Digital Logic :

Next Topic :

Addition of two numbers in the signed-magnitude system

Arithmetic Addition

The addition of two numbers in the signed-magnitude system follows the rules of ordinary arithmetic. If the signs are the same, we add the two magnitudes and give the sum the common sign. If the signs are different, we subtract the smaller magnitude from the larger and give the difference the sign of the larger magnitude. For example, $(+25) + (-37) = -(37 - 25) = -12$ is done by subtracting the smaller magnitude, 25, from the larger magnitude, 37, and appending the sign of 37 to the result. This is a process that requires a comparison of the signs and magnitudes and then performing either addition or subtraction. The same procedure applies to binary numbers

Digital Logic :

Next Topic :

Addition of two numbers in the 2's Complement system

the rule for adding numbers in the signed-complement system does not require a comparison or subtraction, but only addition. The procedure is very simple and can be stated as follows for binary numbers:

The addition of two signed binary numbers with negative numbers represented in signed-2's-complement form is obtained from the addition of the two numbers, including their sign bits. A carry out of the sign-bit position is discarded.

Numerical examples for addition follow:

+ 6	00000110	− 6	11111010
+13	00001101	+13	00001101
+19	00010011	+ 7	00000111
+ 6	00000110	− 6	11111010
−13	11110011	−13	11110011
− 7	11111001	−19	11101101

Note that negative numbers must be initially in 2's-complement form and that if the sum obtained after the addition is negative, it is in 2's-complement form. For example, −7 is represented as 11111001, which is the 2s complement of +7.

In each of the four cases, the operation performed is addition with the sign bit included. Any carry out of the sign-bit position is discarded, and negative results are automatically in 2's-complement form.

The addition of two signed binary numbers with negative numbers represented in signed-2's-complement form is obtained from the addition of the two numbers, including their sign bits. A carry out of the sign-bit position is discarded.

Numerical examples for addition follow:

$$\begin{array}{rr} +\ 6 & 00000110 \\ \underline{+13} & \underline{00001101} \\ +19 & 00010011 \end{array} \qquad \begin{array}{rr} -\ 6 & 11111010 \\ \underline{+13} & \underline{00001101} \\ +\ 7 & 00000111 \end{array}$$

$$\begin{array}{rr} +\ 6 & 00000110 \\ \underline{-13} & \underline{11110011} \\ -\ 7 & 11111001 \end{array} \qquad \begin{array}{rr} -\ 6 & 11111010 \\ \underline{-13} & \underline{11110011} \\ -19 & 11101101 \end{array}$$

Digital Logic :

Next Topic :

signed

Addition of two numbers in the

1's Complement system

1. Input — in 1's Comp Rep.
   output —

2. End Carry must be Added back to Result to get final Result.

5-bits:

$(+4) + (+9) = 13$

$00100 + 01001$

in 1's Comp Rep

$$\begin{array}{r} 00100 \\ 01001 \\ \hline 01101 \\ \hline \end{array} \rightarrow 13$$

5 bits

$(\underline{\underline{+4}}) + (\underline{\underline{-9}}) = -5$

$+9 = \underbrace{01001}$

00100 + 10110

is in 1's Comp.

$$\begin{array}{r} 00100 \\ 10110 \\ \hline 11010 \\ \hline \end{array} \rightarrow -5$$

5 bits

$(-4) + (+9) = +5$ ✓

$+4 = 00100$

$11011 + 01001$

$$\begin{array}{r} 11011 \\ +\ 01001 \\ \hline 100100 \\ +1 \\ \hline 00101 \end{array}$$

100100 → Not the final Result

Final Result of Addition → 00101 → 5 ✓

5 bits

$$\underline{\underline{(-4)}} + \underline{\underline{(-9)}} = -13$$

$$11011 + 10110$$

$$
\begin{array}{r}
\scriptstyle 1\ 1\ 1\quad\ \ \\
11011 \\
+\ 10110 \\
\hline
1\,10001 \\
+1 \\
\hline
10010
\end{array}
$$

10001 → Not the final Result

final Result of Addition → 10010 ✓ → -13 ✓

The following examples illustrate the addition of 1's and 2's complement numbers for a word length of $n = 8$:

1. Add -11 and -20 in 1's complement.

$+11 = 00001011 \qquad +20 = 00010100$

taking the bit-by-bit complement,

-11 is represented by 11110100 and -20 by 11101011

$$
\begin{array}{rl}
11110100 & (-11) \\
\underline{11101011} & \underline{+(-20)} \\
(1)\ 11011111 & \\
\longrightarrow 1 & \text{(end-around carry)} \\
\hline
11100000 = -31 &
\end{array}
$$

in 1's Comp Rep.

Add -11 and -20 in 2's Comp : (8 bits)

11110101 11101100

Discard

1111 0101
1110 1100
1 11100001 $\Rightarrow -31$

final Result

$+20 =$
00010100
11101100

10101 -16 $+5$

11110101

$+11 =$
00001011
11110101

2. Add –8 and +19 in 2's complement

+8 = 00001000

complementing all bits to the left of the first 1, –8, is represented by 11111000

$$
\begin{array}{lll}
 & 11111000 & (-8) \\
 & \underline{00010011} & \underline{+19} \\
\cancel{(1)} & 00001011 = & +11
\end{array}
$$

(discard last carry)

2. Add –8 and +19 in [redacted] sign magnitude system (8 bits)

[redacted]

1 0001 000 + 0001 0011

sign

magnitude

Compare

Final Ans:

001 0011
-000 1000
000 1011

0 0001011

Sign Different $\Longrightarrow$ Compare magnitude

$(-8) + (+19) =$

$$\begin{array}{r} 19 \\ -\ 08 \\ \hline +(11) \end{array}$$

Subtraction of magnitude

Addition of Signed Numbers:

1. in sm system: Same as Decimal system
   - we handle sign bit separately
   - " " magnitude "

If your Digital ckt is using SM system for signed numbers:

- Adder Circuit
- Subtraction Circuit
- O(n) time

Comparator Circuit

(2) In 2's Comp :

$$\begin{array}{r} 0011 \\ +\ 0100 \\ \hline 0111 \\ \hline \end{array}$$

in 2's complement

(2) In 1's Comp ;

$$\begin{array}{r} 0\ 0\ 1\ 1 \\ +\ 0\ 1\ 0\ 0 \\ \hline 0\ 1\ 1\ 1 \end{array} \Bigg\}\ \text{in 1's Complement}$$

+ End Carry $\Longrightarrow$ Extra Addition

Algo 1

All input

1000000000000000 Years

O(1) time

Algo 2

O(n) seconds

Algo People	Reality
Algo 1 ✓	Algo 2

+9 : 001001

-9

1's Comp

110110

2's Comp

110111

$A - B$

minuend

subtrahend

Digital Logic :

Next Topic :

signed

Subtraction of two numbers in the

2's Complement system

Arithmetic Subtraction

Subtraction of two signed binary numbers when negative numbers are in 2's-complement form is simple and can be stated as follows:

Take the 2's complement of the subtrahend (including the sign bit) and add it to the minuend (including the sign bit). A carry out of the sign-bit position is discarded.

Assume:

A, B in 2's Comp Rep.

$A - B = \boxed{A + (2\text{'s Comp of } B)}$

End Carry: Discarded

5 bits:

+4 : 00100 : A

+9 : 01001 : B

$\underline{\underline{(+4)}} - \underline{\underline{(+9)}}$

$00100 \boxed{- \quad 01001}$

00100 + 10111

$$\begin{array}{r} 00100 \\ +10111 \\ \hline 11011 \end{array}$$

-5

Correct Result

5 bits:

-4 : 11100 ; A

$+9$: 01001 ; B

$A - B$

$(-4) - (+9)$

$= -13$

$\overset{A}{(11100)} - \underbrace{\overset{B}{(01001)}}$

$(11100) + (10111)$

$$
\begin{array}{r}
11100 \\
+\;10111 \\
\hline
1\;\boxed{10011}
\end{array}
\longrightarrow -13
$$

Discarded

Correct Result

5 bits:

+4 : 00100 A

−9 : 10110 B

A − B

(+4) − (−9)

$\overset{A}{00100} - \overset{B}{10110}$

00100 + 01010

$$\begin{array}{r} 00100 \\ 01010 \\ \hline 01110 \end{array} \rightarrow +14 \checkmark$$

5 bits:

$-4 : 11100 \quad A$

$-9 : 10111 \quad B$

$A - B$

$(-4) - (-9)$

$\overset{A}{11100} - \overset{B}{10111}$

$11100 + 01001$

Discard

$$\begin{array}{r} 11100 \\ 01001 \\ \hline 1\,00101 \end{array} \rightarrow 5 \checkmark$$

A ; B : Signed Numbers in 2's Comp Rep

A + B

Discard Carry.

Read the output in 2's Comp Rep

A − B

A + (2's Comp of B)

Discard End Carry.

A ; B : Signed Numbers in 1's Comp Rep

$A + B$

Add end Carry.

Read the output in 1's Comp Rep

$A - B$

$A + (\text{1's Comp of } B)$

Add End Carry.

Subtraction of two signed binary numbers when negative numbers are in 2's-complement form is simple and can be stated as follows:

> Take the 2's complement of the subtrahend (including the sign bit) and add it to the minuend (including the sign bit). A carry out of the sign-bit position is discarded.

This procedure is adopted because a subtraction operation can be changed to an addition operation if the sign of the subtrahend is changed, as is demonstrated by the following relationship:

$$(\pm A) - (+B) = (\pm A) + (-B);$$
$$(\pm A) - (-B) = (\pm A) + (+B).$$

But changing a positive number to a negative number is easily done by taking the 2's complement of the positive number. The reverse is also true, because the complement of a negative number in complement form produces the equivalent positive number. To see this, consider the subtraction $(-6) - (-13) = +7$. In binary with eight bits, this operation is written as $(11111010 - 11110011)$. The subtraction is changed to addition by taking the 2's complement of the subtrahend (–13), giving (+13). In binary, this is $11111010 + 00001101 = 100000111$. Removing the end carry, we obtain the correct answer: 00000111 (+7).

Digital Logic :

Next Topic :

Subtraction of two numbers in the 1's Complement system

5 bits:

+4 : 00100 A

−5 : 11010 B

$\underbrace{A - B}$

$(+4) - (-5)$

$= +9$

A − B

$00100 - \boxed{11010}$

$00100 + 00101$

$$\begin{array}{r} 00100 \\ 00101 \\ \hline 01001 \\ \hline \end{array} \rightarrow \underline{\underline{+9}}$$

5 bits:

$-4 : 11011$ A

$-5 : 11010$ B

A − B

$(-4) - (-5)$

A B

$11011 - 11010$

$11011 + 00101$

$$\begin{array}{r} 11011 \\ 00101 \\ \hline 1\,00000 \end{array}$$

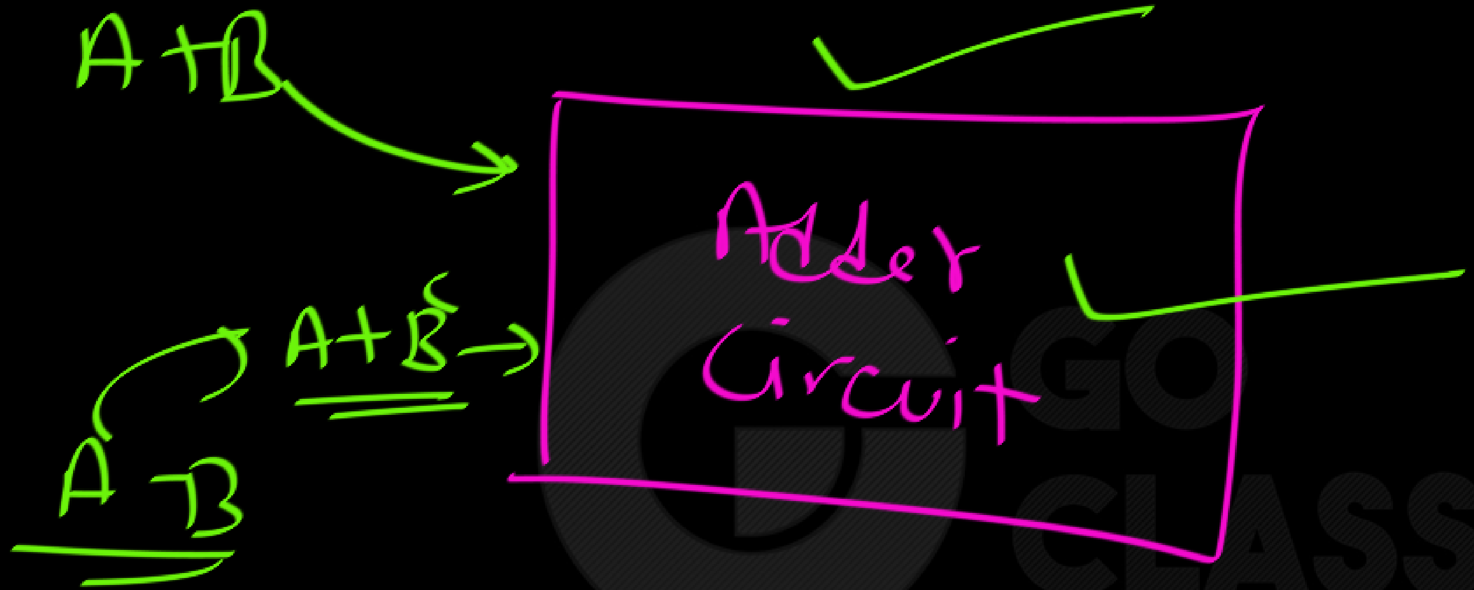
$+1$

00001

✓1

1's comp or 2's comp

It is worth noting that binary numbers in the signed-complement system are added and subtracted by the same basic addition and subtraction rules as unsigned numbers. Therefore, **computers need only one common hardware circuit to handle both types of arithmetic.** This consideration has resulted in the signed-complement system being used in virtually all arithmetic units of computer systems. The user or programmer must interpret the results of such addition or subtraction differently, depending on whether it is assumed that the numbers are signed or unsigned.


A+B
A+B
A B
Adder
Circuit

2's Comp system is better than 1's Comp System for Signed numbers:

1. No Extra Addition of End Carry.
2. No two Rep of 0.
3. Range is more.

Unsigned and Signed Variables

- Unsigned variables use unsigned binary (normal power-of-2 place values) to represent numbers

1	0	0	1	0	0	1	1	= +147
128	64	32	16	8	4	2	1	

- Signed variables use the 2's complement system (Neg. MSB weight) to represent numbers

1	0	0	1	0	0	1	1	= -109
-128	64	32	16	8	4	2	1	