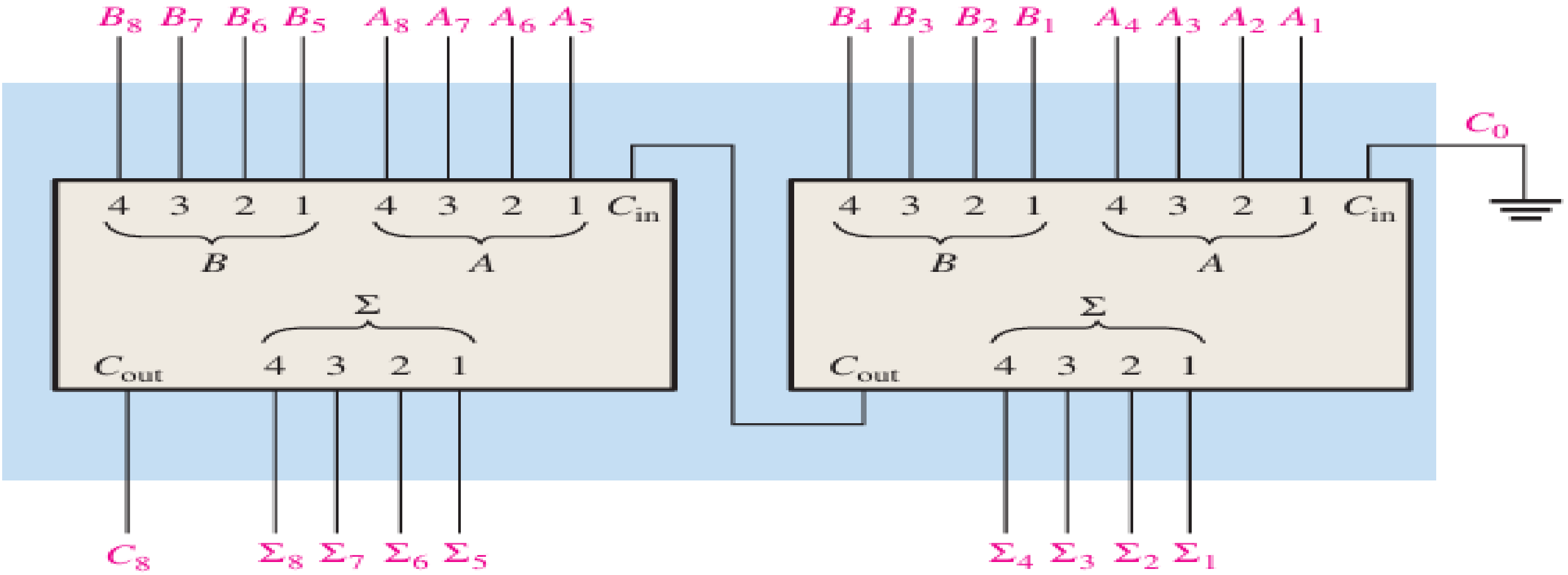
$$\Sigma_4 = \mathbf{1} \quad \text{and} \quad C_4 = \mathbf{1}$$

C_4 becomes the output carry; the sum of 1100 and 1100 is 11000.

Adder Expansion

Adders can be expanded to handle more bits by cascading.

The 4-bit parallel adder can be expanded to handle the addition of two 8-bit numbers by using two 4-bit adders.
The carry input of the low-order adder (C0) is connected to ground because there is no carry into the least significant bit position, and the carry output of the low-order adder is connected to the carry input of the high-order adder. This process is known as cascading. Notice that, in this case, the output carry is designated C8 because it is generated from the eighth bit position.



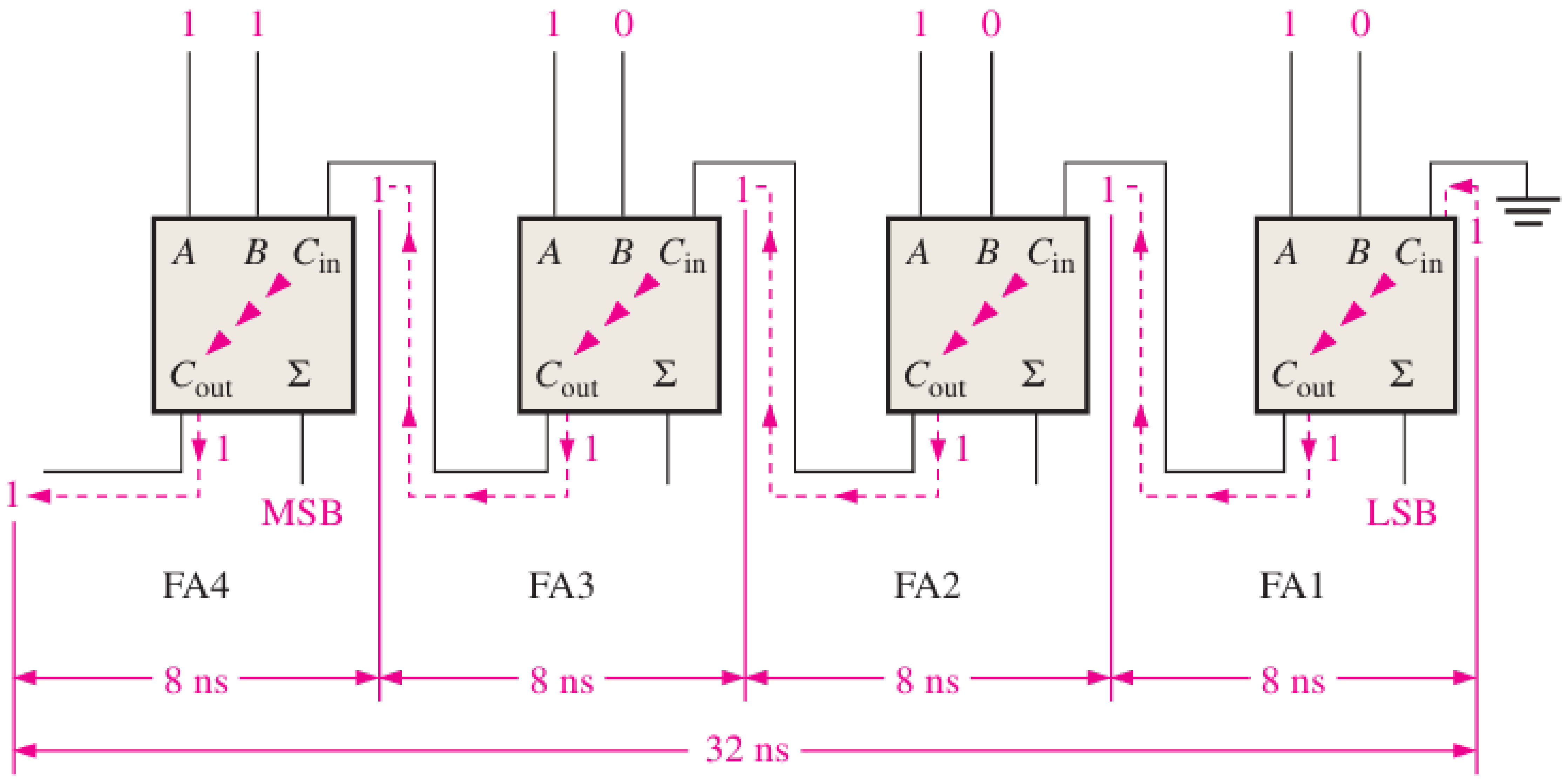
Ripple Carry and Look-Ahead Carry Adders

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; this causes a time delay in the addition process.

- The carry propagation delay for each full-adder is the time from the application of the input carry until the output carry occurs, assuming that the A and B inputs are already present.



1 1
1 0
1 0
1 0
A B C_{in}
C_{out} Σ
1
1
1
1
1
MSB
LSB
FA4
FA3
FA2
FA1
8 ns
8 ns
8 ns
8 ns
32 ns