




---

---

---

---

---

---

---

---

**Summary**

**Binary Addition**

The rules for binary addition are

0 + 0 = 0	Sum = 0, carry = 0
0 + 1 = 0	Sum = 1, carry = 0
1 + 0 = 0	Sum = 1, carry = 0
1 + 1 = 10	Sum = 0, carry = 1

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

**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	
<i>A</i>	<i>B</i>	<i>C<sub>out</sub></i>	$\Sigma$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The logic symbol and equivalent circuit are:

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

**Summary**

**Binary Addition**

When an input carry = 1 due to a previous result, the rules are

$1 + 0 + 0 = 01$	Sum = 1, carry = 0
$1 + 0 + 1 = 10$	Sum = 0, carry = 1
$1 + 1 + 0 = 10$	Sum = 0, carry = 1
$1 + 1 + 1 = 10$	Sum = 1, carry = 1

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

**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.

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

Inputs			Outputs	
A	B	C <sub>in</sub>	C <sub>out</sub>	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

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

**Summary**

**Full-Adder**

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

**Solution** The first half-adder has inputs of 1 and 0; therefore the Sum = 1 and the Carry out = 0.

The second half-adder has inputs of 1 and 1; therefore the Sum = 0 and the Carry out = 1.

The OR gate has inputs of 1 and 0, therefore the final carry out = 1.

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

**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 <sub>in</sub>	C <sub>out</sub>	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

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

---

---

**Summary**

**Binary Addition**

**Example** Add the binary numbers 00111 and 10101 and show the equivalent decimal addition.

**Solution**

$$\begin{array}{r}
 0111 \\
 00111 \quad 7 \\
 10101 \quad 21 \\
 \hline
 11100 = 28
 \end{array}$$

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

---

---

**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<sub>4</sub>) is not ready until it propagates through all of the full adders. This is called *ripple carry*, delaying the addition process.

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

---

---



**Summary**

**1's Complement**

The 1's complement of a binary number is just the inverse of the digits. To form the 1's complement, change all 0's to 1's and all 1's to 0's.

For example, the 1's complement of 11001010 is  
00110101

In digital circuits, the 1's complement is formed by using inverters:

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

**Summary**

**2's Complement**

The 2's complement of a binary number is found by adding 1 to the LSB of the 1's complement.

Recall that the 1's complement of 11001010 is  
00110101 (1's complement)

To form the 2's complement, add 1:

$$\begin{array}{r} 00110101 \text{ (1's complement)} \\ + 1 \\ \hline 00110110 \text{ (2's complement)} \end{array}$$

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

**Summary**

**Signed Binary Numbers**

There are several ways to represent signed binary numbers. In all cases, the MSB in a signed number is the sign bit, that tells you if the number is positive or negative.

Computers use a modified 2's complement for signed numbers. Positive numbers are stored in *true* form (with a 0 for the sign bit) and negative numbers are stored in *complement* form (with a 1 for the sign bit).

For example, the positive number 58 is written using 8-bits as  
00111010  
↑ Sign bit      ↑ Magnitude bits

Floyd, Digital Fundamentals, 10<sup>th</sup> ed © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

**Summary**

**Signed Binary Numbers**

Negative numbers are written as the 2's complement of the corresponding positive number.  
 The negative number -58 is written as:  
 -58 = 11000110 (complement form)

↑ Sign bit      ↓ Magnitude bits

An easy way to read a signed number that uses this notation is to assign the sign bit a column weight of -128 (for an 8-bit number). Then add the column weights for the 1's.

**Example** Assuming that the sign bit = -128, show that 11000110 = -58 as a 2's complement signed number.

**Solution**

Column weights:	-128	64	32	16	8	4	2	1.
	1	1	0	0	0	1	1	0
	-128	+64				+4	+2	= -58

Floyd, Digital Fundamentals, 10<sup>th</sup> ed      © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---

**Circuit for 2's complement Numbers**

Fig. 3-31 Adder-Subtractor Circuit

- o **No Correction is needed** if the signed numbers are in 2's complement representation

63

Floyd, Digital Fundamentals, 10<sup>th</sup> ed      © 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

---

---

---

---

---

---

---

---

---

---