Computer Arithmetic

Modulo Arithmetic: produces the integer value that is the remainder of an integer division.

1 --> 9
Goes up to a certain point then restarts. Examples:

a) Odometer: 99999 + 1 --> start over.
b) Clock: 10 + 4 --> 12.

Human beings traditionally use ten as the number to count with (Base 10).

[Babylon apparently used sixty (Base 60), a sexaqesimal system. It can be divided evenly by two, three, four, five, six, ten, twelve, fifteen, twenty and thirty.]

The second digit, counting from the right, indicates the base.

For example:

Four-thousand five hundred sixty-seven

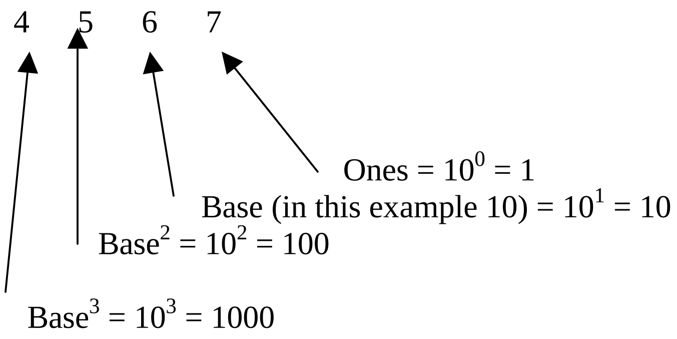
=

4567

=

4 thousands plus
5 hundreds plus
6 tens plus
7 ones

=



4 5 6 7

Ones = $10^0 = 1$

Base (in this example 10) = $10^1 = 10$

$\text{Base}^2 = 10^2 = 100$

$\text{Base}^3 = 10^3 = 1000$

$$= (4 \times 10^3) + (5 \times 10^2) + (6 \times 10^1) + (7 \times 10^{0})$$

The above example can be applied to any base.

As you saw in the video *Giant Brains*, Konrad Zuse decided to use the simplest types of switches, ON/OFF – Boolean, for his computer.

Everything inside a computer must be represented with some combination of ON and OFF. We represent ON = 1 and OFF = 0. Since only two symbols are used we refer to this system as <u>Base 2</u> = the **Binary System**.

Analogy with a chandelier:

2^7 = 128	2^6 = 64	2^5 = 32	2^4 = 16	2^3 = 8	2^2 = 4	2^1 = 2	2^0 = 1

For example:

1000 1001 Binary (Base 2)

=

Base 2 (Binary)	**Base 10 (Decimal)**
2^7 1 x 128	128
2^6 0 x 64	0
2^5 0 x 32	0
2^4 0 x 16	0
2^3 1 x 8	8
2^2 0 x 4	0
2^1 0 x 2	0
2^0 1 x 1	1
	Total: 137

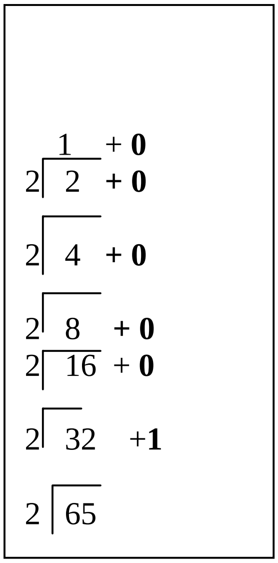
To Convert Decimal to Binary

Example: 65 decimal

2^7 = 128	2^6 = 64	2^5 = 32	2^4 = 16	2^3 = 8	2^2 = 4	2^1 = 2	2^0 = 1
0	1	0	0	0	0	0	1

Or

Using the Method of Division:



=

1000001

Binary Addition

Rules:

0 + 0 = 0
1 + 0 = 1
1 + 1 = 10
1 + 1 + 1 = 11

Example:

```
 10101101
  1011110
100001011
```

There are eight bits (ON/OFF states) in a Byte:

Byte = 8 Bits

Working with 4 Bit examples, add the following:

Binary	Decimal
0101	5
0010	2
0111	7

Binary	Decimal
0101	5
0011	3
1000	8

Binary	Decimal
0101	5
0110	6
1011	11

Binary	Decimal
0111	
1011	
10010	

Overflow! Using a 4 bit machine.

ASCII Code

(American Standard Code for Information Interchange)

Examples:

A = 65
B = 66
C = 67

a = 97
b = 98
c = 99

Exercise: Convert the above to Binary using an 8 bit system.

In lab we have 32 bit machines; (note: the definition of a byte does not change, it still equals 8 bits).

With more bits we can work with larger numbers.

Fore example:

2^8

=

2^8 = 256	2^7 = 128	2^6 = 64	2^5 = 32	2^4 = 16	2^3 = 8	2^2 = 4	2^1 = 2	2^0 = 1
1	0	0	0	0	0	0	0	0

We commonly refer to a **K** as being equal to 1000. However it really is 2^{10} = 1024 since we are dealing with powers of 2 not 10!

MB = a little over a million.
GB = a little over a billion.
TB = a little over a trillion.

Binary Subtraction

Example:

a-b = a +(-b)

Integers: negative and positive.
Binary: positive only (unsigned).

2's Complement

Decimal Example:

27 – 11

=

$27_{10} - 11_{10}$

=

$27_{10} + (-11_{10})$

= 16_{10}

A. Convert:

Decimal	Binary
27_{10}	011011_2
11_{10}	001011_2

B. Get 2’s Complement:

B1. Flip the Bits then add 1:
(Add a number to its complement and one gets all 0’s).

001011 (Original number)
110100 (Flip the Bits)
+ 1 (Add 1)
110101 (2’s Complement)

C. Add the two numbers:

011011
+ 110101
010000 (Restrict to 6 bits for answer; extra bit lost as overflow.

Example: Convert 56_{10} to binary, flip the bits, add 1 and add to its complement.

Hint: Any binary number + its complement = 0.

Example: $16 – 5 = 16_{10} – 5_{10} = 16_{10} + (-5_{10}) = 11_{10}$

1. Convert:

	2^5 = 32	2^4 = 16	2^3 = 8	2^2 = 4	2^1 = 2	2^0 = 1
16_{10} =	0	1	0	0	0	0
5_{10} =	0	0	0	1	0	1

2. Get 2's Complement:

000101	Original number (5_{10})
111010	Flip the bits
$\underline{+1}$	Add 1
111011	2's complement

3. Add the two numbers:

$\boxed{1}$	010000	16_{10}
	$\underline{111011}$	2's complement of 5_{10}
	001011	Answer (11_{10})
		(Hint: compare with Binary table)

Overflow

Example: $32 - 16 = 32_{10} - 16_{10} = 32_{10} + (-16_{10}) = 16_{10}$

1. Convert:

	2^6 = 64	2^5 = 32	2^4 = 16	2^3 = 8	2^2 = 4	2^1 = 2	2^0 = 1
32_{10} =	0	1	0	0	0	0	0
16_{10} =	0	0	1	0	0	0	0

2. Get 2's Complement:

0010000	Original number (16_{10})
1101111	Flip the bits
$\underline{+1}$	Add 1
1110000	2's complement

3. Add the two numbers:

$\boxed{1}$	0100000	32_{10}
	$\underline{1110000}$	2's complement of 16_{10}
	0010000	Answer (16_{10})

Overflow

Example: $56 – 16 = 56_{10} – 16_{10} = 56_{10} + (-16_{10}) = 40_{10}$

1. Convert:

	2^6 = 64	2^5 = 32	2^4 = 16	2^3 = 8	2^2 = 4	2^1 = 2	2^0 = 1
56_{10} =	0	1	1	1	0	0	0
16_{10} =	0	0	1	0	0	0	0

2. Get 2’s Complement:

0010000 Original number (16_{10})

1101111 Flip the bits
+1 Add 1
1110000 2’s complement

3. Add the two numbers:

$\boxed{1}$ 0111000 56_{10}
1110000 2’s complement of 16_{10}
0101000 Answer (40_{10})

Overflow

Hexadecimal (Hex)

- Base 16.
- For large numbers easier to work with than binary for most people.
- An integer on a 32 bit machine, (the type we use in lab), can be written as 4 hexadecimal digits.
- Letters are used for the numbers 10 → 15 as follows:

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

To interpret as base 10 numbers we need to know the powers of 16:

Base

16^3	16^2	16^1	16^0
4096	256	16	1

Example: $F29 = (F * 16^2) + (2 * 16^1) + (9 * 16^0)$
$= (15 * 256) + (2 * 16) + (9 * 1)$
$= 3881$ in base 10

An advantage of hexadecimal is how easy it is to convert from Base 2 (binary) and back.

To Convert:

Every hexadecimal digit is broken down into a 4 digit binary number. These digits are just written down in the same order as the hexadecimal number and one has the equivalent binary (base 2) number.

Example: F29 hexadecimal converted to binary

2^3	2^2	2^1	2^0
8	4	2	1

F = 15 = 1 1 1 1
2 = 0 0 1 0
9 = 1 0 0 1

1111 0010 1001 (base 2) = 3881 (base 10)

Example: E16 hexadecimal converted to binary

2^3	2^2	2^1	2^0
8	4	2	1

E = 14 = 1 1 1 0
1 = 0 0 0 1
6 = 0 1 1 0

1110 0001 0110 (base 2) = 3606 (base 10)

Example: A14 hexadecimal converted to binary

2^3	2^2	2^1	2^0
8	4	2	1

A = 10 = 1 0 1 0
1 = 0 0 0 1
4 = 0 1 0 0

1010 0001 0100 (base 2) = 2580 (base 10)

Example: C18 hexadecimal converted to binary

2^3	2^2	2^1	2^0
8	4	2	1

C = 12 = 1 1 0 0
1 = 0 0 0 1
8 = 1 0 0 0

1100 0001 1000 (base 2) = 3096 (base 10)

Example: Convert 3567_{10} from Decimal to Binary to Hexadecimal.

3567 (Decimal) =

2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
1	1	0	1	1	1	1	0	1	1	1	1

(Binary) =

DEF (Hexadecimal)

(Hint: group by four)

$3567_{10} = \underline{110\ 1}\ \underline{1110}\ \underline{1111}\ _2 = \text{DEF}\ _{16}$