Data Representation

Tuesday, 3 October 2023 8:47 AM

Number Systems

1. Decimal number system (*also called Denary*) | base-10 | digits from 0 > 9
2. Binary number system | base-2 | digits 0 &1
3. Octal number system | base-8 | digits from 0 > 7

4. Hexadecimal number system | base-16 | digits from 0 > 9 & then A > F. 16 digits in total! :3

Hexadecimal *base-16*	Decimal *base-10*	Binary *base-2*
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

↳ To convert from Decimal → Hexadecimal

remainder

16 | 138 — 10 or A — $(138)_{10} \rightarrow (8A)_{16}$

8

16 | 278 — 6 — $(278)_{10} \rightarrow (116)_{16}$

16 | 17 — 1

1

16 | 78 — 14 or E — $(78)_{10} \rightarrow (4E)_{16}$

4

↳ To convert from hexadecimal → decimal

$(AB)_{16} \rightarrow (171)_{10}$

A B

or $10 \times 16^1 + \text{or } 11 \times 16^0$

$160 + 11$

$= 171$

To convert from binary → hexadecimal

group it into sets of 4, starting from the right & if needed, adding extra 0s to the left.

$(0001, 0110, 0110, 1011)_2 \rightarrow (166B)_{16}$

1 6 6 11

1 6 6 B

using your knowledge of all the binary → decimal digits (until decimal 15, hexadecimal F, binary 1111) for each set of 4!

using your knowledge of all the decimal → hexadecimal digits for each.

A 4 digit set in binary would be what gives you the last single hexadecimal digit! C(15)

↕ opposite process :)

To convert from hexadecimal → binary

$(2D6F)_{16} \rightarrow (0010\ 1101\ 0110\ 1111)_2$

2 D 6 F

2 13 6 15

0010 , 1101 , 0110 , 1111

make sure to add extra 0s if needed, to make each set have 4 digits

To add 2 binary numbers directly :)

1101 + 11

1 1 0 1

\+ 1 1 is 2 or 10

1 1

1 0 0 0 0

How numbers are stored in a computer:
Binary numbers are stored in 8 bit+

10 in binary is 1010

it's stored as 00001010 (8 bits!)

here, only 7 bits can be used to store the number itself, leftmost 8th one stores + or - .
2^7 numbers can be represented, 0 → 127 ... that's 01111111
128

-10 in binary is stored as
10001010

The left most digit represents + or -

ASCII - Representation of text in computers:

ASCII, or American Standard Code for Information Interchange, is a character encoding standard for electronic communication.

It is a **7-bit character encoding**, meaning that each character is represented by a unique 7-bit sequence. This allows ASCII to represent a total of 128 different characters.

0 to 9: 48 to 57
A to Z: 65 to 90
a to z: 97 to 122

UNICODE:
It's an established standard for data representation - a single encoding scheme for all languages and characters. The UNICODE data can be used throughout different systems, platforms, and devices (great compatibility).

The difference between UNICODE, and ASCII:
1. Bits used to encode: 32 (UTF 32), 7.
2. Characters representable: over 1 million, 128

The different encoding standards of Unicode are:
UTF-8, UTF-16, UTF-32 *(Unicode Transformation Format!)*

Pixel Colour Representation

R G B
1 1 1

each R, G, & B value of a pixel is represented from 0 → 255
each is stored in an 8 bit value, so 2^8 or 256 possible values.
2 hexadecimal digits can represent

256 × 256 × 256 total possible colours a pixel can (generally is) display.