**第六章 数据的机器级表示**

1.**比特**(**bit**)：信息最小单位（每一个0或1），也叫**位**或**二进制位**。接近0的电压视为0，远离0的电压

视为1。

2.位组合：有k位，则区分2k个值。

3.按位运算

①**按位与运算 &**

**位屏蔽**（**掩码**）：屏蔽位0，凸显位1。

②**按位或运算 |** （同或运算，包含或运算）

③**按位非运算 ~** （补运算）

按位取反

④**按位异或运算 ^**

**相异为1，相同为0**

按位取反：取反位1，保持位0

判断相同：两个数运算结果每一位都等于0则相同。

⑤**左移 <<**

右侧空位用0补充

⑥算术**右移 >>**

左侧空位符号扩展

逻辑右移：左侧空位用0补充

4.**布尔代数**：二进制逻辑运算。

与A·B，或A+B，非，异或(·B)+(A·)

①恒等律：A+0=A，A·1=A

②0/1律：A+1=1，A·0=0

③互补律：A+=1，A·=0

④交换律：A+B=B+A，A·B=B·A

⑤结合律：A+(B+C)=(A+B)+C，A·(B·C)=(A·B)·C

⑥分配律：A·(B+C)=(A·B)+(A·C)，A+(B·C)=(A+B)·(A+C)

⑦德摩根律：·=

5.**数据类型**：计算机对信息的编码表示法。

6.**无符号整数**：有k位，表示0~2k-1

7.**有符号整数**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 原码 | 反码 | 补码 |
| 正数（首位0） | 不变 | 不变 | 不变 |
| 负数（首位1） | 首位变1 | 按位取反 | 按位取反后加1 / 右侧若干0和第一个1不变，其他位取反 |

8.**计算机采用二进制补码表示有符号整数**。有k位，能表示-2k-1~2k-1-1

1000…：绝对值最大的负数（最大正整数的相反数-1）

1111…：-1

0000…：0

|  |  |  |  |
| --- | --- | --- | --- |
| 20=1 | 24=16 | 28=256 | 2-1=0.5 |
| 21=2 | 25=32 | 29=512 | 2-2=0.25 |
| 22=4 | 26=64 | 210=1024 | 2-3=0.125 |
| 23=8 | 27=128 | 211=2048 | 2-4=0.0625 |

9.**ALU**：**算数逻辑单元**

10.保证ALU正确做加法的运算原则

①绝对值相同符号相反的两个整数相加得0。

②每个数加1得到正确结果。（**首位的进位总是被忽略**）

11.二进制转化为十进制：按各位权重相加。

12.十进制转化为二进制

整数部分：除2取余，由下到上。

小数部分：乘2取整，依次排列。

例：45.8125d=0101101.1101b=1.011011101X25b

13.算数进位

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 源A+源B+ carry in | 0 | 1 | 2 | 3 |
| 结果位 | 0 | 1 | 0 | 1 |
| carry out | 0 | 0 | 1 | 1 |

14.**符号扩展**（**SEXT**）

正数以0向左扩展

负数以1向左扩展。

**不同长度补码整数作加法运算需先符号扩展。**

15.**溢出**：加法结果超出表示范围。

表现形式：两个正数相加得负数，两个负数相加得正数。

**一个正数与一个负数相加永不溢出。**

16.**ASCII码**：**美国标准信息交换码**，所有计算机处理单元和输入输出设备转换字符的标准码。

键盘上每个键被唯一个ASCII码识别，8个二进制位表示，首位为0。

0-32号、127号：通讯专用字符或控制字符。

33-126号：可见字符。

17.**定点小数**小数点对齐作加法。

18.**浮点数**用来表示精度少、位数多的小数。

19.**IEEE**：**国际电气和电子工程师协会**

20.浮点数算数运算标准：IEEE-754浮点数。

21.**单精度浮点数**

范围：10±38

精度：7位

符号s（1位，0正1负）+指数域exponent（8位无符号整数）+分数域fraction（23位）

22.**双精度浮点数**

范围：10±306

精度：15位

符号s（1位，0正1负）+指数域exponent（11位无符号整数）+分数域fraction（52位）

23.浮点数运算步骤

①使指数相等

②分数运算

③结果规格化

④丢失位舍入

⑤判断指数溢出

24.**十六进制**和**八进制**方便人们手工处理二进制位。

25.一个十六进制位对应四个二进制位

|  |  |  |  |
| --- | --- | --- | --- |
| 0000=0 | 0100=4 | 1000=8 | 1100=C |
| 0001=1 | 0101=5 | 1001=9 | 1101=D |
| 0010=2 | 0110=6 | 1010=A | 1110=E |
| 0011=3 | 0111=7 | 1011=B | 1111=F |

26.十六进制字面常量书写时使用前缀0x或x，在C语言中十六进制字面常量使用前缀0X或0x。

27.一个**八进制**位对应三个二进制位

|  |  |  |  |
| --- | --- | --- | --- |
| 000=0 | 010=2 | 100=4 | 110=6 |
| 001=1 | 011=3 | 101=5 | 111=7 |

28.字符串：字符序列，以NUL（0）结束

29.图像：像素矩阵

30.声音：定点数序列

31.C语言中的数据类型采用位数与计算机指令集结构和编译器有关。

int：二进制补码整型

char：ASCII码（进行运算时与对应ASCII码等值）

double：双精度浮点数

32.格式说明

%d 十进制补码整数

%x 十六进制

%o 八进制

%c 字符

%f 单精度浮点数

%lf 双精度浮点数