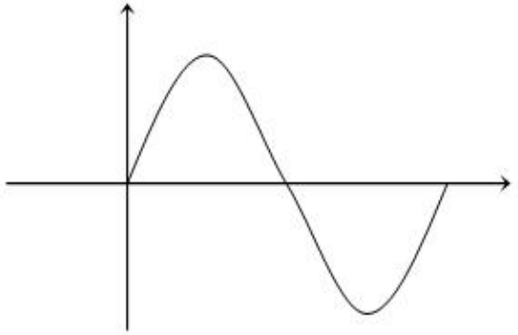
数字音频培训

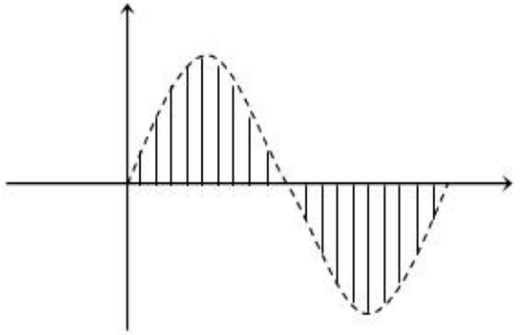
模拟信号：
指在时间和幅度上都连续的信号。
数字信号：
时间和幅度上都不连续的信号。

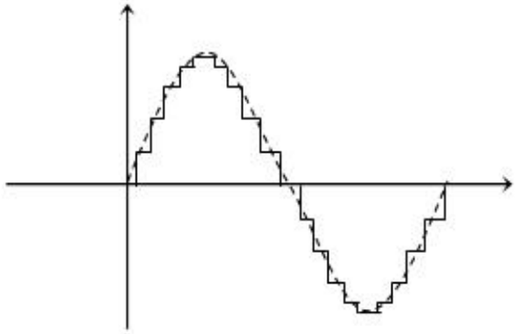
数字信号的形成
将模拟信号转变成数字信号，
需要经过取样、量化和编码
三个步骤。

连续的模拟声音信号



声音信号的采样





离散的音频信号

DIRECTORY

目录

01 取样

1、取样

将模拟信号的幅度值以一定的时间间隔取得样值。

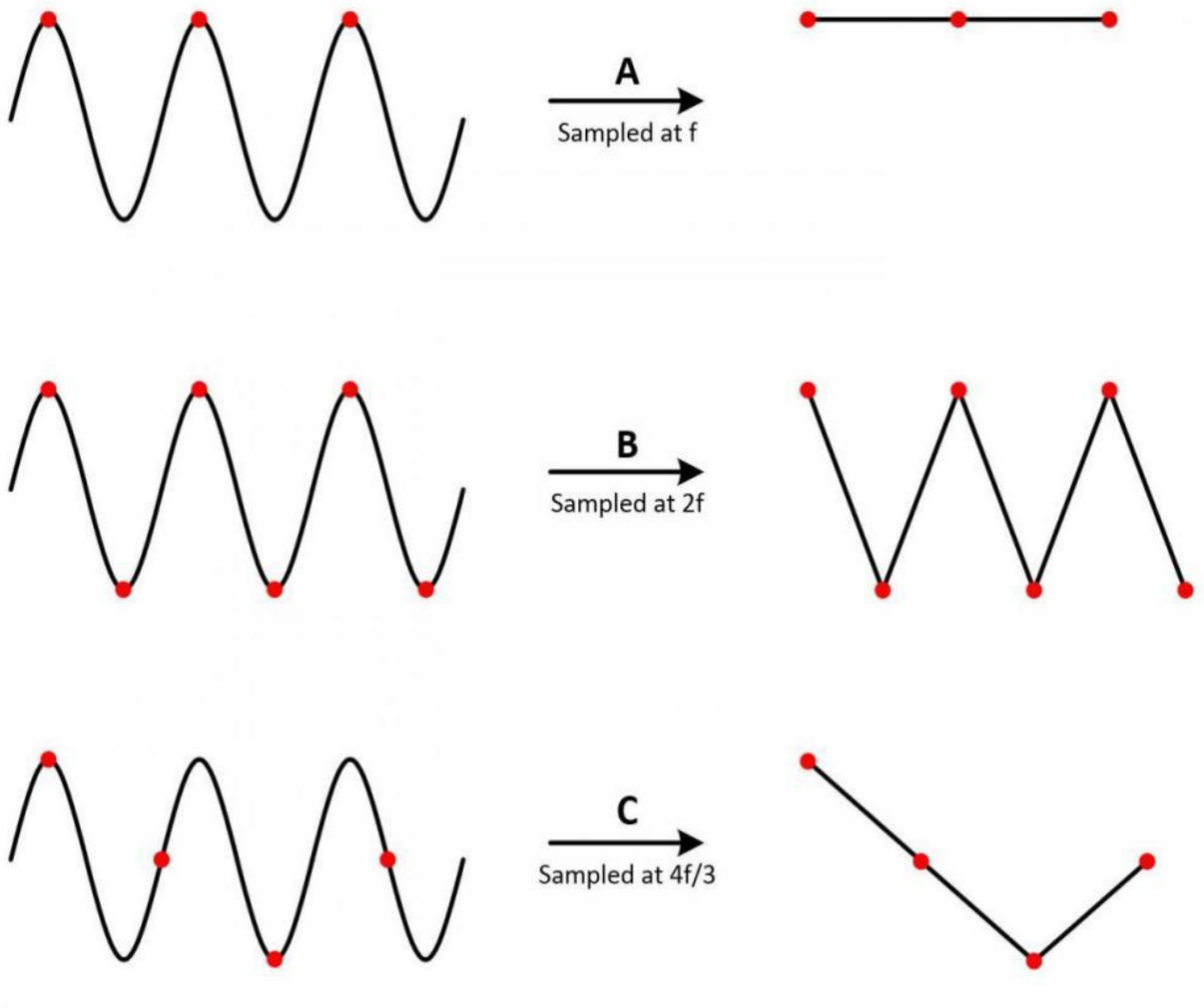
取样时间间隔称为取样周期。

每秒钟取样次数称为取样频率。

奈奎斯特取样定理

在进行模拟/数字信号的转换过程中，当采样频率fs.max大于信号中最高频率fmax的2倍时(fs.max>=2fmax)，采样之后的数字信号完整地保留了原始信号中的信息，一般实际应用中保证采样频率为信号最高频率的5~10倍;采样定理又称奈奎斯特定理。又称香农采样定理。

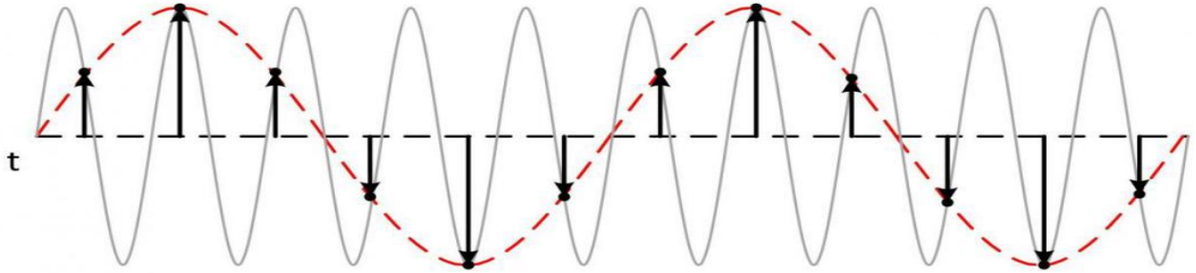
为更好理解其原因，让我们来看看不同速率测量的正弦波。 情况A，频率f的正弦波以同一频率采样。 这些采样标记在原始信号的左侧，在右侧构建时，信号错误地显示为恒定直流电压。 情况B，采样率是信号频率的两倍。 现在信号显示为三角波。 这种情况下，f等于奈奎斯特频率，这也是特定采样频率下为了避免混叠而允许的最高频率分量。 情况C，采样率是4f/3。



奈奎斯特采样定理是1928年由美国电信工程师H.奈奎斯特首先提出来的，因此称为奈奎斯特采样定理。1933年由苏联工程师科捷利尼科夫首次用公式严格地表述这一定理，因此在苏联文献中称为科捷利尼科夫采样定理。1948年信息论的创始人C.E.香农对这一定理加以明确地说明并正式作为定理引用，因此在许多文献中又称为香农采样定理。

混叠

如需按一定速率采样以避免混叠，那么混叠到底是什么？ 如果信号的采样率低于两倍奈奎斯特频率，采样数据中就会出现虚假的低频成分。这种现象便称为混叠。



AES（音频工程师协会）推荐的三种取样频率：48KHz、44.1KHz、32KHz。

取样保持：在A/D转换器之前，为使取样值保持一定时间而设定的。

02 量化

量化是将模拟信号的取样值，经过四舍五入转换成一种数字信号的过程。

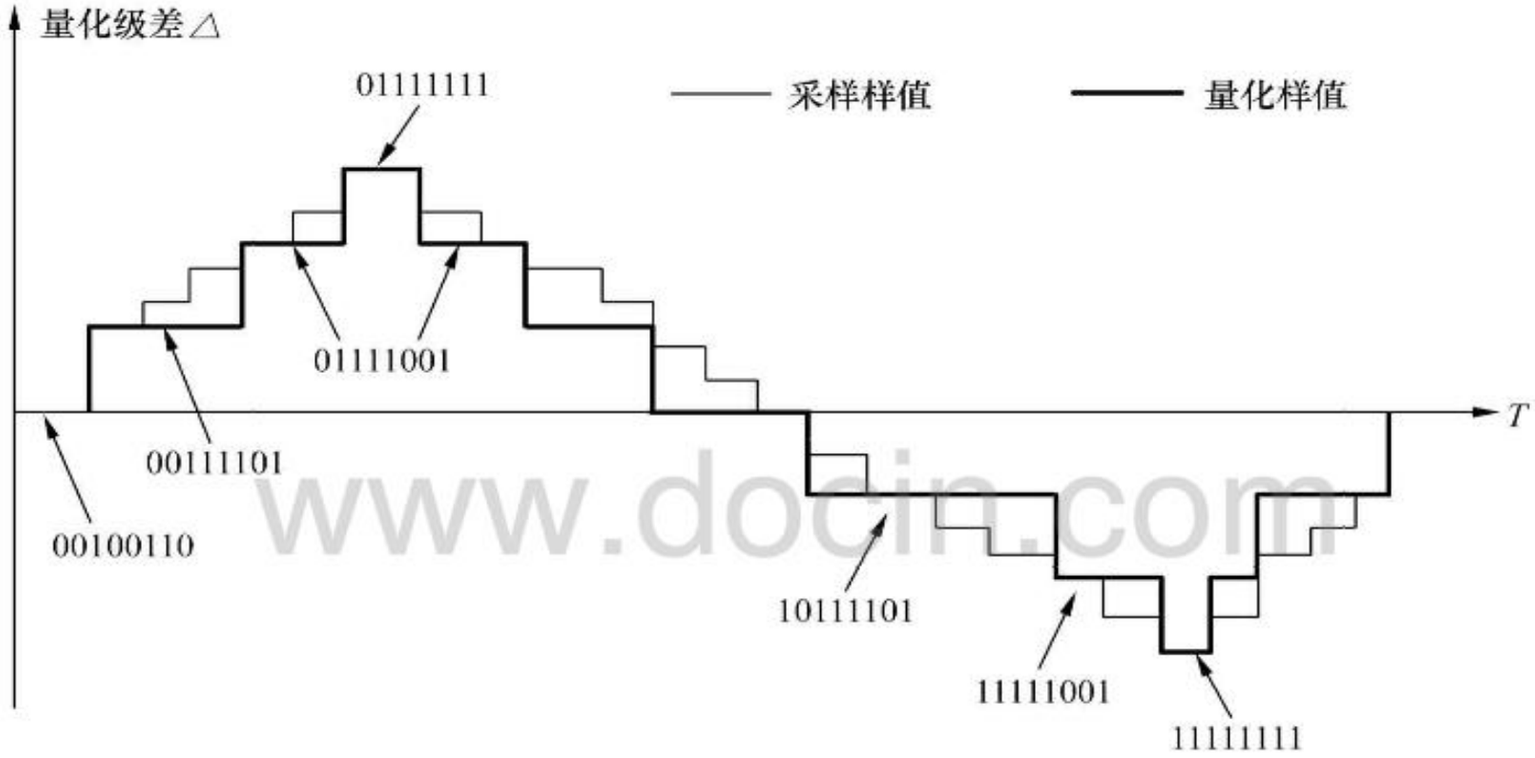
量化比特数：采用二进制的有效位数。常见的是8bit和16bit。

量 化 噪 声：在实际量化过程中，要对每个取样信号进行四舍五入，每个取样点都会产生舍入误差，并存在与这种舍入误差相应的失真和噪声。也称为量化失真。

数字系统的动态范围=6×位数+1.8dB

量化阶梯数也称为量化级数，指量化时所能取的值数。对二进制来说，位数越多，则可表示的二进制数也越多。

例如用8位(1字节)二进制表示十进制整数，只能表示出-128~127之间的整数值，也就是256个量化级。如果用16位二进制数，则具有64K(65536)个量化级。



03 编码

3.1 编码分类

编码是将数字化后的样值，通过一定的二进制模式表现的过程（个人理解）。

PCM（Pulse Code Modulation，脉冲编码调制）：将模拟信号转换成数码，然后再转换成二进制数字信号的方法。

编码分类

根据编码方式的不同，音频编码技术分为三种:波形编码、参数编码和混合编码。一般来说，波形编码的话音质量高，但编码速率也很高;参数编码的编码速率很低，产生的合成语音的音质不高;混合编码使用参数编码技术和波形编码技术，编码速率和音质介于它们之间。

3.1.1 波形编码

波形编码是指不利用生成音频信号的任何参数，直接将时间域信号变换为数字代码，使重构的语音波形尽可能地与原始语音信号的波形形状保持一致。波形编码的基本原理是在时间轴上对模拟语音信号按一定的速率抽样，然后将幅度样本分层量化，并用代码表示。

波形编码方法简单、易于实现、适应能力强并且语音质量好。不过因为压缩方法简单也带来了一些问题:压缩比相对较低，需要较高的编码速率。一般来说，波形编码的复杂程度比较低，编码速率较高、通常在16 kbit/s以上，质量相当高。但编码速率低于16 kbit/s时，音质会急剧下降。

最简单的波形编码方法是PCM它只对语音信号进行采样和量化处理。优点是编码方法简单，延迟时间短，音质高，重构的语音信号与原始语音信号几乎没有差别。不足之处是编码速率比较高(64 kbit/s)，对传输通道的错误比较敏感。

3.1.2 参数编码

参数编码是从语音波形信号中提取生成语音的参数，使用这些参数通过语音生成模型重构出语音，使重构的语音信号尽可能地保持原始语音信号的语意。也就是说，参数编码是把语音信号产生的数字模型作为基础，然后求出数字模型的模型参数，再按照这些参数还原数字模型，进而合成语音。

参数编码的编码速率较低，可以达到2.4 kbit/s，产生的语音信号是通过建立的数字模型还原出来的，因此重构的语音信号波形与原始语音信号的波形可能会存在较大的区别、失真会比较大。而且因为受到语音生成模型的限制，增加数据速率也无法提高合成语音的质量。不过，虽然参数编码的音质比较低，但是保密性很好，一直被应用在军事上。典型的参数编码方法为LPC(Linear Predictive Coding，线性预测编码)。

3.1.3 混合编码

混合编码是指同时使用两种或两种以上的编码方法进行编码。这种编码方法克服了波形编码和参数编码的弱点，并结合了波形编码高质量和参数编码的低编码速率，能够取得比较好的效果。

3.1.4 编码对比

编码技术	算法	编码标准	码率(kbit/s)	质量	应用领域
波形编码	PCM	G.711	64	4.3	PSTN、ISDN
	ADPCM	G.721	32	4.1	-
	SB-ADPCM	G.722	64/56/48	4.5	-
参数编码	LPC	-	2.4	2.5	保密语音
混合编码	CELPC	-	4.8	3.2	-
	VSELPC	GIA	8	3.8	移动通信、语音信箱
	RPE-LTP	GSM	13.2	3.8	-
	LD-CELP	G.728	16	4.1	ISDN
	MPE	MPE	128	5.0	CD

说明:质量评价共五个等级(1、2、3、4、5)，其中5.0为最高分。
上表中各种算法、应用领域中缩略语的中文和英文全称参见下面说明。

3.1.4 编码对比

PCM:Pulse Code Modulation，脉冲编码调制。

ADPCM:Adaptive Differential Pulse Code Modulation，自适应差分脉冲编码调制。

SB-ADPCM:Subband Adaptive Differential Pulse Code Modulation，子带-自适应差分脉冲编码调制。

LPC:Linear Predictive Coding，线性预测编码。

CELPC:Code Excited Linear Predictive Coding，码激励线性预测编码。

VSELPC:Vector Sum Excited Linear Predictive Coding，矢量和激励线性预测编码。

RPE-LTP:Regular Pulse Excited-Long Term Predictive，规则脉冲激励长时预测。

LD-CELP:Low Delay-Code Excited Linear Predictive，低时延码激励线性预测。

MPE:Multi-Pulse Excited，多脉冲激励。

PSTN:Public Switched Telephone Network，公共交换电话网。

ISDN:Integrated Services Digital Network，综合业务数字网。

3.1.4 编码各种二进制码

表10-1 各种二进制码

十进制码	二进制码				
量化电平	自然二进制码	偏移二进制码	补码	偏移反射二进制码	折叠二进制码
+7	1 1 1	1 1 1 1	0 1 1 1 +7	1 0 0 0	0 1 1 1
+6	1 1 0	1 1 1 0	0 1 1 0 +6	1 0 0 1	0 1 1 0
+5	1 0 1	1 1 0 1	0 1 0 1 +5	1 0 1 1	0 1 0 1
+4	1 0 0	1 1 0 0	0 1 0 0 +4	1 0 1 0	0 1 0 0
+3	0 1 1	1 0 1 1	0 0 1 1 +3	1 1 1 0	0 0 1 1
+2	0 1 0	1 0 1 0	0 0 1 0 +2	1 1 1 1	0 0 1 0
+1	0 0 1	1 0 0 1	0 0 0 1 +1	1 1 0 1	0 0 0 1
+0	0 0 0	1 0 0 0	0 0 0 0 0	1 1 0 0	0 0 0 0
-0		—	—	—	1 0 0 0
-1		0 1 1 1	1 1 1 1 -1	0 1 0 0	1 0 0 1
-2		0 1 1 0	1 1 1 0 -2	0 1 0 1	1 0 1 0
-3		0 1 0 1	1 1 0 1 -3	0 1 1 1	1 0 1 1
-4		0 1 0 0	1 1 0 0 -4	0 1 1 0	1 1 0 0
-5		0 0 1 1	1 0 1 1 -5	0 0 1 0	1 1 0 1
-6		0 0 1 0	1 0 1 0 -6	0 0 1 1	1 1 1 0
-7		0 0 0 1	1 0 0 1 -7	0 0 0 1	1 1 1 1
-8		0 0 0 0	1 0 0 0 -8	0 0 0 0	

3.2 原码, 反码, 补码

在学习原码, 反码和补码之前, 需要先了解机器数和真值的概念

3.2.1 机器数

一个数在计算机中的二进制表示形式，叫做这个数的机器数。机器数是带符号的，在计算机用一个数的最高位存放符号, 正数为0, 负数为1.

比如，十进制中的数 +3 ，计算机字长为8位，转换成二进制就是00000011。如果是 -3 ，就是 10000011 。

那么，这里的 00000011 和 10000011 就是机器数。

3.2.2 真值

因为第一位是符号位，所以机器数的形式值就不等于真正的数值。例如上面的有符号数 10000011，其最高位1代表负，其真正数值是 -3 而不是形式值131（10000011转换成十进制等于131）。所以，为区别起见，将带符号位的机器数对应的真正数值称为机器数的真值。 以下为例：

0000 0001的真值 = +000 0001 = +1，1000 0001的真值 = –000 0001 = –1

3.2.3 原码, 反码, 补码的基础概念和计算方法

在探求为何机器要使用补码之前，让我们先了解原码，反码和补码的概念。

对于一个数，计算机要使用一定的编码方式进行存储。

原码，反码，补码是机器存储一个具体数字的编码方式。

3.2.4 原码

原码就是符号位加上真值的绝对值，即用第一位表示符号，其余位表示值。

比如如果是8位二进制:　[+1]原 = 0000 0001

[-1]原 = 1000 0001

第一位是符号位。因为第一位是符号位，所以8位二进制数的取值范围就是:

[1111 1111 , 0111 1111]

即

[-127 , 127]

原码是人脑最容易理解和计算的表示方式。

3.2.5 反码

反码的表示方法是：

正数的反码是其本身。

负数的反码是在其原码的基础上，符号位不变，其余各个位取反。

[+1] = [00000001]原 = [00000001]反

[-1] = [10000001]原 = [11111110]反

可见如果一个反码表示的是负数，人脑无法直观的看出来它的数值。

通常要将其转换成原码再计算。

3.2.5 补码

补码的表示方法是：

正数的补码就是其本身。

负数的补码是在其原码的基础上，符号位不变，其余各位取反，最后+1。

(即在反码的基础上+1)

[+1] = [00000001]原 = [00000001]反 = [00000001]补

[-1] = [10000001]原 = [11111110]反 = [11111111]补

对于负数，补码表示方式也是人脑无法直观看出其数值的。

通常也需要转换成原码在计算其数值。

3.3 为何要使用原码, 反码和补码

在开始深入学习前, 我的学习建议是先"死记硬背"上面的原码, 反码和补码的表示方式以及计算方法.

现在我们知道了计算机可以有三种编码方式表示一个数.

对于正数因为三种编码方式的结果都相同:

[+1] = [00000001]原 = [00000001]反 = [00000001]补

所以不需要过多解释. 但是对于负数:

[-1] = [10000001]原 = [11111110]反 = [11111111]补

可见原码, 反码和补码是完全不同的. 既然原码才是被人脑直接识别并用于计算表示方式, 为何还会有反码和补码呢?

3.3 为何要使用原码, 反码和补码

首先, 因为人脑可以知道第一位是符号位, 在计算的时候我们会根据符号位, 选择对真值区域的加减. (真值的概念在本文最开头). 但是对于计算机, 加减乘数已经是最基础的运算, 要设计的尽量简单. 计算机辨别"符号位"显然会让计算机的基础电路设计变得十分复杂! 于是人们想出了将符号位也参与运算的方法. 我们知道, 根据运算法则减去一个正数等于加上一个负数, 即: 1-1 = 1 + (-1) = 0 , 所以机器可以只有加法而没有减法, 这样计算机运算的设计就更简单了.

于是人们开始探索 将符号位参与运算, 并且只保留加法的方法. 首先来看原码:

计算十进制的表达式: 1-1=0

1 - 1 = 1 + (-1) = [00000001]原 + [10000001]原 = [10000010]原 = -2

如果用原码表示, 让符号位也参与计算, 显然对于减法来说, 结果是不正确的.这也就是为何计算机内部不使用原码表示一个数.

3.3 为何要使用原码, 反码和补码

为了解决原码做减法的问题, 出现了反码:

计算十进制的表达式: 1-1=0

1 - 1 = 1 + (-1) = [0000 0001]原 + [1000 0001]原= [0000 0001]反 + [1111 1110]反 = [1111 1111]反 = [1000 0000]原 = -0

发现用反码计算减法, 结果的真值部分是正确的. 而唯一的问题其实就出现在"0"这个特殊的数值上. 虽然人们理解上+0和-0是一样的, 但是0带符号是没有任何意义的. 而且会有[0000 0000]原和[1000 0000]原两个编码表示0.

于是补码的出现, 解决了0的符号以及两个编码的问题:

1-1 = 1 + (-1) = [0000 0001]原 + [1000 0001]原 = [0000 0001]补 + [1111 1111]补 = [0000 0000]补=[0000 0000]原

这样0用[0000 0000]表示, 而以前出现问题的-0则不存在了.

3.4 调制

PCM就是一种调制方式，它是将模拟信号经过取样、量化、编码后，以1和0所组成的二进制码来表示的。

1和0用什么样的波形来表达的方式成为调制方式。

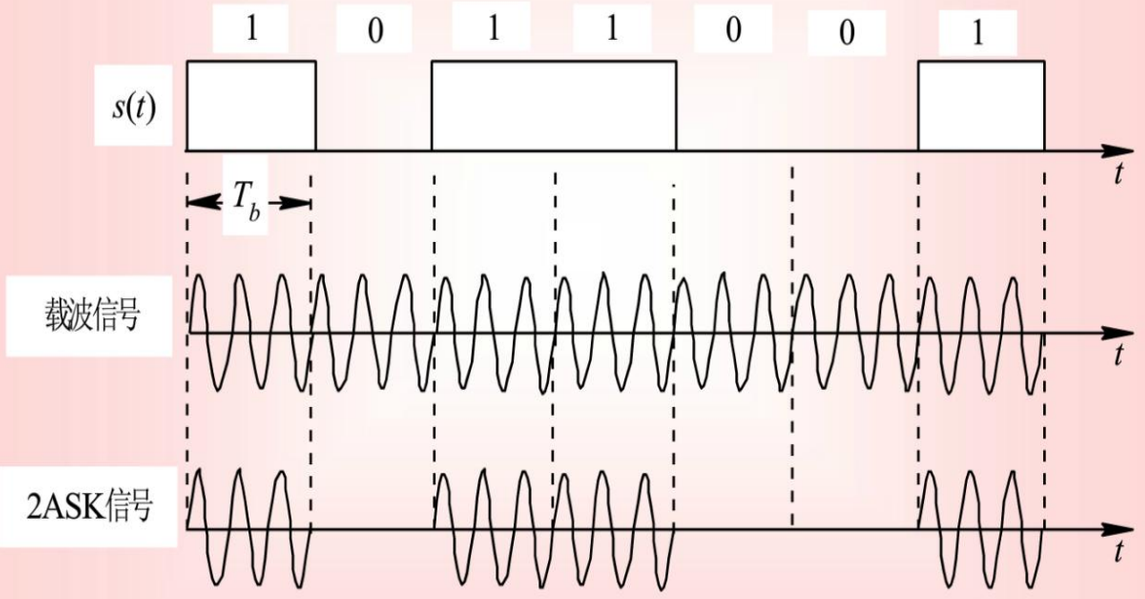
主要有三种形式：移幅键控ASK、移频键控法FSK、移相键控法PSK

3.4.1 调制

移幅键控（ASK）：

即按载波的幅度受到数字数据的调制而取不同的值，例如对应二进制0，载波振幅为0；对应二进制1，载波振幅为1。调幅技术实现起来简单，但容易受增益变化的影响，是一种低效的调制技术。在电话线路，通常只能达到1200bps的速率。

2ASK信号的时间波形随二进制基带信号s(t)通断变化，所以又称为通断键控信号（OOK信号）。

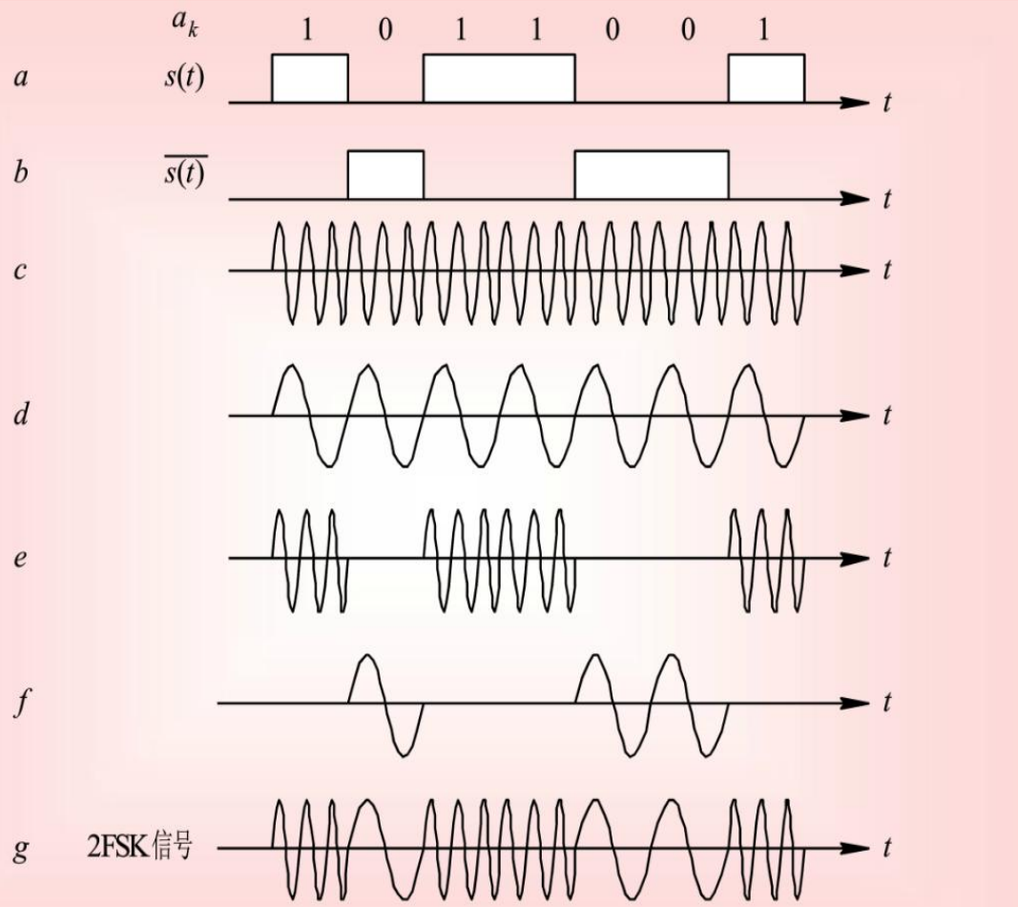


二进制振幅键控信号时间波型

3.4.2 调制

频移键控（FSK）：

即按数字数据的值（0或1）调制载波的频率。例如对应二进制0的载波频率为F1，而对应二进制1的载波频率为F2。该技术抗干扰性能好，但占用带宽较大。在电话线路，使用FSK可以实现全双工操作，通常可达到1200bps的速率。

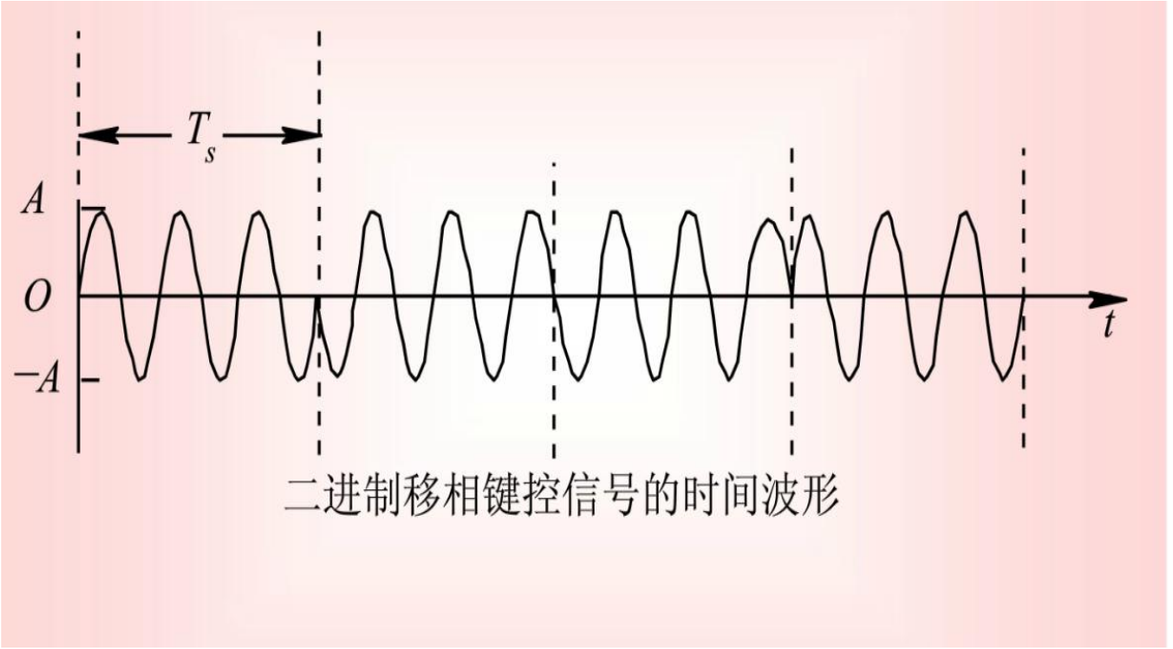


二进制移频键控信号的时间波形

3.4.3 调制

相移键控（PSK）：

即按数字数据的值调制载波相位。例如用180相移表示1，用0相移表示0。这种调制技术抗干扰性能最好，且相位的变化也可以作为定时信息来同步发送机和接收机的时钟，并对传输速率起到加倍的作用。

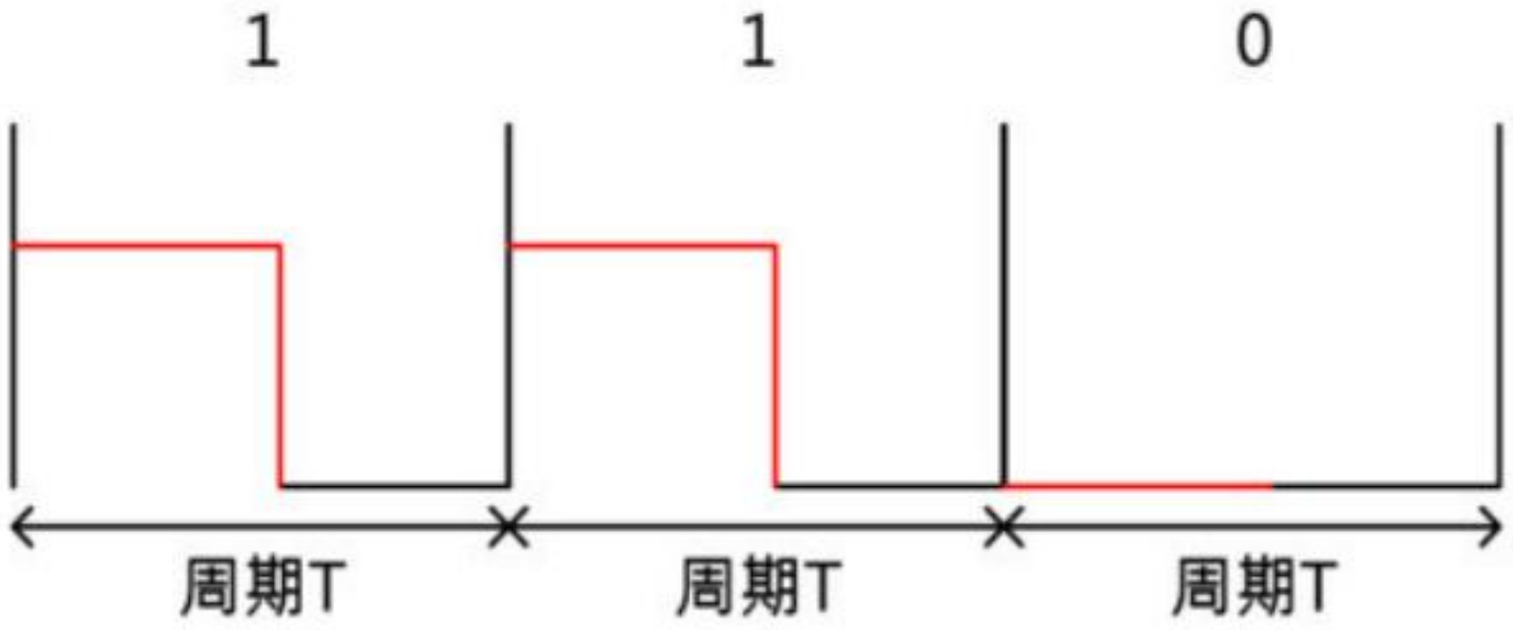


二进制移相键控信号的时间波形

3.5 RZ、NRZ、NRZI、曼彻斯特编码

3.5.1 RZ(Return Zero Code)编码

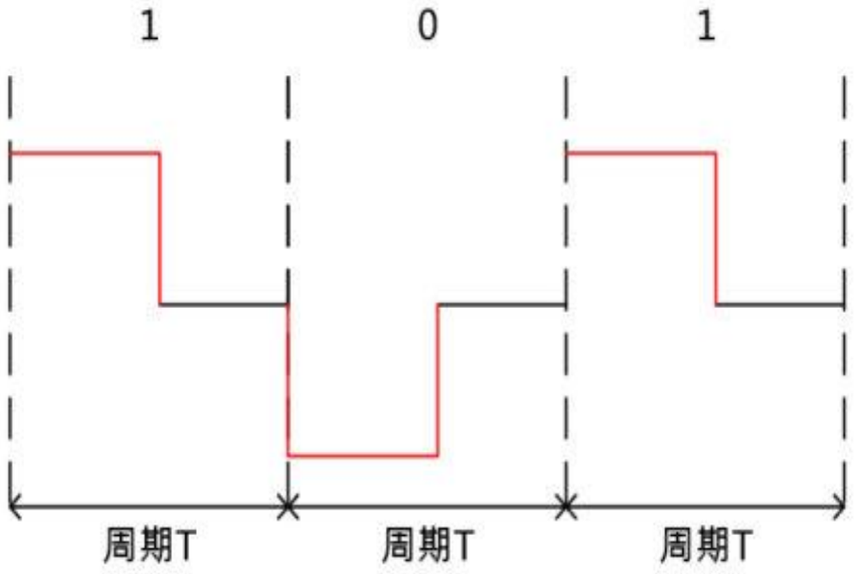
RZ编码也成为归零码，归零码的特性就是在一个周期内，用二进制传输数据位，在数据位脉冲结束后，需要维持一段时间的低电平。举个图例吧：



RZ码示意图

3.5.1 RZ(Return Zero Code)编码

图中红色的线表示数据，只占据一部分的周期，剩下周期部分为归零段。而归零码而分为单极性归零码和双极性归零码，图1表示的是单极性归零码，即低电平表示0，正电平表示1。对于双极性归零码来说，则是高电平表示1，负电平表示0。如下图所示：

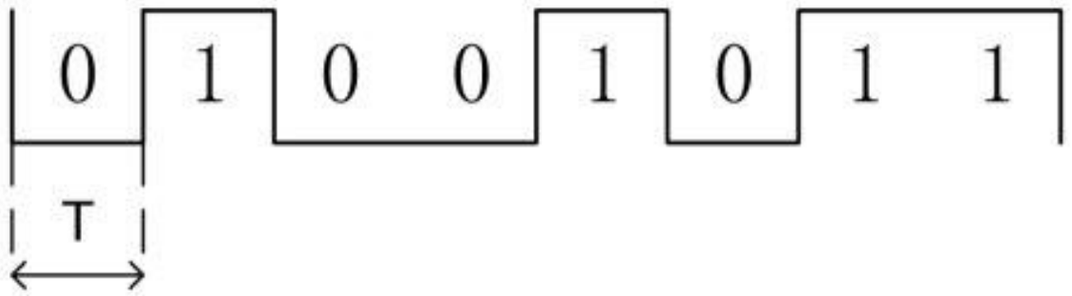


双极性RZ码示意图

这种编码方式虽说能够同时传递时钟信号和数据信号，但由于归零需要占用一部分的带宽，故传输效率也就收到了一定的限制，假设数据传输时间为t，一个周期时间为T，则这种传输效率$\eta=t/T$。

3.5.2 NRZ(Non Return Zero Code)编码

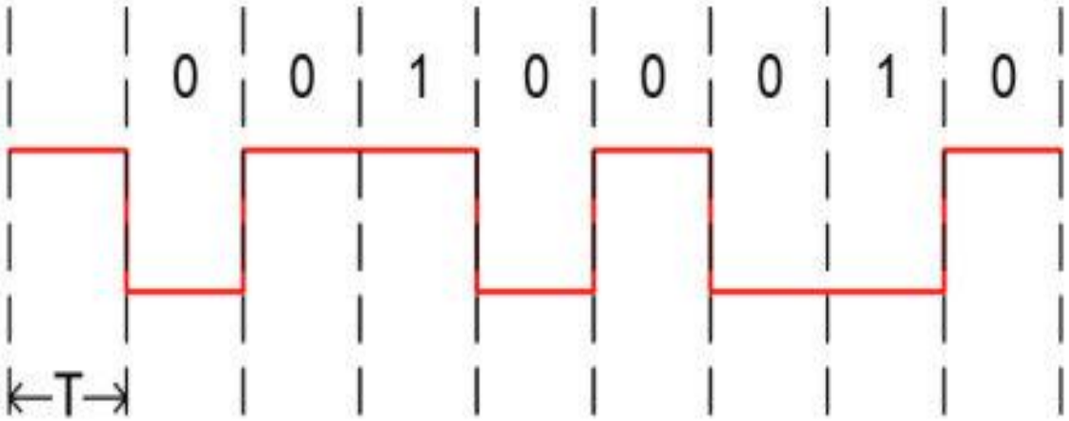
NRZ编码也成为不归零编码，也是我们最常见的一种编码，即正电平表示1，低电平表示0。它与RZ码的区别就是它不用归零，也就是说，一个周期可以全部用来传输数据，这样传输的带宽就可以完全利用。一般常见的带有时钟线的传输协议都是使用NRZ编码或者差分的NRZ编码。因此，使用NRZ编码若想传输高速同步数据，基本上都要带有时钟线，因为本身NRZ编码无法传递时钟信号。但在低速异步传输下可以不存在时钟线，但在通信前，双方设备要约定好通信波特率，例如UART。



NRZ编码示意图

3.5.3 NRZI(Non Return Zero Inverted Code)

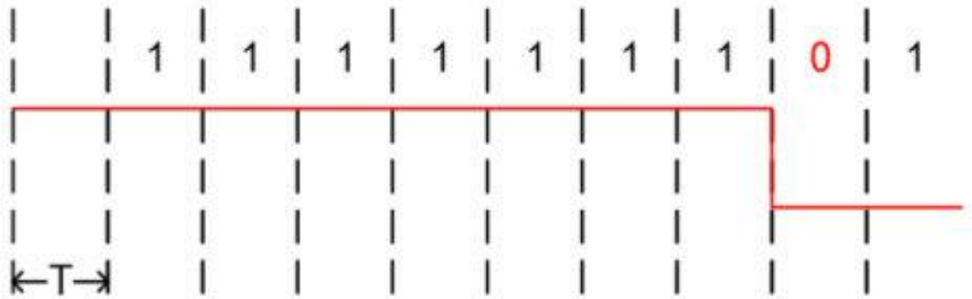
NRZI编码的全称为反向不归零编码，也叫不归零倒相，这种编码方式集成了前两种编码的优点，即既能传输时钟信号，又能尽量不损失系统带宽。对于USB2.0、AES10(MADI)通信的编码方式就是NRZI编码。其实NRZI编码方式非常的简单，即信号电平翻转表示0，信号电平不变表示1。例如想要表示00100010(B)，则信号波形如下图所示：



NRZI编码示意图

3.5.3 NRZI(Non Return Zero Inverted Code)

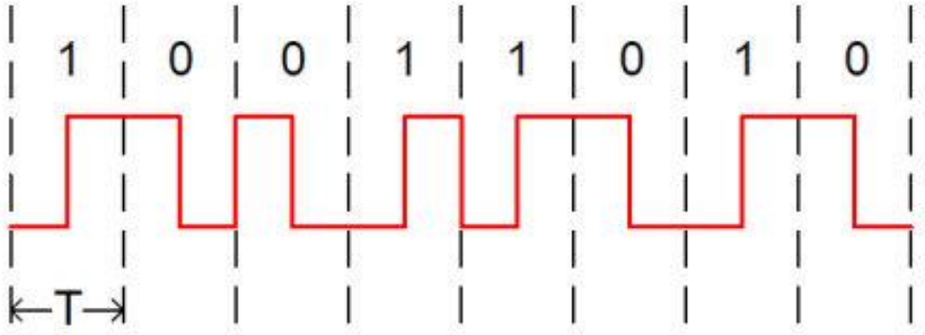
由图可以看到，当电平状态发生变化时，表示的数据为0。在传输的数据中，很少出现全1的状态，故接收端可以根据发送端的电平变化确定采样时钟频率。但是有时候依然会出现数据为全1的状态，也就是说信号线一直保持一个状态，这个时候时钟信号就无法传输，接收端就无法同步时钟信号，这该如何解决呢？解决方式就是在一定数量的1之后强行插入一个0，就是说若信号线状态一直持续一段时间不变的话，发送端强行改变信号线的状态，接收端则只需要将这个变化忽略掉就可以了。在USB2.0的协议中规定为传输7个1则在数据中插入一个0。例如有一段数据为：1111 1111 (B)要发送，则整个传输线上的电平状态是这样的：



3.5.4 曼彻斯特编码

曼彻斯特编码(Manchester Encoding)，也叫做相位编码(Phase Encode，简写PE)，是一个同步时钟编码技术，被物理层使用来编码一个同步位流的时钟和数据。它在以太网媒介系统中的应用属于数据通信中的两种位同步方法里的自同步法(另一种是外同步法)，即接收方利用包含有同步信号的特殊编码从信号自身提取同步信号来锁定自己的时钟脉冲频率，达到同步目的。

曼彻斯特编码方式和NRZI编码十分相似，只不过它是利用信号的跳变方向来决定数据的。在位中间，信号由高向低跳变表示数据0，信号由低向高跳变表示数据1。举个图例吧，若要表示数据1001 1010(B)，则信号波形图如下图所示：



曼彻斯特编码示意图

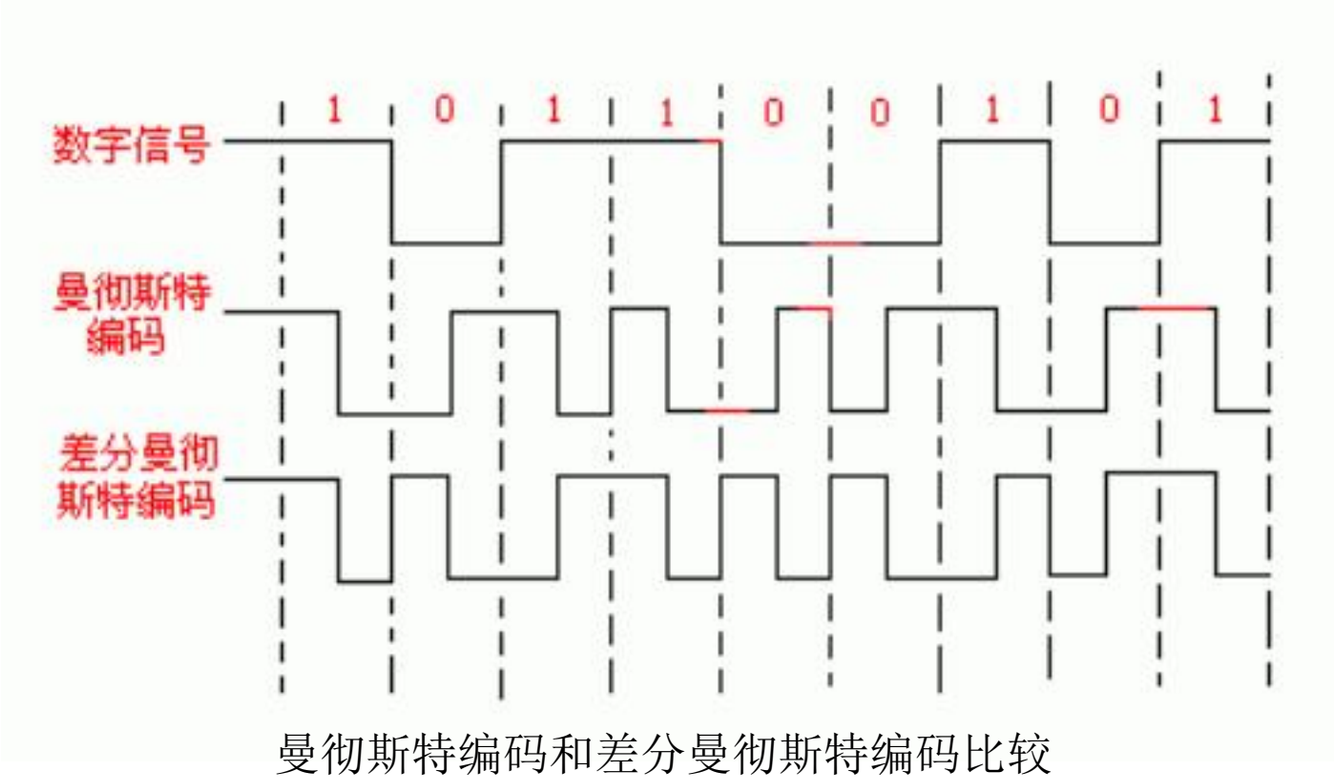
3.5.4 曼彻斯特编码

曼彻斯特编码方式也如前面所说，虽然传输了时钟信号，但也损失了一部分的带宽，主要表现在相邻相同数据上。但对于高速数据来说，这种编码方式无疑是这几种编码方式中最优的，相比NRZI编码，曼彻斯特编码不存在长时间信号状态不变导致的时钟信号丢失的情况，所以所有802.3系统都采用它进行编码。其高信号电平为+0.85V，低信号电平为-0.85V，这样直流电压为0V。

曼彻斯特编码，常用于局域网传输。曼彻斯特编码将时钟和数据包含在数据流中，在传输代码信息的同时，也将时钟同步信号一起传输到对方，每位编码中有一跳变，不存在直流分量，因此具有自同步能力和良好的抗干扰性能。但每一个码元都被调成两个电平，所以数据传输速率只有调制速率的1/2。

3.5.5 差分曼彻斯特编码

每位中间的跳变仅提供时钟定时，而用每位开始时有无跳变表示"0"或"1"，有跳变为"0"，无跳变为"1"。两种情况下，都能保证在区间的中部有电压跳变，这种方案需要的设备更复杂，但能提供更好地噪声抑制性能。



曼彻斯特编码和差分曼彻斯特编码比较

3.5.5 差分曼彻斯特编码

AES3采用的编码方式叫双相标志编码，其实就是利用了曼彻斯特编码的原理，我们常说AES信号可以自同步，也是这个道理。

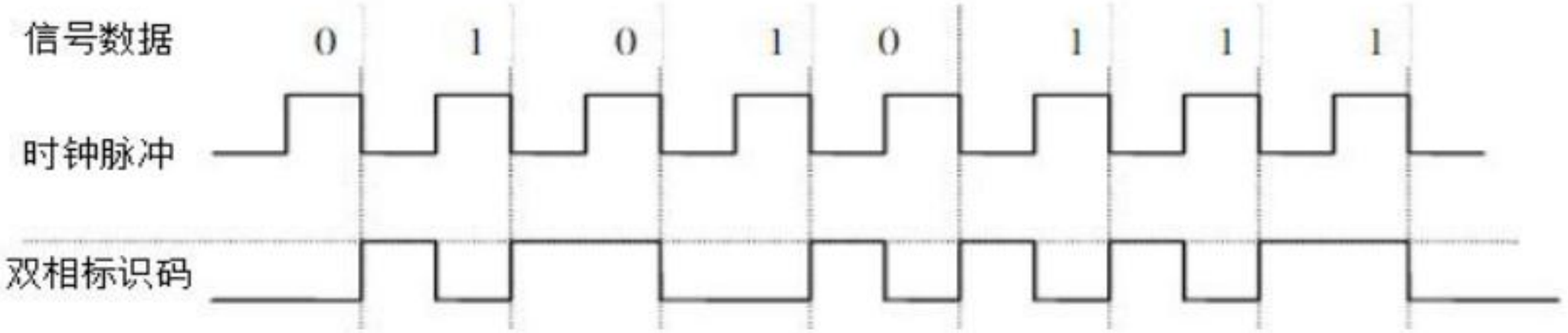


图 1 双相标识码

其编码规则是无论码元为“1”或“0”，在每个数据比特周期的开始都有一个电平转换，而且每个码元“1”的中间有一个跳变，为归零码，对于码元“0”，在整个比特周期之内保持电平不变。双相标志码编码的数据流中不会出现两个连续的“1”或“0”。

3.5.6 4B/5B编码

在讲4B/5B编码前，我们先了解一下帧同步信号。

对于数据链路层，通常要把比特流分成离散的帧，并对每一帧计算出校验和。帧的传输是网络传输的基本单位，但是如何把比特流分成帧却不容易，通常的办法是在帧之间插入时间间隔，就像在英文单词之间加入空格，以区分每一个单词。但是由于网络很难保证计时准确，所以无法依赖计时来确定每一帧，因此人们就提出了加入特殊编码的同步位的办法。

简单来说，就是用一些特殊的二进制组合来确定帧头帧尾，这些特殊组合不能在正常的数据中出现，这就是我们常说的同步信号。4B/5B编码就能解决这一问题。

4B/5B编码就是每4位二进制代码由5位编码表示，这5位编码称为编码组(code group)，并且由NRZI方式传输。

3.5.6 4B/5B编码

4B:5B编码方案是把数据转换成5位符号,供传输。这些符号保持线路的交流(AC)平衡;在传输中,其波形的频谱为最小。信号的直流(DC)分量变化小于额定中心点的10%。

这种编码的特点是将欲发送的数据流每4bit作为一个组，然后按照4B/5B编码规则将其转换成相应5bit码。5bit码共有32种组合，但只采用其中的16种对应4bit码的16种，其他的16种或者未用或者用作控制码，以表示帧的开始和结束、光纤线路的状态(静止、空闲、暂停)等。

几种应用实例FDDI、100BASE-TX和100BASE-FX，数字音频领域的AES10(MADI)

8B/10B编码与4B/5B概念类似，例如在千兆以太网中就采用了8B/10B的编码方式。

编码效率:为用5位数字表示4位数字，故编码效率为4/5=80%

3.5.6 4B/5B编码

4B/5B编码规则有哪些?

4B/5B编码其实就是用5bit的二进制码来代表4bit二进制码。此编码的效率是80%，比Manchester码高。4B/5B编码的目的在前面已经说过了，就是让码流产生足够多的跳变。4位二进制共有16种组合，5位二进制共有32种组合，如何从32种组合种选取16种来使用呢？这里需要满足两个规则：

1). 每个5比特码组中不含多于3个“0”；

2). 或者5比特码组中包含不少于2个“1”；

3.5.6 4B/5B编码

4B/5B码举例

实用4B/5B码的最大优点是能够在很大程度上降低线路传输中的调制速率，从而降低了对线路的要求。

所谓调制速率是指单位时间内线路状态变化的数目，以波特(baud)为单位。

我们一般所说的数据信号传送速率是指单位时间内传送的信息量，以每秒多少比特(位)为单位，即bps（bit per second）。

注意：不要将数据传送速率与调制速率相混淆，如果1个调制时间间隔和1个代码相对应，则两者在数值上相等。如是多相调制，那么它们在数值上不同，例如曼彻斯特编码，它在每个调制时间间隔内跳动两次，故其波特率将是数据传送速率的两倍。

3.5.6 4B/5B编码

在快速以太网中，数据传输速率为100Mbps，如果采用曼彻斯特编码，波特率将达200Mbps，对传输介质和设备的技术要求都将提高，也增大了传输成本。

使用4B/5B编码后，在传输速率为100Mbps的情况下，其调制速率为：

$$100M \times (5/4) = 125M(baud)$$

即波特率仅为125M baud，大大低于曼彻斯特编码时的200M baud

十六进制数	4 位二进制数	4B/5B 编码	十六进制数	4 位二进制数	4B/5B 编码
0	0000	11110	8	1000	10010
1	0001	01001	9	1001	10011
2	0010	10100	A	1010	10110
3	0011	10101	B	1011	10111
4	0100	01010	C	1100	11010
5	0101	01011	D	1101	11011
6	0110	01110	E	1110	11100
7	0111	01111	F	1111	11101

3.5.6 4B/5B编码

8B/10B，也叫做8字节/10字节或8B10B。8B/10B方式最初由IBM公司于1983年发明并应用于ESCON(200M互连系统），由Al Widmer和Peter Franaszek在IBM的刊物“研究与开发”上描述。

8b/10b编码的特性之一是保证DC 平衡，采用8b/10b编码方式，可使得发送的“0”、“1”数量保持基本一致，连续的“1”或“0”不超过5位，即每5个连续的“1”或“0”后必须插入一位“0”或“1”，从而保证信号DC平衡，它就是说，在链路超时时不致发生DC失调。通过8b/10b编码，可以保证传输的数据串在接收端能够被正确复原，除此之外，利用一些特殊的代码(在PCI-Express总线中为K码)，可以帮助接收端进行还原的工作，并且可以在早期发现数据位的传输错误，抑制错误继续发生。

3.5.6 4B/5B编码

8b/10b编码是将一组连续的8位数据分解成两组数据，一组3位，一组5位，经过编码后分别成为一组4位的代码和一组6位的代码，从而组成一组10位的数据发送出去。相反，解码是将1组10位的输入数据经过变换得到8位数据位。数据值可以统一的表示为DX.Y或KX.Y，其中D表示为数据代码，K表示为特殊的命令代码，X表示输入的原始数据的低5位EDCBA，Y 表示输入的原始数据的高3位HGF。

8b/10b编码是目前许多高速串行总线采用的编码机制，如 USB3.0、1394b、Serial ATA、PCI Express、Infini-band、Fiber Channel、RapidIO等总线或网络等。

3.5.7 同步传输和异步传输

同步传输是一种以数据块为传输单位的数据传输方式，该方式下数据块与数据块之间的时间间隔是固定的，必须严格地规定它们的时间关系。每个数据块的头部和尾部都要附加一个特殊的字符或比特序列，标记一个数据块的开始和结束，一般还要附加一个校验序列，以便对数据块进行差错控制。

同步传输是以同步的时钟节拍来发送数据信号的，因此在一个串行的数据流中，各信号码元之间的相对位置都是固定的（即同步的）。

3.5.7 同步传输和异步传输

同步传输在同步传输的模式下，数据的传送是以一个数据区块为单位，因此同步传输又称为区块传输。

异步传输一般以字符为单位，不论所采用的字符代码长度为多少位，在发送每一字符代码时，前面均加上一个“起”信号，其长度规定为1个码元，极性为“0”，即空号的极性；字符代码后面均加上一个“止”信号，其长度为1或者2个码元，极性皆为“1”，即与信号极性相同，加上起、止信号的作用就是为了能区分串行传输的“字符”，也就是实现了串行传输收、发双方码组或字符的同步。

3.5.7 同步传输和异步传输

同步与异步传输的区别

1,异步传输是面向字符的传输，而同步传输是面向比特的传输。

2,异步传输的单位是字符而同步传输的单位是帧。

3,异步传输通过字符起止的开始和停止码抓住再同步的机会，而同步传输则是以数据中抽取同步信息。

4,异步传输对时序的要求较低，同步传输往往通过特定的时钟线路协调时序。

5,异步传输相对于同步传输效率较低。

3.5.7 同步传输和异步传输

简单说

同步传输就是，数据没有被对方确认收到则调用传输的函数就不返回。

接收时，如果对方没有发送数据，则你的线程就一直等待，直到有数据了才返回，可以继续执行其他指令

异步传输就是，你调用一个函数发送数据，马上返回，你可以继续处理其他事，

接收时，对方的有数据来，你会接收到消息，或者你的相关接收函数会被调用。

形象点说

异步传输: 你传输吧，我去做我的事了，传输完了告诉我一声 。

同步传输: 你现在传输，我要亲眼看你传输完成，才去做别的事。

3.5.8 AES3编码

AES/EBU的全称是Audio Engineering Society/European Broadcast Union（音频工程师协会/欧洲广播联盟），是一种通过基于单根数字双绞线来传输数字音频数据的串行位传输协议，其中AES是指AES3-2003标准（AES-1992的修订版）：《双通道线性表示的数字音频数据串行传输格式》，EBU是指EBU发表的数字音频接口标准EBU3250，两者内容在实质上是相同的，统称为AES/EBU数字音频接口。

按照我国的国家标准规定，采用串行接口数据格式，数字音频以 48 kHz 频率取样，每个取样值最多为 24 比特精度传输单声道或立体声节目，时钟基准和辅助信息同音频数据同时传送。该接口规定允许使用 32 kHz 或 44.1 kHz 取样信号。

编码方式采用双相位标记编码，也有的地方叫双相标志编码，编码原理前文已经说过.

3.5.8 AES3编码

GY

中华人民共和国广播电影电视行业标准

GY/T 158－2000

演播室数字音频信号接口

Digital audio signal interface for broadcasting studios

3.5.8 AES3编码

传输距离

但是AES建议数字双绞线只适用于短距离（100米）左右的AES/EBU信号传输，而现在随着广播业务的扩大，许多电台的技术机房分布越来越广泛，机房间的传输距离往往超过100米，这时使用AES/EBU信号传输就会有风险，AES协会的解决建议是采用非平衡的75Ω同轴电缆进行传输（AES-3id-2001），最长传输距离可达1000米。但是在使用这种方式传输时，应该在线路的两头使用一种110Ω/75Ω的阻抗适配器，即一头是卡侬XLR接头（公/母），另一头为BNC插座。这种阻抗适配器较为常见，但是质量差别很大，劣质产品会对信号产生较大的衰减，符合广播播出级别的需要经过严格测试。一般来说，75Ω同轴线缆成本与数字双绞线差不多，但是符合广播级别的阻抗适配器价格并不便宜，而且使用量很大（每个音频通路需要两个），必然增加布线成本，并且同轴线缆的线径一般较粗，需要更多的布线空间。所以，在实际使用中，采用同轴电缆方案布线的十分少见。当然，在更长的距离上，也可以使用光端机将AES/EBU信号转换成光信号进行传输。但是光缆传输方式，必然需要增加传输设备的成本，并且增加以及设备也增加了故障环节，也并非良策。

3.5.8 AES3编码

数字双绞线传输AES/EBU信号是否真的只能在100米之内呢？若传输距离超过100米，是否AES/EBU信号就不能传输呢？在笔者所在电台近期进行的一次技术机房改造项目中，就遇到这个问题。此次改造中，相距最远距离的机房之间布线长度接近150米，而由于布线空间和经费有限，不适于采用同轴电缆布线方案，对于网络音频传输及MADI传输方式又有设备成本和系统安全等方面的考虑没有采用。若要采用数字双绞线传输，必须验证这种方式的安全性和可靠性。

3.5.8 AES3编码

根据我国广播电影电视行业标准GY/T 224-2007《数字视频、数字音频电缆技术要求和测量方法》：

根据该标准，该表中数据是在“最低允许的输出信号振幅为2V，最低允许的输入信号振幅为200mV”的情况下得出的，即与AES/EBU标准的最低输出信号振幅和最低输入信号振幅要求一致。因此，该标准建议48kHz数字音频信号使用规格为26AWG（美国线规，导线直径约为0.404毫米）的数字音频双绞线传输时，最远可以传输242米；而若使用24 AWG（美国线规，导线直径约为0.511毫米）的数字音频双绞线传输时，最远可以传输334米；若使用22 AWG（美国线规，导线直径约为0.643毫米）的数字音频双绞线传输时，最远可以传输465米。这个数据可大大超出了AES规定100米，并且随着线缆导线的加粗，传输的距离也会增强。但是，该标准采用是的测量衰减常数的方法，这样得出的数据是否能够直接应用于AES/EBU信号的传输，并且，在实际环境中，存在着各类干扰、电缆的多处弯折以及多个跳接环节造成的传输损耗后，传输距离还能达到200米、300米、甚至400米吗?

3.5.8 AES3编码

在实际环境中，存在着各类干扰、电缆的多处弯折以及多个跳接环节造成的传输损耗后，传输距离还能达到200米、300米、甚至400米吗?

1) 数字信号幅值（Vp-p）：描述数字信号的峰峰电压值，根据AES3规定，设备输出信号幅值在2V-7V之间，设备接收信号不小于200mV。

3.5.8 AES3编码

2) 抖动：表示数字信号在时间周上的位置差异，如抖动幅度过大，易造成误码，单位常用UI，UI是指一个双相位码的持续时间。

在AES/EBU数字音频信号中，音频信息以数据帧的方式传输，其中每个数据帧包含左、右两个子帧，并以串行的方式排列传输，每个子帧含有32bit数据。在48kHz的采样频率下，传输码率为：

32*48000*2=3.072Mbps

则每帧中的每一个bit数据持续时间为：

1/3.072 Mbps =325.5ns

AES/EBU格式采用双相位码来进行编码，即用2位二进制数来表示一个bit数据。

则一个双相位码的持续时间为每个bit数据的一半，即：

1UI=325.5ns/2=163ns

3.5.8 AES3编码

3)Confidence Error：The logical OR of UNLOCK and BIP. The input data stream may be near error condition due to jitter degradation。具体是指UNLOCK参数报错（指锁相环失锁）或者BIP参数报错（指接收到的双相位码编码错误），这意味着输入数据流由于Jitter劣化已经处于错误边缘。反映线路传输过长，容性负载过大造成传输信号高频缺失，易造成误码。

4) UNLOCK：锁相环失锁，即信号失锁，反映信号频率异常，通常为频率不稳，或偏离基准频率过多。

3.5.8 AES3编码

5)BIP Error：双相位码编码错误，即接收解码错误。

6)Parity Error：奇偶校验错误。

7)CRC Error：循环冗余校验码错误。AES3格式中，将192个数据桢组成一个数据块传输，每个数据块末尾包含CRC校验，如果接收端计算的CRC值不等于数据块中的CRC值，则表示传输中出现了误码。

3.6 AES10(MADI)多通道数字声频接口格式

将AES3协议进行了延伸，可以在50米的距离内通过一根BNC端口的电缆串行传输56个通道的线性量化音频数据。

声频量化比特数可以为24比特。

MADI没有采用AES3的双相位标志编码，而是采用了具有4B/5B编码格式的NRZI编码（基于FDDI协议）。链路传输率为125Mbps，数据传输率为100Mbps。

MADI采用的是异步传输模式。所以发射机和接收机要采用缓存器，所有互联的发射机和接收机必须使用分配的主同步信号。

3.6.1 CRC（循环冗余校验）

什么是CRC校验?

CRC即循环冗余校验码：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查（CRC）是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

3.6.1 CRC（循环冗余校验）

CRC校验原理：

其根本思想就是先在要发送的帧后面附加一个数（这个就是用来校验的校验码，但要注意，这里的数也是二进制序列的，下同），生成一个新帧发送给接收端。当然，这个附加的数不是随意的，它要使所生成的新帧能与发送端和接收端共同选定的某个特定数整除（注意，这里不是直接采用二进制除法，而是采用一种称之为“模2除法”）。到达接收端后，再把接收到的新帧除以（同样采用“模2除法”）这个选定的除数。因为在发送端发送数据帧之前就已通过附加一个数，做了“去余”处理（也就已经能整除了），所以结果应该是没有余数。如果有余数，则表明该帧在传输过程中出现了差错。

3.6.2 模2运算

5. CRCC

在介绍CRCC(Cyclic Redumdancy Check Code，循环冗余校验码)之前，先来看一下二进制数的多项式表示和它的运算。

a. 二进制数的多项式表示

将1991这样的十进制数用位权值和位值表示则为：

$$1991 = 1\times10^3 + 9\times10^2 + 9\times10^1 + 1\times10^0 \tag{10-7}$$

同理，二进制数1000100101可以表示为

$$1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1 = 1\times x^9 + 0\times x^8 + 0\times x^7 + 0\times x^6 + 1\times x^5$$

$$x^9\ x^8\ x^7\ x^6\ x^5\ x^4\ x^3\ x^2\ x^1\ x^0$$

$$+ 0\,x^4 + 0\times x^3 + 1\times x^2 + 0\times x^1 + 1\times x^0$$

$$= x^9 + x^5 + x^2 + 1 \tag{10-8}$$

对于自然二进制码，将x代以2就可得到它的值，但也可以用其它二进制码，因此用x表示比较方便。

b. 二进制数的运算

由于二进制运算不进位，即1+1=0，所以减法与加法结果相同，即没有减法。

(1) 移位：例如将信息序列$M(x) = x^9 + x^5 + x^2 + 1$乘以$x^5$时，就对应于将该序列向前移5位，得

$$x^5M(x) = x^5(x^9 + x^5 + x^2 + 1) = x^{14} + x^{10} + x^7 + x^5 \tag{10-9}$$

(2) 加法[求模2(mode2)和，即1+1=0]

两个多项式之和是同幂项抵消，其它则与一般算术运算相同(模2和用⊕符号表示。)

例如，设

$$M(x) = x^9 + x^5 + x^2 + 1 \tag{10-10a}$$

$$N(x) = x^8 + x^5 + x^3 + 1 \tag{10-10b}$$

则

$$M(x)\oplus N(x) = x^9 + x^8 + \underbrace{(x^5 + x^5)}_{\parallel\ 0} + x^3 + x^2 + \underbrace{(1+1)}_{\parallel\ 0}$$

$$= x^9 + x^8 + x^3 + x^2 \tag{10-10c}$$

(3) 乘法

设

$$F(x) = x^2 + x + 1 \tag{10-11a}$$

$$G(x) = x^5 + x^4 + x^2 + 1 \tag{10-11b}$$

则

$$F(x)\times G(x) = (x^2 + x + 1)(x^5 + x^4 + x^2 + 1)$$

$$= x^2(x^5 + x^4 + x^2 + 1) + x(x^5 + x^4 + x^2 + 1) + (x^5 + x^4 + x^2 + 1)$$

$$= x^7 + x^6 + x^4 + x^2 + x^6 + x^5 + x^3 + x + x^5 + x^4 + x_2 + 1$$

$$= x^7 + x^3 + x + 1 \tag{10-11c}$$

(4) 除法

3.6.3 模2除法

模2除法：

模2除法与算术除法类似，但每一位除的结果不影响其它位，即不向上一位借位，所以实际上就是异或。在循环冗余校验码（CRC）的计算中有应用到模2除法。

例：

3.6.4 CRC校验步骤

模2运算通常用生成二进制来表示。

CRC校验步骤：

CRC校验中有两个关键点，一是预先确定一个发送送端和接收端都用来作为除数的二进制比特串（或多项式），可以随机选择，也可以使用国际标准，但是最高位和最低位必须为1；二是把原始帧与上面计算出的除数进行模2除法运算，计算出CRC码。

3.6.4 CRC校验步骤

具体步骤：

1. 选择合适的除数

2. 看选定除数的二进制位数，然后再要发送的数据帧上面加上这个位数-1位的0，然后用新生成的帧以模2除法的方式除上面的除数，得到的余数就是该帧的CRC校验码。注意，余数的位数一定只比除数位数少一位，也就是CRC校验码位数比除数位数少一位，如果前面位是0也不能省略。

3. 将计算出来的CRC校验码附加在原数据帧后面，构建成一个新的数据帧进行发送；最后接收端在以模2除法方式除以前面选择的除数，如果没有余数，则说明数据帧在传输的过程中没有出错。

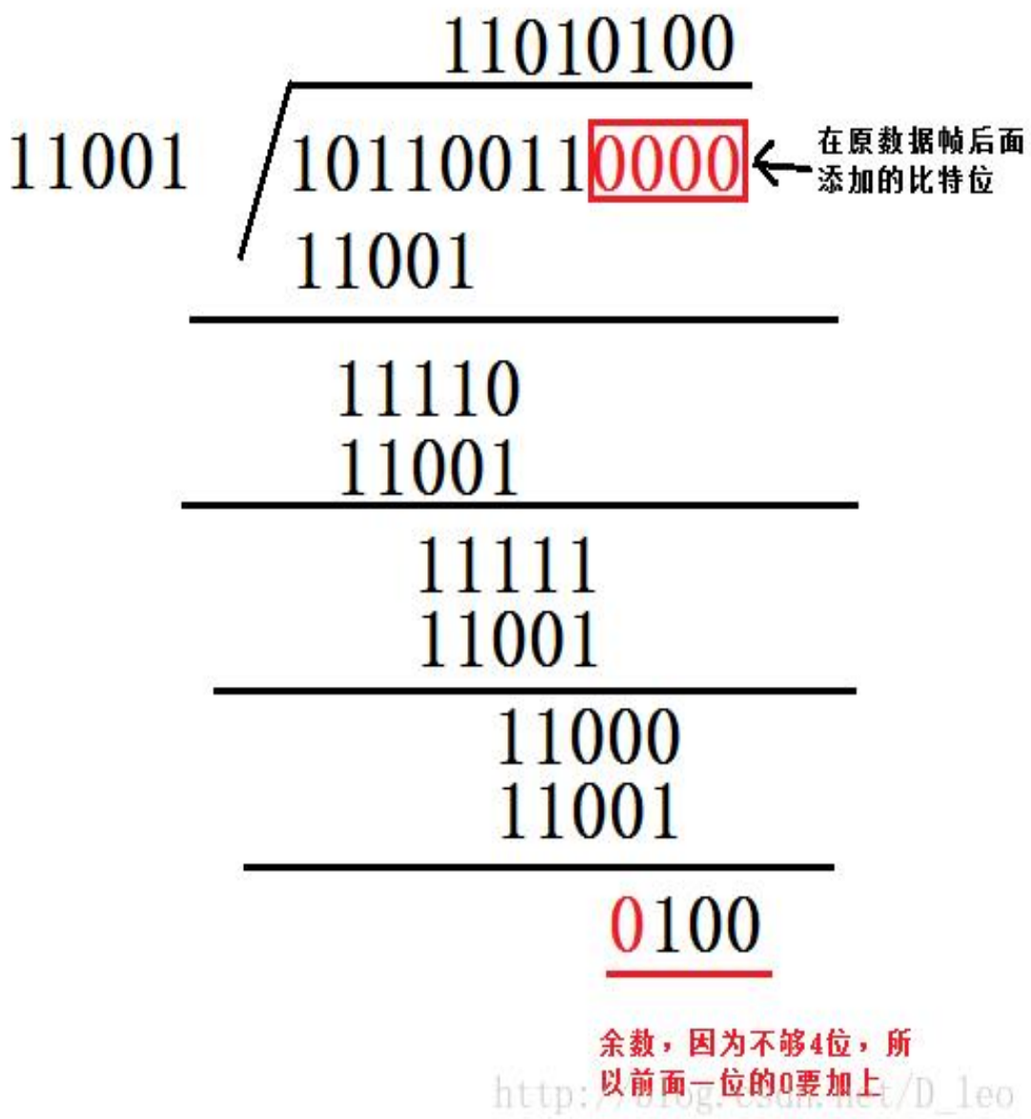
3.6.4 CRC校验步骤

CRC校验码计算示例：

现假设选择的CRC生成多项式为G（X） = X4 + X3 + 1，要求出二进制序列10110011的CRC校验码。

①将多项式转化为二进制序列，由G（X） = X4 + X3 + 1可知二进制一种有五位，第4位、第三位和第零位分别为1，则序列为11001

②多项式的位数位5，则在数据帧的后面加上5-1位0，数据帧变为101100110000，然后使用模2除法除以除数11001，得到余数。



3.6.4 CRC校验步骤

③将计算出来的CRC校验码添加在原始帧的后面，真正的数据帧为101100110100，再把这个数据帧发送到接收端。

④接收端收到数据帧后，用上面选定的除数，用模2除法除去，验证余数是否为0，如果为0，则说明数据帧没有出错。

生成多项式列表

表10-6 CRCC举例

名　　称	多 项 式 表 示	位　　表　　示
信息多项式M(x)	$x^8+x^5+x^2+1$	100100101
$x^5M(x)$	$x^{14}+x^{10}+x^7+x^5$	100100101000000
生成多项式G(x)	$x^5+x^4+x^2+1$	110101
商多项式Q(x)	$x^9+x^8+x^7+x^5+x^2+x+1$	1110001111
剩余多项式R(x)	$x+1$	11
发送多项式U(x)	$x^{14}+x^{10}+x^7+x^5+x+1$	100100101000011
		$x^{14}x^{13}x^{12}x^{11}x^{10}x^9x^8x^7x^6x^5x^4x^3x^2x^1x^0$

自定义

自定义	
CRC-4/ITU	x4+x+1
CRC-5/EPC	x4+x3+1
CRC-5/ITU	x5+x4+x2+1
CRC-5/USB	x5+x2+1
CRC-6/ITU	x6+x+1
CRC-7/MMC	x7+x3+1
CRC-8	x8+x2+x+1
CRC-8/ITU	x8+x2+x+1
CRC-8/ROHC	x8+x2+x+1
CRC-8/MAXIM	x8+x5+x4+1
CRC-16/IBM	x16+x15+x2+1
CRC-16/MAXIM	x16+x15+x2+1
CRC-16/USB	x16+x15+x2+1
CRC-16/MODBUS	x16+x15+x2+1
CRC-16/CCITT	x16+x12+x5+1
CRC-16/CCITT-FALSE	x16+x12+x5+1
CRC-16/X25	x16+x12+x5+1
CRC-16/XMODEM	x16+x12+x5+1
CRC-16/DNP	x16+x13+x12+x11+x10+x8+x6+x5+x2+1
CRC-32	x32+x26+x23+x22+x16+x12+x11+x10+x8+x7+x5+x4+x2+x+1
CRC-32/MPEG-2	x32+x26+x23+x22+x16+x12+x11+x10+x8+x7+x5+x4+x2+x+1

谢谢大家