第1单元 程序设计初步

第3讲 数据的表示与存储

哈尔滨工业大学
赵玲玲 *zhaoll@hit.edu.cn*

- 计算机要处理的信息是多种多样的
  - 文字、符号、图形、图像和语言、十进制数…
- 对于计算机来说，它们都是“一样”的
  - 都被用数表示的

二进制数

- Q1: 数在计算机里怎么表示的？

二进制（Binary）

$5 = 1*2^2+0*2^1+1*2^0$

$16 = 1*2^4+0*2^3+0*2^2+0*2^1+0*2^0$

……

二进制数

* 二进制

数码：0、1

基数：2

运算规律：逢二进一，借一当二

二进制数的权展开式：

$$(101.01)_B = 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$$

各数位的权是 2 的幂

二进制数

加法规则：0+0=0，0+1=1，1+0=1，1+1=10

乘法规则：0•0=0， 0 • 1=0 ，1 • 0=0，1 • 1=1

Exp1: $(1101.01)_2 + (11.01)_2 = ?$

$$
\begin{array}{r}
1101.01 \\
+\quad 11.01 \\
\hline
10000.10
\end{array}
$$

* Q2：为什么计算机用二进制而不是十进制来存储数据？

二进制数只有0和1两个数码，它的每一位都可以用电子元件来实现，且运算规则简单，相应的运算电路也容易实现。

二进制数

- Q3：负数怎么表示？

  二进制补码表示

- exp2：-1的二进制表示

符号位　　　　-1 的原码

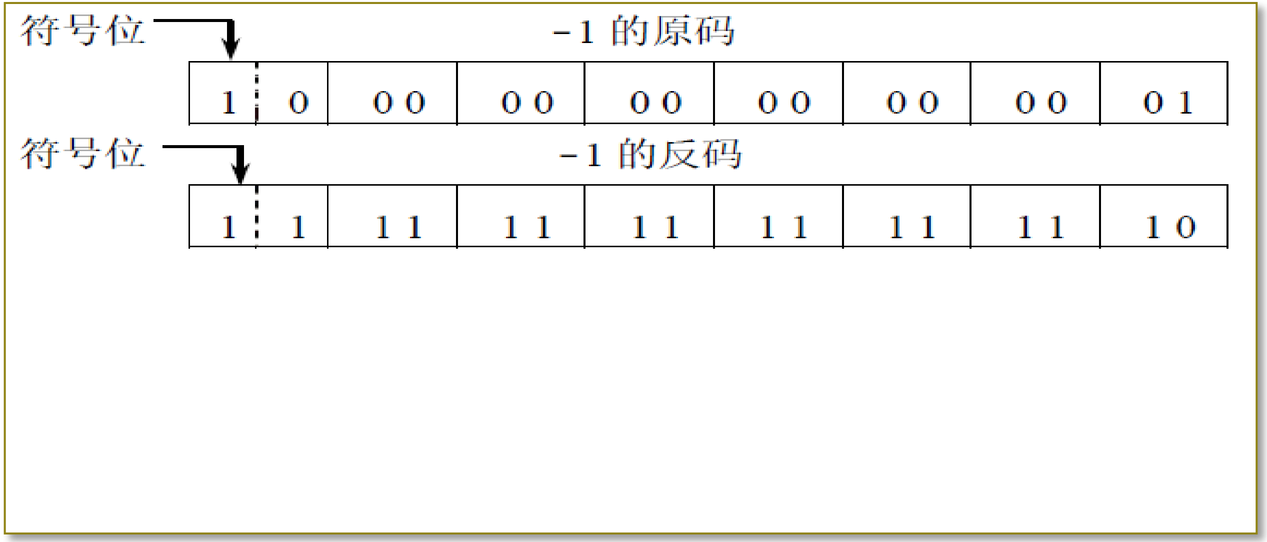
1	0	0 0	0 0	0 0	0 0	0 0	0 0	0 1

二进制数

- Q3：负数怎么表示？

  二进制补码表示

- exp2：-1的二进制表示



二进制数

- Q3：负数怎么表示?

  二进制补码表示

- exp2：-1的二进制表示

符号位 ↓ -1 的原码

1	0	0 0	0 0	0 0	0 0	0 0	0 0	0 1

符号位 ↓ -1 的反码

1	1	1 1	1 1	1 1	1 1	1 1	1 1	1 0

符号位 ↓ -1 的反码加 1 后的结果

1	1	1 1	1 1	1 1	1 1	1 1	1 1	1 1

二进制数

- Q3：负数怎么表示？

  二进制补码表示

- exp2：-1的二进制表示

符号位 -1 的原码

1	0	0 0	0 0	0 0	0 0	0 0	0 0	0 1

符号位 -1 的反码

1	1	1 1	1 1	1 1	1 1	1 1	1 1	1 0

符号位 -1 的反码加 1 后的结果

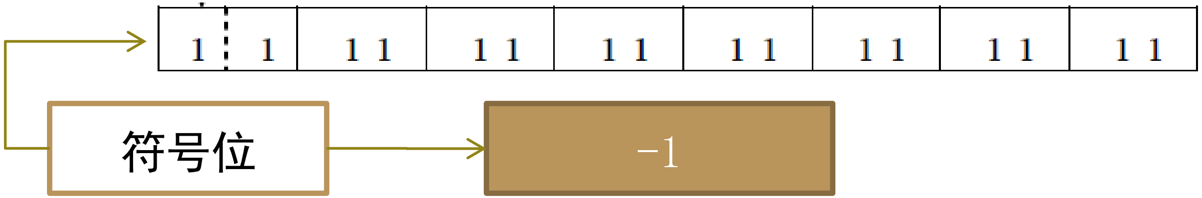
1	1	1 1	1 1	1 1	1 1	1 1	1 1	1 1

符号位 -1 的补码

1	1	1 1	1 1	1 1	1 1	1 1	1 1	1 1

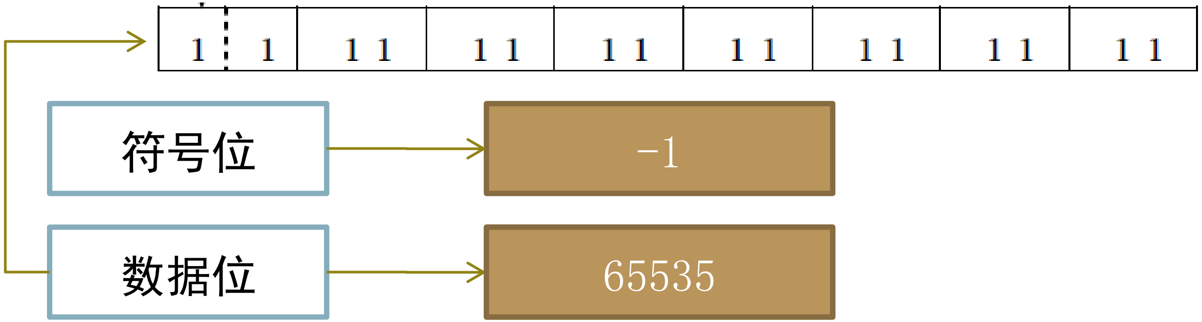
二进制数

- Q4:有符号数与无符号数有什么不同?



二进制数

- Q4:有符号数与无符号数有什么不同?



- Q4-1：16位无符号整数，表数范围?
- Q4-2：16位有符号整数，表数范围?
- Q4-3：这些数值怎么排列的?

二进制数

* Q4-1：16位无符号整数，表数范围？

00	00	00	00	00	00	00	00	0
00	00	00	00	00	00	00	01	1
			⋮					⋮
11	11	11	11	11	11	11	10	$2^{16}-2$
11	11	11	11	11	11	11	11	$2^{16}-1$

表数区间[0，$2^{16}-1$]，共2^{16}个数

二进制数

* Q4-2：16位有符号整数，表数范围？

00	00	00	00	00	00	00	00
⋮							
01	11	11	11	11	11	11	11
10	00	00	00	00	00	00	00

0

⋮

2^{15}–1：32767

?

表数区间[? ， ?]，共2^{16}个数

二进制数

- Q4-2：16位有符号整数，表数范围？

00	00	00	00	00	00	00	00	0
			⋮					
01	11	11	11	11	11	11	11	$2^{15}-1$：32767
10	00	00	00	00	00	00	00	-2^{15} ：-32768
10	00	00	00	00	00	00	01	$-2^{15}+1$：-32767
			⋮					
11	11	11	11	11	11	11	11	-1

表数区间[? ， ?]，共2^{16}个数

二进制数

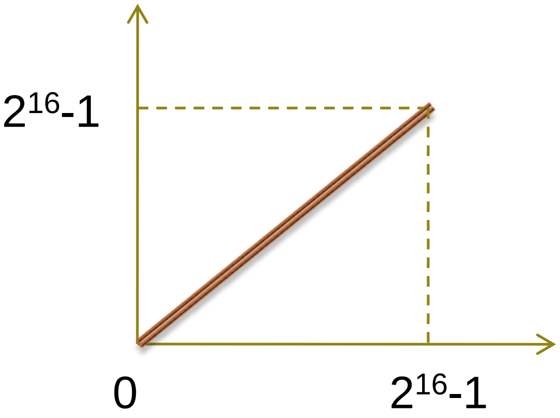
* Q4-2：16位有符号整数，表数范围？

00	00	00	00	00	00	00	00	0
⋮								
01	11	11	11	11	11	11	11	$2^{15}-1$：32767
10	00	00	00	00	00	00	00	-2^{15} ：-32768
10	00	00	00	00	00	00	01	$-2^{15}+1$：-32767
⋮								
11	11	11	11	11	11	11	11	-1

表数区间$[-2^{15}，2^{15}-1]$，共2^{16}个数

二进制数

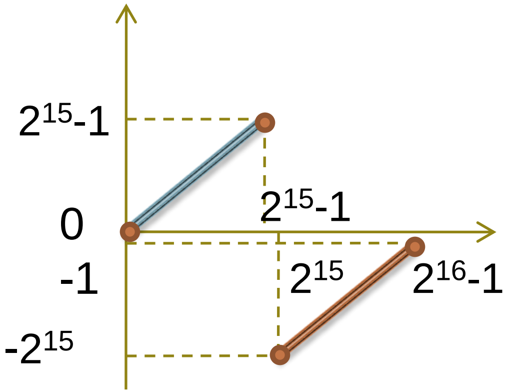
- Q4-3：这些数值怎么排列的？
  - 16位无符号数



00	00	00	00	00	00	00	00	0
00	00	00	00	00	00	00	01	1
⋮								⋮
11	11	11	11	11	11	11	10	$2^{16}-2$
11	11	11	11	11	11	11	11	$2^{16}-1$

二进制数

* Q4-3：这些数值怎么排列的？
  * 16位有符号数



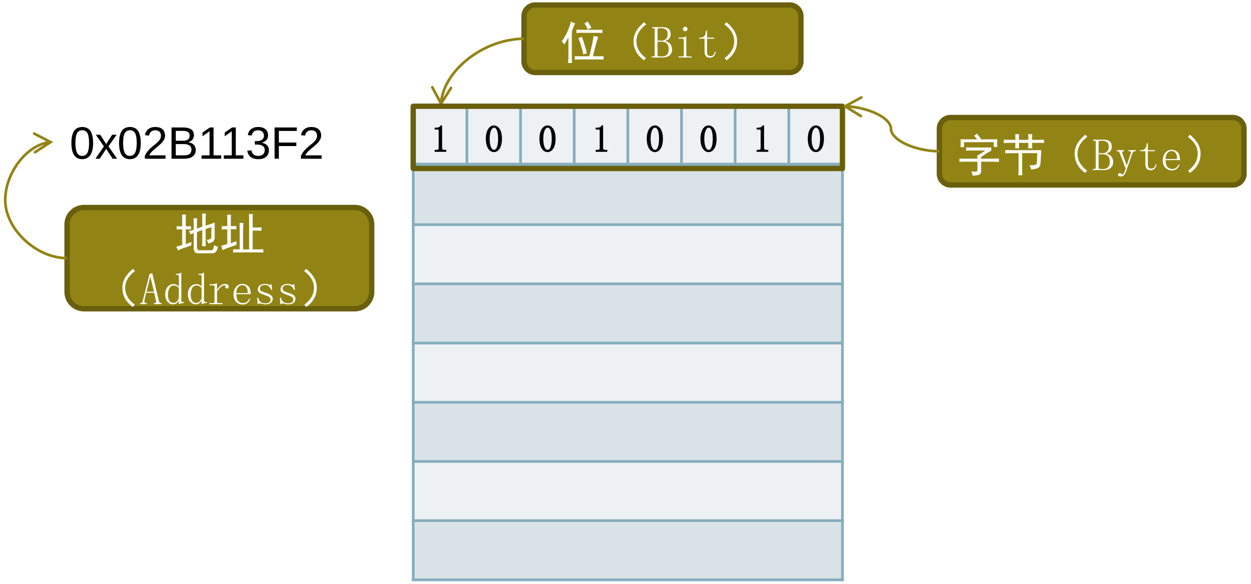
00	00	00	00	00	00	00	00	0
⋮								
01	11	11	11	11	11	11	11	$2^{15}-1$：32767
10	00	00	00	00	00	00	00	-2^{15} ：-32768
10	00	00	00	00	00	00	01	$-2^{15}+1$：-32767
⋮								
11	11	11	11	11	11	11	11	-1

Discussion:

为什么要采用补码来表达负数？

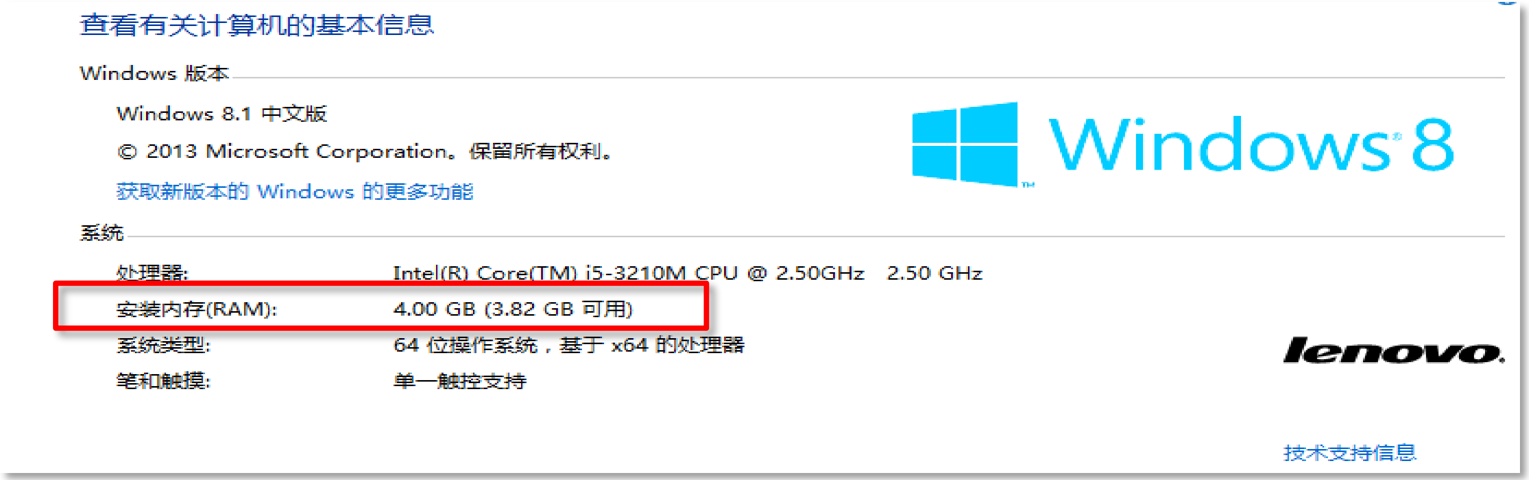
数据与内存

- Q1:计算机内存怎么组织?



数据与内存

- Q2:内存大小怎样衡量?



数据与内存

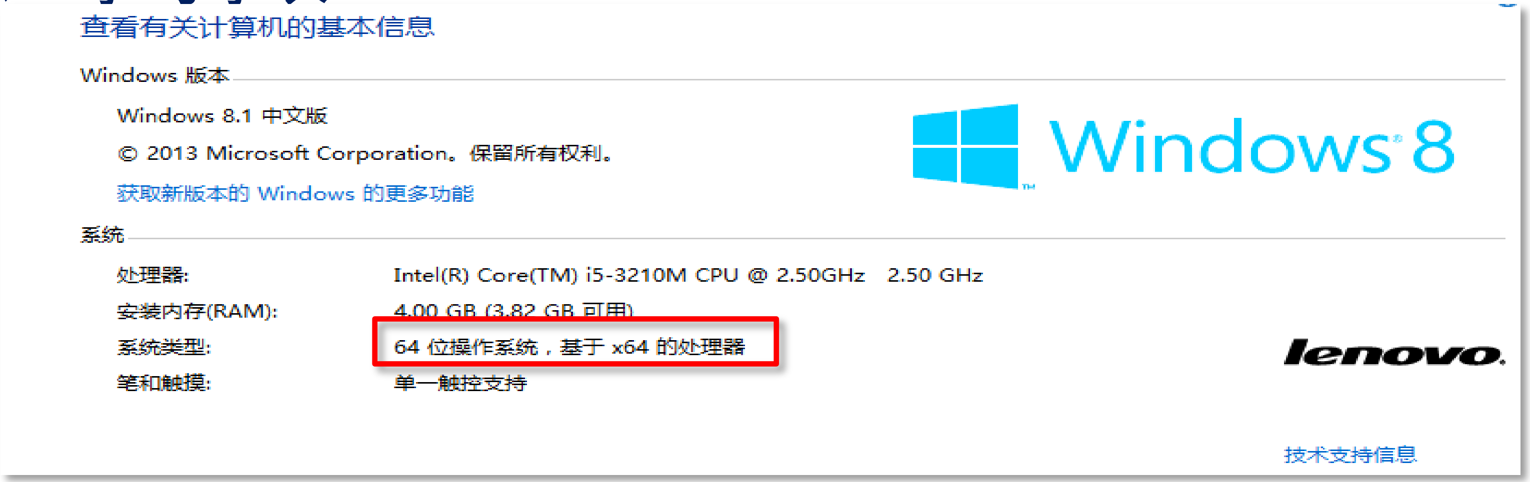
* Q3:内存大小怎样衡量?

英文称谓	容量大小（单位字节）	换算方法
B	2 的 0 次方	1 B == 8 b
KB	2 的 10 次方	1 KB == 1,024 B
MB	2 的 20 次方	1 MB == 1,024 KB
GB	2 的 30 次方	1 GB == 1,024 MB
TB	2 的 40 次方	1 TB == 1,024 GB
PB	2 的 50 次方	1 PB == 1024 TB
EB	2 的 60 次方	1 EB == 1024 PB
ZB	2 的 70 次方	1 ZB == 1024 EB
YB	2 的 80 次方	1 YB == 1024 ZB

4GB = 4 294 967 296 B
4GB/128MB = 32

数据与内存

* ## Q4:字与字长



在计算机中，一串数码是*作为一个整体来处理或运算*的，称为一个*计算机字，简称字（**Word**）*。计算机存储一个字所需的字节的长度，称为***字长（Word Size）***。

小数的表示

- **Q1：小数怎么表示？**
  - 定点数（**Fixed-Point Number**）:小数点位置固定不变
    - **定点小数（纯小数）**

N_0	●	N_{-1}	N_{-2}	N_{-3}	.	.	.	N_{-m}

符号位　固定小数点　数值部分

0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

表数范围为: $|N| <= 1-2^{-m}$

小数的表示

- **Q1：小数怎么表示？**
  - 定点整数

N_0		N_m	N_{m-1}	N_{m-2}	.	.	.	N_1	
符号位					数值部分				固定小数点

表数范围为 $|N| \le 2^m-1$

貌似不太好的地方:
表达的数值范围非常有限

小数的表示

- **Q2：怎样表达更大更精确的小数？**
  - 科学计数法：$N = R^E * M$ （Radix, Exponent, Mantissa）

    $123.456=0.123456 * 10^3$

  - 二进制科学计数法：$N = 2^E * M$

    $11011.01 = 2^{101} * 0.1101101$

  - 浮点数（Floating-Point Number）
    - 小数点位置不固定；
    - 表示整数部分、又有小数部分的实数（Real Number）

小数的表示

* Q2：怎样表达更大更精确的小数？

浮点数（Floating-Point Number）

$$N = 2^{E} * M$$

数符（Ms）	阶码(E)	尾数(M)
表示M的符号	表示小数点的位置 可采用补码、移码等形式	表示数的有效数值（即数字部分） 通常采用原码形式的二进制小数表示

阶码和尾数谁来决定范围，谁来决定精度？

小数的表示

* ## Q2：怎样表达更大更精确的小数？
  * Exp3. 采用8位阶码，23位规格化尾数时，阶码表数范围是多少？

二进制数	0000 0001~~0111 1111	0000 0000	1111 1111~~1000 0000
表示的值	1~~127	0	-1~~-128

阶码的表示范围为： [-128， 127]

小数的表示

* **Q2：怎样表达更大更精确的小数？**
  * Exp1. 采用8位阶码23位规格化尾数时，尾数表数范围是多少？
    * 尾数的规格化表示

$1.1001 = 0.0011001*2^{11}$　　$0.11001*2^{1}$　　$0.011001*2^{10}$

1）尾数用纯小数形式给出

2）当其值不为0时，其绝对值应大于或等于0.5，即最高位非零。

貌似应该储存的值	1000 0000 0000 0000 0000 000	1111 1111 1111 1111 1111 111

实际储存的值	0000 0000 0000 0000 0000 000	1111 1111 1111 1111 1111 111

小数的表示

* Q2：怎样表达更大更精确的小数？
  * Exp1. 采用8位阶码23位规格化尾数时，尾数表数范围是多少？
    * 隐藏位技术

储存的值	0000 0000 0000 0000 0000 000	1111 1111 1111 1111 1111 111
运算的值	0.1 00000000000000000000000	0.1 11111111111111111111111
表示的值	2^{-1}	$1-2^{-24}$

尾数部表示的范围为 $\pm 2^{-1}$~ $\pm(1-2^{-24})$

小数的表示

- Q2：怎样表达更大更精确的小数？
  - Exp1. 采用8位阶码23位规格化尾数时，表数范围是多少？

阶码的表示范围为：　[-128，127]

尾数部表示的范围为：[$\pm 2^{-1}$, $\pm(1-2^{-24})$]

此结构的规格化形式表示的范围为：

$$\pm 2^{-1} \times 2^{-128} \ \square\ \pm(1-2^{-24}) \times 2^{127}$$

$$\pm 2^{-1} \times 2^{-2^{x-1}} \sim \pm 1 - 2^{-(y+1)} \times 2^{2^{x-1}-1}$$

x，阶码位数；
y，尾数位数

Discussion:

Exp1中如果尾数为非规格化，那么表数范围是多少？

小数的表示

- **Q2：怎样表达更大更精确的小数？**
  - 浮点数的标准格式 IEEE754

单精度single precision
双精度double precision

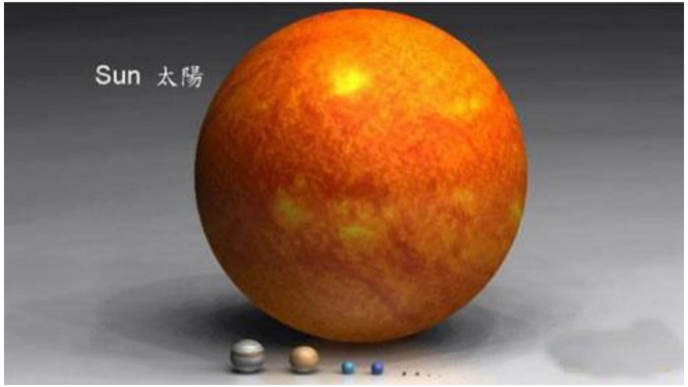
	1	8	23
32位	Ms	E	M

	1	11	52
64位	Ms	E	M

尾数M, 纯小数表示, 小数点放在尾数域的最前面。采用原码表示, 规格化，隐藏位。
阶码E，采用“移码”表示（移码可表示阶符）;
阶符，采用隐含方式，即采用移码方法来表示正负指数。

小数的表示

* Q2：怎样表达更大更精确的小数？
  * 浮点数的标准格式 IEEE754
    * 单精度浮点型的表数范围
      $-3.402823466\times10^{38}$~$3.402823466\times10^{38}$
    * 4字节定点数的表数范围
      -2.147483648×10^{9}~2.147483647×10^{9}



* 但是，其精度是有限的，仍然是实数的近似表达。

小数的表示

- Q3：更深入的问题-是否所有小数都能用二进制精确表示？

  $0.625_{(10)} = 0.101_{(2)}$

  $0.6251_{(10)} = ?$

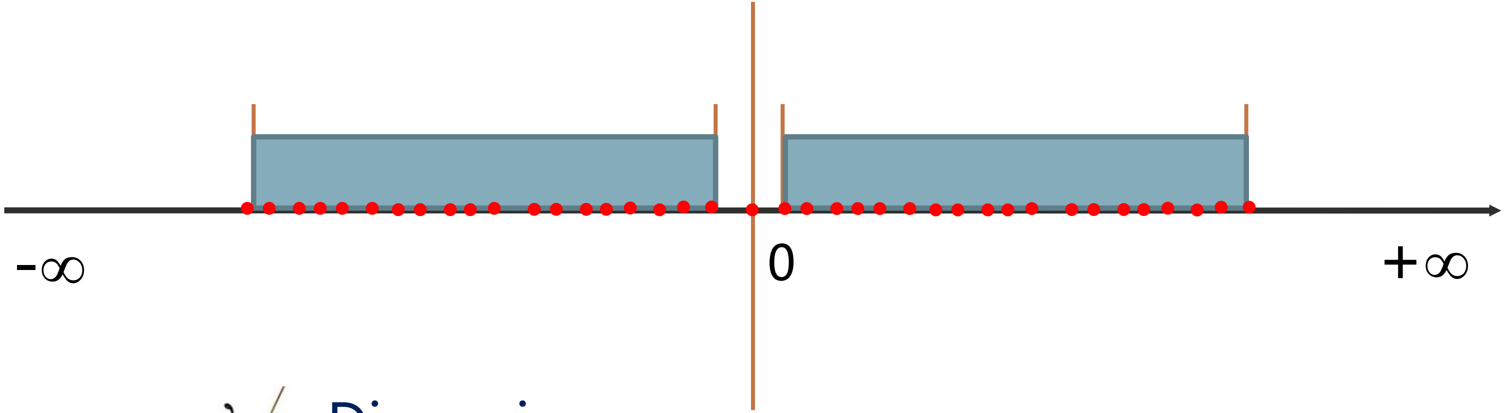
  0.101~0.110

  - 二进制小数与十进制小数之间并不是一一对应的关系
    - 一个二进制小数一定对应一个十进制小数
    - 而一个十进制小数却不一定刚好有一个二进制小数与之对应
  - 有效数字(**Significant Digit**)
    - 0.6251
    - 对于一个近似数，从左边第一个非0的数字起到精确到的位数为止，其间的所有数字。

单精度浮点型	6~7位
双精度浮点型	16位

小数的表示

- Q3：更深入的问题-是否所有数都能表示？



Discussion：

什么特征的数能用浮点数精确表示？