Jiangxi University of Science and Technology

DIGITAL DESIGN

Binary Arithmetic Addition and Subtraction of Binary Numbers

The CARRY flag and OVERFLOW flag in binary arithmetic

Binary Addition

It is a key for binary subtraction, multiplication, division.
There are four rules of binary addition.
In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

Case	A + B	Sum	Carry
1	0 + 0	0	0
2	0 + 1	1	0
3	1 + 0	1	0
4	1 + 1	0	1

Binary Subtraction

- **Subtraction and Borrow**, these two words will be used very frequently for the binary subtraction. There are four rules of binary subtraction

Case	A - B	Subtract	Borrow
1	0 - 0	0	0
2	1 - 0	1	0
3	1 - 1	0	0
4	0 - 1	0	1

Carry Flag

The rules for turning on the carry flag in binary/integer math are two:

1. The carry flag is set if the addition of two numbers causes a carry out of the most significant (leftmost) bits added.

   1111 + 0001 = 0000 (carry flag is turned on)

2. The carry (borrow) flag is also set if the subtraction of two numbers requires a borrow into the most significant (leftmost) bits subtracted.

   0000 - 0001 = 1111 (carry flag is turned on)

- Otherwise, the carry flag is turned off (zero).

  * 0111 + 0001 = 1000 (carry flag is turned off [zero])

  * 1000 - 0001 = 0111 (carry flag is turned off [zero])

In unsigned arithmetic, watch the carry flag to detect errors.
In signed arithmetic, the carry flag tells you nothing interesting.

OVERFLOW flag

- The rules for turning on the overflow flag in binary/integer math are two:
- 1. If the sum of two numbers with the sign bits off yields a result number with the sign bit on, the "overflow" flag is turned on.

  0100 + 0100 = 1000 (overflow flag is turned on)
- 2. If the sum of two numbers with the sign bits on yields a result number with the sign bit off, the "overflow" flag is turned on.

  1000 + 1000 = 0000 (overflow flag is turned on)
- Otherwise, the overflow flag is turned off.

  *** 0100 + 0001 = 0101 (overflow flag is turned off)**

  *** 0110 + 1001 = 1111 (overflow flag is turned off)**

  *** 1000 + 0001 = 1001 (overflow flag is turned off)**

  *** 1100 + 1100 = 1000 (overflow flag is turned off)**

OVERFLOW flag

- Note that you only need to look at the sign bits (leftmost) of the three numbers to decide if the overflow flag is turned on or off.
- If you are doing two's complement (signed) arithmetic, overflow flag on means the answer is wrong - you added two positive numbers and got a negative, or you added two negative numbers and got a positive.
- If you are doing unsigned arithmetic, the overflow flag means nothing and should be ignored.
- The rules for two's complement detect errors by examining the sign of the result. A negative and positive added together cannot be wrong, because the sum is between the addends. Since both of the addends fit within the allowable range of numbers, and their sum is between them, it must fit as well. Mixed-sign addition never turns on the overflow flag.
- In signed arithmetic, watch the overflow flag to detect errors.
- In unsigned arithmetic, the overflow flag tells you nothing interesting.

Binary Subtraction/addition

0011010 - 001100 = 00001110

$$
\begin{array}{rl}
\;\;\;\;1\,1\;\;\;\;\; & \text{borrow} \\
0\,0\,\cancel{1\,1}\,0\,1\,0 & = 26_{10} \\
-\,0\,0\,0\,1\,1\,0\,0 & = 12_{10} \\
\hline
0\,0\,0\,1\,1\,1\,0 & = 14_{10}
\end{array}
$$

0011010 + 001100 = 00100110

$$
\begin{array}{rl}
1\,1\;\;\;\;\; & \text{carry} \\
0\,0\,1\,1\,0\,1\,0 & = 26_{10} \\
+\,0\,0\,0\,1\,1\,0\,0 & = 12_{10} \\
\hline
0\,1\,0\,0\,1\,1\,0 & = 38_{10}
\end{array}
$$

Binary Multiplication

- Binary multiplication is similar to decimal multiplication.
- It is simpler than decimal multiplication because only 0s and 1s are involved.
- There are four rules of binary multiplication.

Case	A x B	Multiplication
1	0 x 0	0
2	0 x 1	0
3	1 x 0	0
4	1 x 1	1

Multiplication (binary)

It's interesting to note that binary multiplication is a sequence of shifts and adds of the first term (depending on the bits in the second term.

$$
\begin{array}{r}
\boxed{1101} \\
\times 1011 \\
\hline
\boxed{1101} \\
\boxed{1101}0 \\
+\boxed{1101}000 \\
\hline
10001111
\end{array}
$$

*1101*00 is missing here because the corresponding bit in the second terms is 0.

Multiplication

Example:

0011010 x 001100 = 100111000

$$
\begin{array}{r@{\quad}l}
0011010 & = 26_{10} \\
\times\, 0001100 & = 12_{10} \\
\hline
0000000 & \\
0000000 & \\
0011010 & \\
0011010 & \\
\hline
0100111000 & = 312_{10}
\end{array}
$$

Binary Division

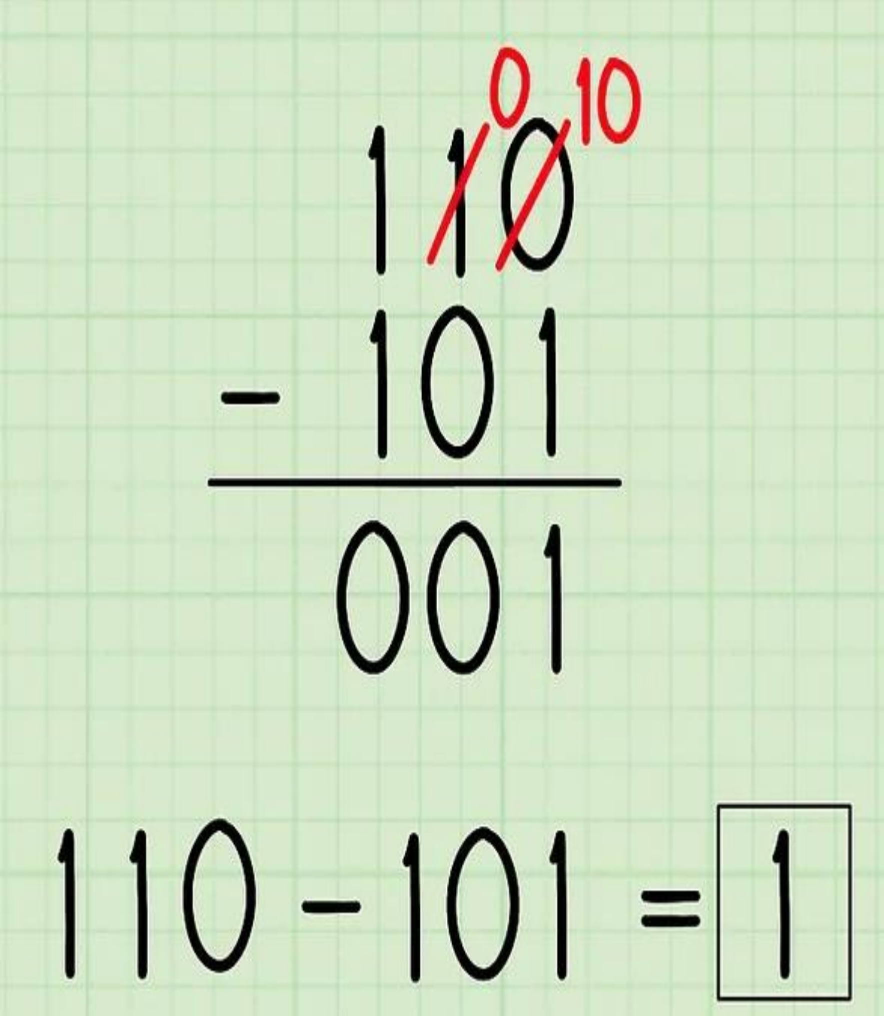
Binary division is similar to decimal division.

It is called as the long division procedure.

101010 / 000110 = 000111

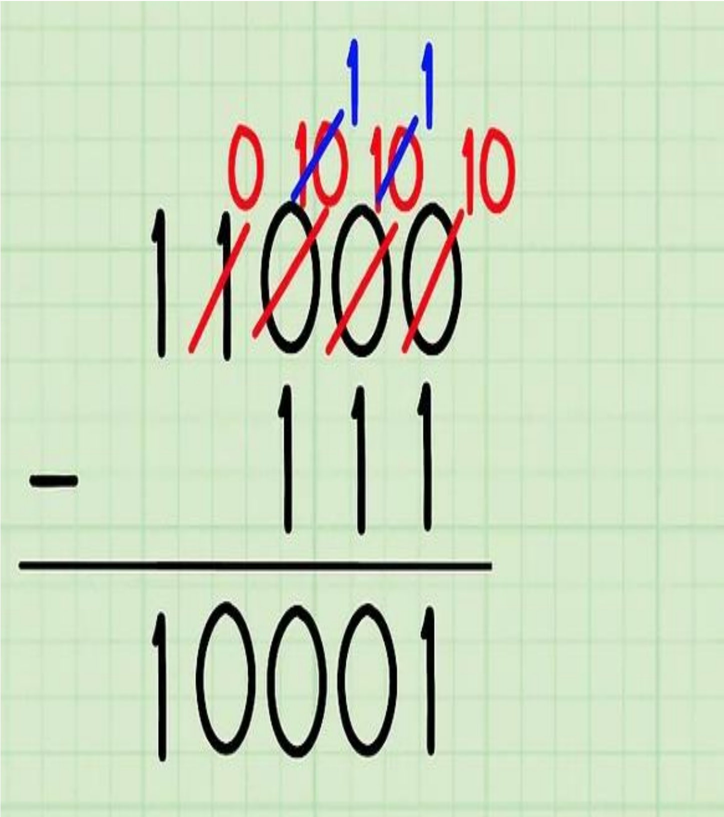
```
                 1 1 1        = 7₁₀
        ____________
000110 ) 1 0 1 0 1 0          = 42₁₀
        -1 1 0                = 6₁₀
        ------
          1 0 0 1
         -1 1 0
         -------
            1 1 0
           -1 1 0
           ------
                0
```

Addition and Subtraction



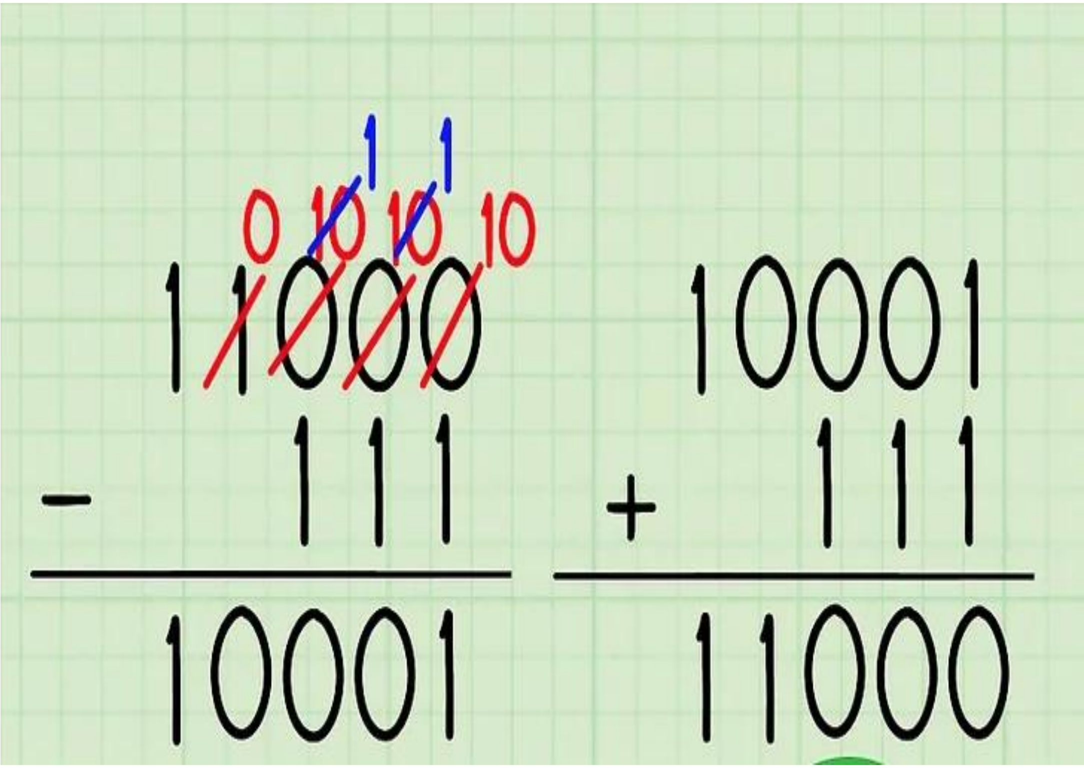
Solve the rightmost column. Now each column can be solved as usual.
Here's how to solve the rightmost column (the ones place) in this problem:
101100 - 101 = ?
The rightmost column is now: 10 - 1 = 1.
If you can't figure out how to reach this answer, here's how to convert the problem back to decimal:
102 = (1 x 2) + (0 x 1) = 210.
(The sub numbers indicate which base the number is written in.)
12 = (1x1) = 110.
Therefore, in decimal form this problem is 2 - 1 = ?, so the answer is 1.

Addition and Subtraction



- Borrowing comes up a lot in binary multiplication, and sometimes you'll need to borrow multiple times just to solve one column. For example, here's how to solve **11000 - 111**. We can't "borrow" from a 0, so we need to keep borrowing from the left until we turn it into something we can borrow from: $\mathbf{1^{0}\cancel{1}^{10}\cancel{0}00 - 111 =}$
- $\mathbf{1^{0}\cancel{1}^{1\cancel{10}}\cancel{0}^{10}\cancel{0}0 - 111 =}$ (remember, 10 - 1 = 1)
- $\mathbf{1^{0}\cancel{1}^{1\cancel{10}}\cancel{0}^{1\cancel{10}}\cancel{0}^{10}\cancel{0} - 111 =}$
- Here it is written more tidily: $\mathbf{1011^{10}\cancel{0} - 111 =}$
- Solve column by column: **_ _ _ _ 1 = _ _ _ 0 1 = _ _ 0 0 1 = _ 0 0 0 1 = 1 0 0 0 1**

Addition and Subtraction



Addition and Subtraction

- **101 from 1001**

  Solution:

  101 from 1001

  1 Borrow

  1 0 0 1

  1 0 1

  1 0 0

111 from 1000

Solution:

111 from 1000

1 Borrow

1 0 0 0

1 1 1

0 0 0 1

Addition and Subtraction

- **11010.101 from 101100.011**

  Solution:

  11010.101 from 101100.011

  1   1   1   Borrow

  1 0 1 1 0 0 . 0 1 1

  1 1 0 1 0 . 1 0 1

  1 0 0 0 1 . 1 1 0

1010101.10 from 1111011.11

Solution:

1010101.10 from 1111011.11

1 Borrow

1 1 1 1 0 1 1 . 1 1

1 0 1 0 1 0 1 . 1 0

1 0 0 1 1 0 . 0 1

Jiangxi University of Science and Technology

Subtraction by 1’s Complement

Subtraction by 1’s Complement

- In subtraction by 1’s complement we subtract two binary numbers using carried by 1’s complement.

- **The steps to be followed in subtraction by 1’s complement are:**
  - i) To write down 1’s complement of the subtrahend.
  - ii) To add this with the minuend.
  - iii) If the result of addition has a carry over then it is dropped and an 1 is added in the last bit.
  - iv) If there is no carry over, then 1’s complement of the result of addition is obtained to get the final result and it is negative.

Subtraction by 1's Complement

- **(i) 110101 – 100101**

1's complement of 10011 is 011010. Hence

Minued -		1 1 0 1 0 1
1's complement of subtrahend -		0 1 1 0 1 0
Carry over -	1	0 0 1 1 1 1
		1
		0 1 0 0 0 0

The required difference is 10000

Subtraction by 1's Complement

- **101011 – 111001**
- **Solution:**

1's complement of 111001 is 000110. Hence

Minued -	1 0 1 0 1 1
1's complement -	0 0 0 1 1 0
	1 1 0 0 0 1

- **Hence the difference is – 1 1 1 0**

Subtraction by 1's Complement

- **1011.001 – 110.10**
- **Solution:**

1's complement of 0110.100 is 1001.011 Hence

Minued -		1 0 1 1 . 0 0 1
1's complement of subtrahend -		1 0 0 1 . 0 1 1
Carry over -	1	0 1 0 0 . 1 0 0
		1
		0 1 0 0 . 1 0 1

Subtraction by 1's Complement

- **10110.01 – 11010.10**
- **Solution:**

1's complement of 11010.10 is 00101.01

$$\begin{array}{r} 10110.01 \\ \underline{\;\;00101.01} \\ 11011.10 \end{array}$$

- **Hence the required difference is – 00100.01 i.e. – 100.01**

8-bit 1's complement addition

- Ex. Let X = $A8_{16}$ and Y = 86_{16}.
- Calculate Y - X using 1's complement.

$$Y = 1000\ 0110_2 = -121_{10}$$

$$X = 1010\ 1000_2 = -87_{10}$$

$$\sim X = 0101\ 0111_2$$

$$\begin{array}{r} 10000110 \\ +\,01010111 \\ \hline 1101\,1101 \end{array}$$

(Note: C=0 out of msb.)

Y - X = -121 + 87 = -34 (base 10)

8-bit 1's complement addition

- Ex. Let X = $A8_{16}$ and Y = 86_{16}.
- Calculate X - Y using 1's complement.

$X = 1010\ 1000_2 = -87_{10}$

$Y = 1000\ 0110_2 = -121_{10}$

$\sim Y = 0111\ 1001_2$

(Note: C=1 out of msb.)

$$
\begin{array}{r}
10101000 \\
+\ 01111001 \\
\hline
\boxed{1}00100001 \\
+1 \\
\hline
00100010
\end{array}
$$

end around carry

X - Y = -87 + 121 = 34 (base 10)

Jiangxi University of Science and Technology

Subtraction by 2’s Complement

Subtraction by 2's Complement

- With the help of subtraction by 2's complement method we can easily subtract two binary numbers

- **The operation is carried out by means of the following steps:**
  - (i) At first, 2's complement of the subtrahend is found.
  - (ii) Then it is added to the minuend.
  - (iii) If the final carry over of the sum is 1, it is dropped and the result is positive.
  - (iv) If there is no carry over, the two's complement of the sum will be the result and it is negative.

Subtraction by 2's Complement

- **110110 - 10110**
- **Solution:**
- The numbers of bits in the subtrahend is 5 while that of minuend is 6. We make the number of bits in the subtrahend equal to that of minuend by taking a `0' in the sixth place of the subtrahend.
- Now, 2's complement of 010110 is (101101 + 1) i.e.101010. Adding this with the minuend.
-

	1	1 0 1 1 0	Minuend
	1	0 1 0 1 0	2's complement of subtrahend
Carry over 1	1	0 0 0 0 0	Result of addition

- After dropping the carry over we get the result of subtraction to be 100000.
-

Subtraction by 2's Complement

- **10110 – 11010**
- **Solution:**

2's complement of 11010 is (00101 + 1) i.e. 00110. Hence

Minued -	1 0 1 1 0
2's complement of subtrahend -	0 0 1 1 0
Result of addition -	1 1 1 0 0

As there is no carry over, the result of subtraction is negative and is obtained by writing the 2's complement of 11100 i.e.(00011 + 1) or 00100.

Hence the difference is – 100

Subtraction by 2's Complement

- **1010.11 – 1001.01**
- **Solution:**
- 2's complement of 1001.01 is 0110.11. Hence
- Minued - 1 0 1 0 . 1 1

  2's complement of subtrahend - 0 1 1 0 . 1 1

  Carry over 1 0 0 0 1 . 1 0

- After dropping the carry over we get the result of subtraction as 1.10.

Subtraction by 2's Complement

- **10100.01 – 11011.10**
- **Solution:**
- 2's complement of 11011.10 is 00100.10. Hence
- Minued - 1 0 1 0 0 . 0 1

  2's complement of subtrahend - 0 1 1 0 0 . 1 0

  Result of addition - 1 1 0 0 0 . 1 1

- As there is no carry over the result of subtraction is negative and is obtained by writing the 2's complement of 11000.11.
- Hence the required result is – 00111.01.

Addition of 2's complement binary numbers

- Consider 8-bit 2's complement binary numbers.
  - Then the msb (bit 7) is the sign bit. If this bit is 0, then this is a positive number; if this bit is 1, then this is a negative number.
  - Addition of 2 positive numbers.
  - Ex. 40 + 58 = 98

$$
\begin{array}{r}
{\scriptstyle 1\ 1\ 1\ \ \ \ \ \ } \\
00101000 \\
+\,00111010 \\
\hline
01100010
\end{array}
$$

Addition of 2's complement binary numbers

- Consider 8-bit 2's complement binary numbers.
  - Addition of a negative to a positive.

  - What are the values of these 2 terms?
    - -88 and 122
    - -88 + 122 = 34

$$
\begin{array}{r}
1\ 1\ 1\ 1 \\
10101000 \\
+01111010 \\
\hline
1\ 00100010
\end{array}
$$

Addition of 2's complement binary numbers

- Consider 8-bit 2's complement binary numbers.
- <u>Subtraction</u> is nothing but addition of the 2's complement.
  - Ex. 58 – 40 = 58 + (-40) = 18

discard carry

$$
\begin{array}{r}
1\ 1\ 1\ 1 \\
00111010 \\
+11011000 \\
\hline
\boxed{1}\ 00010010
\end{array}
$$

Addition of 2's complement binary numbers

- Carry vs. overflow when adding A + B
  - If A and B are of opposite sign, then overflow cannot occur.
  - If A and B are of the same sign but the result is of the opposite sign, then overflow has occurred (and the answer is therefore incorrect).
    - Overflow occurs iff the carry into the sign bit differs from the carry out of the sign bit.

Addition of 2's complement binary numbers

- Consider 8-bit 2's complement binary numbers.
  - Then the msb (bit 7) is the sign bit. If this bit is 0, then this is a positive number; if this bit is 1, then this is a negative number.
  - Addition of 2 positive numbers.
  - Ex. 40 + 58 = 98

$$
\begin{array}{r}
1\,1\,1 \\
00101000 \\
+\underline{00111010} \\
01100010
\end{array}
$$

江西理工大學
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Addition of 2's complement binary numbers

- Consider 8-bit 2's complement binary numbers.
  - Addition of a negative to a positive.
  - What are the values of these 2 terms?
    - -88 and 122
    - -88 + 122 = 34

$$
\begin{array}{r}
1\ 1\ 1\ 1 \\
10101000 \\
+\,01111010 \\
\hline
1\ 00100010
\end{array}
$$

Addition of 2's complement binary numbers

- Carry vs. overflow when adding A + B
  - If A and B are of opposite sign, then overflow cannot occur.

  - If A and B are of the same sign but the result is of the opposite sign, then overflow has occurred (and the answer is therefore incorrect).

    - Overflow occurs iff the carry into the sign bit differs from the carry out of the sign bit.

Assignment

Example 1
Calculate, using binary numbers:

(a)	111 – 101	10
(b)	110 – 11	11
(c)	1100 – 101	111

Example 2
Calculate the binary numbers:

(a)	111 + 100	1011
(b)	101 + 110	1011
(c)	1111 + 111	10110

Reference

https://www.tutorialspoint.com/computer_logical_organization/binary_arithmetic.htm

The CARRY flag and OVERFLOW flag in binary arithmetic, - Ian! D. Allen - idallen@idallen.ca - www.idallen.com