**Number Representation, Number bases, and Data Representation**

**Indices**

The power of a number is indicated by an index, sometimes called an exponent. e.g.

a4 = a x a x a x a

**Index Rules**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Rule** | | |  |  | **Example** | | |
| am.an | = | a(m + n) |  |  | a3.a5 | = | a8 |
| am  an | = | a(m - n) |  |  | a6  a4 | = | a2 |
| (am)n | = | am.n |  |  | (a4)2 | = | a8 |
| a-m | = | 1  am |  |  | a-6 | = | 1  a6 |
| a0 | = | 1 |  |  | 40 | = | 1 |
| a1/2 | = | √a |  |  | 91/2 | = | √9 = 3 |
| a1/3 | = | 3√a |  |  | 81/3 | = | 3√8 = 2 |

**Scientific Notation**

This is also known as standard form. There is one digit before the decimal point and the remaining significant figures after it. The number is multiplied by a power of 10 to give it its correct value. e.g.

70 = 7 x 10

700 = 7 x 100 = 7 x 10 x 10 = 7 x 102

7200 = 7.2 x 1000 = 7.2 x 10 x 10 x 10 = 7.2 x 103

Small numbers are represented using negative powers of ten. e.g.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.006 | = | 6  1000 | = | 6  10 x 10 x 10 | = | 6  103 | = | 6 x 10-3 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.000083 | = | 8.3  100000 | = | 8.3  10 x 10 x 10 x 10 x 10 | = | 8.3  105 | = | 8.3 x 10-5 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.000000092 | = | 9.2  100000000 | = | 9.2  10 x 10 x 10 x 10 x 10 x 10 x 10 x 10 | = | 9.2  108 | = | 9.2 x 10-8 |

This representation of numbers is known as scientific notation.

**Examples**

Represent the following numbers in scientific notation

* 0.000000102 = 1.02 x 10-7
* 1.045600000000 = 1.0456 x 100
* 189,000,000,000,000,000,000,000 = 1.89 x 1023
* 2,300,000 = 2.3 x 106

**Engineering Notation**

Engineering Notation is similar to [scientific notation](http://staffweb.cms.gre.ac.uk/~sp02/numberbases/lecture.html#Scientific) except the powers are restricted to multiples of three and symbols are used to represent these powers. i.e.

|  |  |
| --- | --- |
| Y | Yotta 1024 |
| Z | Zetta 1021 |
| E | Exa 1018 |
| P | Peta 1015 |
| T | Tera 1012 |
| G | Giga 109 |
| M | Mega 106 |
| k | kilo 103 |
| m | milli 10-3 |
|  | micro 10-6 |
| n | nano 10-9 |
| p | pico 10-12 |
| f | femto 10-15 |
| a | atto 10-18 |
| z | zepto 10-21 |
| y | yocto 10-24 |
| v | vimto 10-27 |

**Examples**

|  |  |  |
| --- | --- | --- |
| **Number** | **Scientific Notation** | **Engineering Notation** |
| 45000000 | 4.5 x 107 | 45 M |
| 80000 | 8 x 104 | 80 k |
| 189000000000000000000000 | 1.89 x 1023 | 189 Z |
| 0.00000065 | 6.5 x 10-7 | 650 |

**Using SI Symbols**

* A symbol is never followed by a period (unless, of course, it happens to fall at the end of a sentence).
* The letter "s" is never added to a symbol to indicate a plural. - In other words, 2 minutes is written 2 min and not 2 mins.
  + Symbols are case-sensitive and must be written as they are defined.
  + There is a tradition in the metric system that the first (or only) letter of an unprefixed unit symbol is capitalized if (and only if) the unit's name comes from a proper name. Thus W is the symbol for the watt and A is the symbol for the ampere, because these units are named for scientists.
  + It makes a big difference whether a symbol is capitalized or not, because often the same letter represents different units: t stands for the tonne and T for the tesla, for example.
  + There is one loophole in the rule on capitalization: it's acceptable to use the symbol L instead of l for the liter, since the letter l is so easily confused with the number 1.
* The case of symbol prefixes is specified, upper and lower, and must not be changed. For example, the symbol for kilo- is k-, so kW and not KW is the symbol for the kilowatt.
* The superscripts 2 and 3 are always used for "square" and "cubic", respectively. - Thus the square kilometer, for example, is written km2, not sq km.
* A raised dot (also called a middle dot or half-high dot) is recommended when symbols are multiplied. It is permissible to use a space instead, but symbols should not be placed next to one another with nothing between them. - For example, A·h is the recommended symbol for the ampere hour. A h is also permitted, but not Ah or amp hr
* The slash (solidus) / is used for "per". Furthermore, only one slash is allowed per symbol. - This means the SI unit of acceleration is written m/s2 rather than m/s/s, even though it is often spoken "meters per second per second". (Negative exponents can also be used: m/s2 can be written m·s-2.)
* Symbols are separated from the numerical quantity they follow by a space. Thus 5 kilograms is written 5 kg, not 5kg.

**Common Mistakes when using Engineering Notation and SI Notation.**

* Using invented abbreviations instead of correct international symbols
* Using a wrong symbol
* Pluralising symbols
* Writing symbols in italics
* Getting the case wrong - i.e. capitals instead of lower case (or vice versa)
* Omitting the oblique stroke (forward slash) in quoting prices
* Omitting the space between number and symbol

**Examples**

|  |  |  |
| --- | --- | --- |
| **Mistake** | **Why is it incorrect?** | **Correct version** |
| The temperature was 25C | C is the symbol for coulomb (a unit of electrical charge). Should use correct symbol. Also no space between number and symbol. | The temperature was 25 ºC |
| The speed limit is 50KPH | Non-standard abbreviation (language dependent). Should use international symbol and leave space after number. | The speed limit is 50 km/h |
| Cathedral 2Kms | Symbol should be lower case and not pluralised. Should leave space between number and symbol. | Cathedral 2 km |
| Contents 5 LTRS | LTRS is a clumsy, invented abbreviation. Should use symbol L (not pluralised). | Contents 5 L |
| 3kw heater | Symbol for “watt” is always upper case (capital) - even when combined with a prefix. Also there should be a space between number and symbol. | 3 kW heater |
| Printed on 90gsm paper | Non-standard invented abbreviation. Should use correct symbol g/m2. Also there should be a space between number and symbol. | Printed on 90 g/m2 paper |
| Frequency is measured in Hertz. | The SI convention is that all SI units, when written in full, are written entirely in lower case letters. | Frequency is measured in hertz. |
| Set the meter to the 20 Volt  range.  Set the power supply to 15 v | The SI convention is that all SI units, when written in full, are written entirely in lower case letters. For those units named in honour of individuals only their symbol is capitalised | Set the meter to the 20 volt  range.  Set the power supply to 15 V |

Source - http://ukma.org.uk/docs/ukma-style-guide.pdf

Memory on Computer Stores

* Program
* Data

Data Types

* Boolean or Logical data
* Integer
* Floating-point / Real numbers
* Character data
* Address or Reference

**Number Classifications**

* Integer

Unsigned or Cardinal numbers (zero or positive) Signed (allowing positive and negative values).

e.g. 1897, 3, -5, 789 -12 -7, 0

* Real Numbers, Floating point numbers

e.g.0.356, -567.0, 0.000987, 1457.8

* Examples

89.0

0

89

-0

12.45

- 0.000000007

0.0

Decimal Number System

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Consider

The number 321

|  |  |  |  |
| --- | --- | --- | --- |
| 103 | 102 | 101 | 100 |
| 0 | 3 | 2 | 1 |

321 = (0 x 1000) + (3 x 100) + (2 x 10) + (1 x 1)

Consider

The number 1567

|  |  |  |  |
| --- | --- | --- | --- |
| 103 | 102 | 101 | 100 |
| 1 | 5 | 6 | 7 |

1567 = (1 x 1000) + (5 x 100) + (6 x 10) + (7 x 1)

Computers process and store information as binary numbers

Binary numbers can be stored using only two states - on or off.

Binary Representation

|  |  |
| --- | --- |
| 0 | 0 |
| 1 | 1 |
| 2 | 10 |
| 3 | 11 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |
| 8 | 1000 |
| 9 | 1001 |
| 10 | 1010 |

Consider 100110002

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

100110002 = (1 x 27) + (0 x 26) + (0 x 25) + (1 x 24) + (1 x 23) + (0 x 22) + (0 x 21) + (0 x 20)

100110002 = (1 x 128) + (0 x 64) + (0 x 32) + (1 x 16) + (1 x 8) + (0 x 4) + (0 x 2) + (0 x 1)

100110002 = 128 + 16 + 8 = 15210

Consider 111012

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 24 | 23 | 22 | 21 | 20 |
| 1 | 1 | 1 | 0 | 1 |

111012 = (1 x 24) + (1 x 23) + (1 x 22) + (0 x 21) + (1 x 20)

111012 = (1 x 16) + (1 x 8) + (1 x 4) + (0 x 2) + (1 x 1)

111012 = 16 + 8 + 4 + 1 = 2910

Converting Decimal to Binary

Example: Convert 2310 to binary

|  |  |  |
| --- | --- | --- |
|  | Quotient | remainder |
| 23 / 2 | 11 | 1 |
| 11 / 2 | 5 | 1 |
| 5 / 2 | 2 | 1 |
| 2 / 2 | 1 | 0 |
| 1 / 2 | 0 | 1 |

2310 = 101112

Example: Convert 56810 to binary

|  |  |  |
| --- | --- | --- |
|  | Quotient | remainder |
| 568 / 2 | 284 | 0 |
| 248 / 2 | 142 | 0 |
| 142 / 2 | 71 | 0 |
| 71 / 2 | 35 | 1 |
| 35 / 2 | 17 | 1 |
| 17 / 2 | 8 | 1 |
| 8 / 2 | 4 | 0 |
| 4 / 2 | 2 | 0 |
| 2 / 2 | 1 | 0 |
| 1 / 2 | 0 | 1 |

56810 = 10001110002

There are and infinite number of other bases, However, computer scientists are mainly interested in:

* Binary
* Octal
* Decimal
* Hexadecimal

Octal Representation

|  |  |  |
| --- | --- | --- |
| Decimal | Binary | Octal |
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 10 | 2 |
| 3 | 11 | 3 |
| 4 | 100 | 4 |
| 5 | 101 | 5 |
| 6 | 110 | 6 |
| 7 | 111 | 7 |
| 8 | 1000 | 10 |
| 9 | 1001 | 11 |
| 10 | 1010 | 12 |

Consider 7528

|  |  |  |
| --- | --- | --- |
| 82 | 81 | 80 |
| 7 | 5 | 2 |

7528 = (7 x 82) + (5 x 81) + (2 x 80)

7528 = (7 x 64) + (5 x 8) + (2 x 1)

7528 = 448 + 40 + 2 = 49010

7528 = 49010

Consider 602418

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 84 | 83 | 82 | 81 | 80 |
| 6 | 0 | 2 | 4 | 1 |

602418 = (6 x 84) + (0 x 83) + (2 x 82) + (4 x 81) + (1 x 80)

602418 = (6 x 4096) + (0 x 512) + (2 x 64) + (4 x 8) + (1 x 1)

602418 = 24576 + 128 + 32 + 1 = 2473710

Converting Decimal to Octal

Example: Convert 12510 to Octal

|  |  |  |
| --- | --- | --- |
|  | Quotient | remainder |
| 125 / 8 | 15 | 5 |
| 15 / 8 | 1 | 7 |
| 1 / 8 | 0 | 1 |

12510 = 1758

Example: Convert 296510 to Octal

|  |  |  |
| --- | --- | --- |
|  | Quotient | remainder |
| 2965 / 8 | 370 | 5 |
| 370 / 8 | 46 | 2 |
| 46 / 8 | 5 | 6 |
| 5 / 8 | 0 | 5 |

296510 = 56258

Consider:

Convert 1758 to Binary

|  |  |
| --- | --- |
| Octal | Binary |
| 0 | 0 |
| 1 | 1 |
| 2 | 10 |
| 3 | 11 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

|  |  |  |
| --- | --- | --- |
| 1 | 7 | 5 |
| 1 | 111 | 101 |

1758 = 11111012

Convert 10011001112 to Octal

|  |  |
| --- | --- |
| Octal | Binary |
| 0 | 0 |
| 1 | 1 |
| 2 | 10 |
| 3 | 11 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 001 | 100 | 111 |
| 1 | 1 | 4 | 7 |

10011001112 = 11478

Hexadecimal Representation

|  |  |  |  |
| --- | --- | --- | --- |
| Decimal | Binary | Octal | Hexadecimal |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 10 | 2 | 2 |
| 3 | 11 | 3 | 3 |
| 4 | 100 | 4 | 4 |
| 5 | 101 | 5 | 5 |
| 6 | 110 | 6 | 6 |
| 7 | 111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |

Consider A8F216

|  |  |  |  |
| --- | --- | --- | --- |
| 163 | 162 | 161 | 160 |
| A | 8 | F | 2 |

A8F216 = (A x 163) + (8 x 162) + (F x 161) + (2 x 160)

A8F216 = (A x 4096) + (8 x 256) + (F x 16) + (2 x 1)

A8F216 = 40960 + 2048 + 240 + 2 = 4326410

A8F216 = 4325010

Converting Decimal to Hexadecimal

Example: Convert 4766310 to Hexadecimal

|  |  |  |
| --- | --- | --- |
|  | Quotient | remainder |
| 47663 / 16 | 2978 | F |
| 2978 / 16 | 186 | 2 |
| 186 / 16 | 11 | A |
| 11 / 16 | 0 | B |

4766310 = BA2F16

Consider:

Convert F81C16 to Binary

|  |  |
| --- | --- |
| Hexadecimal | Binary |
| 0 | 0 |
| 1 | 1 |
| 2 | 10 |
| 3 | 11 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

|  |  |  |  |
| --- | --- | --- | --- |
| F | 8 | 1 | C |
| 1111 | 1000 | 0001 | 1100 |

F81C16 = 1111,1000,0001,11002

Convert 111001110012 to Hexadecimal

|  |  |
| --- | --- |
| Hexadecimal | Binary |
| 0 | 0 |
| 1 | 1 |
| 2 | 10 |
| 3 | 11 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

|  |  |  |
| --- | --- | --- |
| 111 | 0011 | 1001 |
| 7 | 3 | 9 |

111001110012 = 73916

Enter 79867510 on your calculator and press convert to binary.

The solution is 1100,0010,1111,1101,0011

Why could the calculator not solve the problem?

Unique Combinations = 2n

e.g 2 bit register 22 = 4

|  |  |
| --- | --- |
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

e.g 3 bit register 23 = 8

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

* Crumb 2 bits
* Nibble 4 bits
* Byte 8 bits
* Word 16 bits
* Long 32 bits

Consider the following binary addition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 0 | 1 | 1 |
|  | 1 | 0 | 1 | 0 |
|  | | | | |
| 1 | 0 | 1 | 0 | 1 |
|  | | | | |
| 1 |  | 1 |  |  |

Condition Flags:- (Condition Code Register) -68000

* Negative
* Zero
* Overflow
* Carry

Numbers Representations

* Sign and Magnitude
* 1's Complement
* 2's Complement

All arithmetic operations are performed by an adder circuit: Consider

|  |  |  |  |
| --- | --- | --- | --- |
| b3b2b1b0 | Sign and Magnitude | 1's Complement | 2's Complement |
| 0111 | +7 | +7 | +7 |
| 0110 | +6 | +6 | +6 |
| 0101 | +5 | +5 | +5 |
| 0100 | +4 | +4 | +4 |
| 0011 | +3 | +3 | +3 |
| 0010 | +2 | +2 | +2 |
| 0001 | +1 | +1 | +1 |
| 0000 | +0 | +0 | +0 |
| 1000 | -0 | -7 | -8 |
| 1001 | -1 | -6 | -7 |
| 1010 | -2 | -5 | -6 |
| 1011 | -3 | -4 | -5 |
| 1100 | -4 | -3 | -4 |
| 1101 | -5 | -2 | -3 |
| 1110 | -6 | -1 | -2 |
| 1111 | -7 | -0 | -1 |

Two's compliment is the normal standard.

To calculate: - invert and add 1

Example

Find the 2' compliment of C16 using an eight bit register.

|  |  |
| --- | --- |
| 00001100 | C: Unsigned |
| 11110011 | C: 1's Complement |
| 11110100 | C: 2's Complement |

The representable range of signed integers in an N-bit register is:

- 2(n - 1) to + 2(n - 1) - 1

So for an 8-bit register, the range is -128 to +127; for a 16-bit register, -32768 to +32767.

Why use 2's compliment

Consider the following examples using 8 bit registers

|  |  |  |
| --- | --- | --- |
| 16 |  | 00010000 |
| -5 |  | 11111011 |
|  |  |  |
| 11 |  | 00001011 |

|  |  |  |
| --- | --- | --- |
| 16 |  | 00010000 |
| -31 |  | 11100001 |
|  |  |  |
| -15 |  | 11110001 |

|  |  |  |
| --- | --- | --- |
| 64 |  | 01000000 |
| 127 |  | 01111111 |
|  |  |  |
| -65 |  | 10111111 |

the answer is negative (-65) and wrong

overflow flag set as warning

Can catch programmers unaware

One circuit - an adder circuit can perform all functions.

* Addition
* Subtraction
* Multiplication
* Division

**BCD**

|  |  |
| --- | --- |
| Decimal Digit | BCD Code |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

Floating Point Numbers

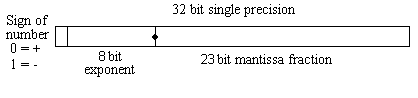
Fractions can be represented by binary numbers: Consider:-

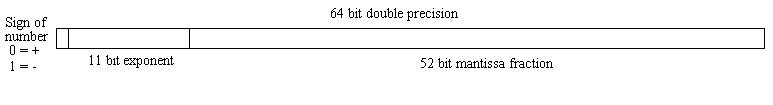
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2-1 | 2-2 | 2-3 | 2-4 | 2-5 | 2-6 | 2-7 | 2-8 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

100110012 = (1 x 2-1) + (0 x 2-2) + (0 x 2-3) + (1 x 2-4) + (1 x 2-5) + (0 x 2-6) + (0 x 2-7) + (1 x 2-8)

100110012 = (1 x 0.5) + (0 x 0.25) + (0 x 0.125) + (1 x 0.0625) + (1 x 0.03125) + (0 x 0.015625) + (0 x 0.0078125) + (1 x 0.00390625)

100110002 = 0.5 + 0.0625 + 0.03125 + 0.00390625 = 0.5976562510





[More on Floating Point Numbers](http://staffweb.cms.gre.ac.uk/~sp02/numberbases/FloatingPointArithmetic.html)

* Integers: Fast, Limited Range
* Floating point: Slow

**ASCII**

The ASCII character set defines 128 characters (0 to 127 decimal, 0 to FF hexadecimal).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Bit positions 654 | | | | | | | |
| Bit Positions 3210 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 0000 | NUL | DLE | SPACE | 0 | @ | P | ' | p |
| 0001 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 0010 | STX | DC2 | " | 2 | B | R | b | r |
| 0011 | ETX | DC3 | # | 3 | C | S | c | s |
| 0100 | EOT | DC4 | $ | 4 | D | T | d | t |
| 0101 | ENQ | NAK | % | 5 | E | U | e | u |
| 0110 | ACK | SYN | & | 6 | F | V | f | v |
| 0111 | BEL | ETB | ' | 7 | G | W | g | w |
| 1000 | BS | CAN | ( | 8 | H | X | h | x |
| 1001 | HT | EM | ) | 9 | I | Y | i | y |
| 1010 | LF | SUB | \* | : | J | Z | j | z |
| 1011 | VT | ESC | + | ; | K | [ | k | { |
| 1100 | FF | FS | , | < | L | \ | l | | |
| 1101 | CR | GS | - | = | M | ] | m | { |
| 1110 | SO | RS | . | > | N | ^ | n | ~ |
| 1111 | SI | US | / | ? | O | \_ | o | DEL |

**Control Characters**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Char** | **Control Action** | **Char** | **Control Action** | **Char** | **Control Action** | **Char** | **Control Action** |
| NUL | Null character | BS | Backspace | DLE | Data link escape | CAN | Cancel (error in data) |
| SOH | Start of heading | HT | Horizontal tab | DC1 | Device control 1 | EM | End of medium |
| STX | Start of text | LF | Line Feed | DC2 | Device control 2 | SUB | Substitute |
| ETX | End of text | VT | Vertical tab | DC3 | Device control 3 | ESC | Escape |
| EOT | End of transmission | FF | Form Feed | DC4 | Device control 4 | FS | File separator |
| ENQ | Enquiry | CR | Carriage Return | NAK | Negative acknowledge | GS | Group separator |
| ACK | Acknowledge | SO | Shift Out | SYN | Synchronous idle | RS | Record separator |
| BEL | Bell | SI | Shift In | ETB | End transmission block | US | Unit separator |