Signed Integer Representations

Sign magnitude: the most significant bit is assigned to the algebraic sign.

$3_{10} = 0011 \qquad -3_{10} = 1011$

Offset binary: subtract half the largest possible number to get the value represented. I.e., you use half the largest number as the "zero" of the scale. For four bits:

Largest number:	1111	To get +3	1000	To get -3	1000
Subtract 7	-0111		0011		-0011
Working zero	1000		1011		0101

$3_{10} = 1011 \qquad -3_{10} = 0101$

2's Complement: negative integer is the complement of the positive integer plus one.

$3_{10} = 0011 \qquad -3_{10} = 1101$

Integer +3	0011
Complement	1100
Add 1	1101 =-3

Other Integer Representations

Binary Coded Decimal (BCD or 8421 BCD): each individual digit of the decimal number is represented by a 4-bit binary number.

$28_{10} = 0010 \quad 1000$

Excess-3: Add 3 to the number, then represent by 4-bit binary number.

$28_{10} = 0101 \quad 1011$

4221 Code: Four bits represent 4,2,2,1 instead of 8,4,2,1

$28_{10} = 0010 \quad 1110$

Gray Code: Starting at zero, make one change in the least significant possible bit to take you to the next state

$28_{10} = 0011 \quad 1100$

Number Systems

Index

Electronics concepts

Digital Circuits

HyperPhysics*****Electricity and magnetism

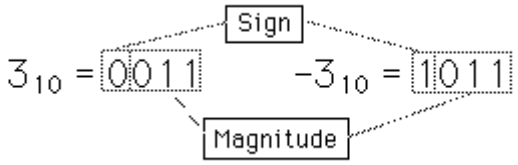
R Nave

Go Back

"Sign Magnitude" Representation

The most significant bit of a binary number can be used to represent the sign of the number, using the other bits to represent its magnitude. It is a natural kind of representation of signed integers, but is not used much because it is awkward for doing arithmetic compared to a system like 2's complement. There are also two zeros, since + and - zero will have a different code.

Example:



Number Systems

Index

Electronics concepts

Digital Circuits

Offset Binary Representation

One logical way to represent signed integers is to have enough range in binary numbers so that the zero can be offset to the middle of the range of positive binary numbers. Then the magnitude of a negative binary number can be simply subtracted from that zero point. This system has the advantage of a simple binary progression from negative to positive numbers. It is useful for binary counters and for A/D and D/A conversion. It is awkward for computation compared to the 2's complement representation.

As a simple example, consider a 4-bit binary number with maximum number 15. Subtracting 7 gives the following representation:

Index

Electronics concepts

Digital Circuits

Largest number: 1111 To get +3 1000 To get -3 1000 Subtract 7 -0111 0011 -0011 Working zero 1000 1011 0101 $3_{10} = 1011$ $-3_{10} = 0101$ Number Systems	
HyperPhysics*****Electricity and magnetism R Nave	Go Back

2's Complement Representation

This code is the most widely used code for integer computation. Positive numbers are represented by unsigned binary numbers. Negative numbers are formed by the following procedure.

To produce the number -5:

1. Write the binary number for 5 0101
2. Take the complement of it 1010
3. Add 1 1011

This code has the advantage that arithmetic operations are straightforward. Subtraction is accomplished by binary addition of the 2's complement. Multiplication also can be carried out.

```
 7   0111      2   0010
-5   1011     -5   1011
=2   0010     =-3  1101

 2   0010
x-3  1101
     0010
    0000
   0010
  0010
=-6 11010
```

Number Systems

Index

Electronics concepts

Digital Circuits

HyperPhysics*****Electricity and magnetism R Nave	Go Back