

基于虚拟仪器的数字基带通信系统设计

摘要

从数据终端来的原始数据信号，如计算机输出的二进制序列、电传机输出的代码，或者是来自模拟信号经数字化处理后的 PCM 码组、AM 序列等都是数字信号，这些信号往往包含丰富的低频分量或者直流分量，因而称为数字基带信号。在某些具有低通特性的有线信道中，特别是传输距离不太远的情况下，可以直接传输数字基带信号，这个过程被称为数字基带传输。而大多数信道，如各种无线信道和光信道，则是带通型，数字基带信号必须经过载波调制后，把频谱搬移到高载频处这才可以在信道中传输，通常把这种传输称为数字频带（调制或载波）传输。

目前，虽然在实际应用场合，数字基带传输不如频带传输那样广泛，但对于基带传输系统的研究仍是十分有意义的。一是因为在利用对称电缆构成的近程数据通信系统广泛采用了这种传输方式；二是因为数字基带传输中包含频带传输的许多基本问题，同时，基带传输系统的许多问题也是频带传输系统必须考虑的问题；三是因为任何一个采用线性调制的频带传输系统可等效为基带传输系统来研究。

关键词：数字基带通信；虚拟仪器；基带传输系统

引言

基带传输系统的输入端是脉冲序列的终端设备或信道编码器，信号发生器是基带信号的固有变换。这种转换主要是通过对目标码的转换来实现的，这是匹配信道、便于传输、减少符号间串扰的关键技术。第2节中介绍了信道信号形成器里的波形变换是数字基带通信系统的发送滤波器，其主要功能是使其与信道和接收滤波器一起，满足无码间干扰条件这即是奈奎斯特第一准则。基带信道是允许基带信号通过的媒质，信道的传输特性是随机变化的，一般不满足无失真传输条件，另外信道还会进入噪声，而在通信系统的分析中，我们常常把噪声 $n(t)$ 等效集中在信道中引入。第3节分析和设计了部分响应系统，详细介绍了每一部分的功能与实现，在第4节将利用眼图分析噪声对数字基带通信系统性能的影响，接收滤波器的主要作用是滤除带外噪声，对信道特性均衡，使输出的基带波形有利于抽样判决。抽样判决器是在传输特性不理想及噪声背景下，为了恢复或再生基带信号，在规定时刻（由位定时脉冲控制）对接收滤波器的输出波形进行抽样判决。而用来抽样的位定时脉冲则依靠同步提取电路从接收信号中提取，位定时的准确与否将直接影响判决效果。最后在第5节里根据现有的虚拟仪器设计了综合的数字基带通信系统。

1 基带数字信号的码变换器

1.1 基带数字信号的码变换原理

现实的基带传输系统中，不是所有代码的电波形都可以在信道中进行传输。例如，含有直流分量和较丰富低频分量的单极性基带波形就不适宜在低频传输特性差的信道中传输，有可能造成信号严重畸变。当消息代码中含有长串又连续“1”或“0”符号时，非归零波形呈现出连续而又的固定电平，从而无法获取定时信息。同时单极性归零码在输送连“0”时，也存在相同的问题。因此，对传输用的基带信号主要有两个方面的要求：

- (1) 对代码的要求，原始消息代码必须编成合适传输的码型；
- (2) 对所选码型的电波形要求，电波形应适用在基带系统的传输。

而前者属于传输码型的选择，后者则是对于基带脉冲的选择，这是两个既独立而有联系的问题。传输码（或称线路码）的结构将取决于系统工作条件和实际信道特性。通常，传输码的结构应具有下列主要特性：

- (1) 相应的基带信号无直流分量，且低频分量少；
- (2) 便于从信号中提取定时信息；
- (3) 信号中高频分量越少越好，这可以减少码间串扰并节省带宽；
- (4) 能适应信息源的变化，不受信息源统计特性的影响；

(5) 具有内在的检错能力，传输码型应具有一定的规律，以便利用这一规律进行监测；

(6) 编译码设备要尽可能简单等。

满足或部分满足以上特性的传输码型种类繁多，常用码型有下列几种。

1.1.1 AMI 码

AMI 码是传号交替反转码，其编码规则是将二进制消息代码“1”（传号）交替地变换为传输码的“+1”和“-1”，而“0”（空号）则保持不变。例如：表格 1.1 消息代码转 AMI 码

消息代码:	1	0 0 1	1	0 0 0 1	1	1	...
AMI 码:	+1	0 0 -1	+1	0 0 0 -1	+1	-1	...

AMI 码的传号交替反转，它所决定的基带信号将出现正、负脉冲交替，而“0”电位保持不变的规律。由此看出，这种基带信号没有直流成分，只有很小的低频成分，因此它特别适合在不允许这些成分通过的信道中传输。

由 AMI 码的编码规则可以看出，它从一个二进制符号序列变成了一个三制符号序列，即是一个二进制符号变换成一个三进制符号。把一个二进制符号变换成一个三进制符号所构成的码型被称为 1B/1T 码型。

AMI 码除了有上述特点外，还有便于观察误码情况及编译码电路简单等优点，它是一种基本的线路码，这使得其在高密度信息流的数据传输中，得到了较为广泛的应用。

AMI 码的不足之处是，当原信码出现连“0”串时，信号的电平会长时间的不跳变，造成提取定时信号时困难，而解决连“0”码问题的方式之一就是采用 HDB3 码。

1.1.2 HDB3 码 HDB3 码的全称是三阶高密度双极性码，其编码原理为：先检查消息代码（二进制）的连“0”串情况，当没有 4 个或 4 个以上连“0”串时按照 AMI 码的编码规则进行编码；当出现 4 个或 4 个以上连“0”串时，则将第 4 个连“0”小段的第 4 个“0”变换成与其前一非“0”符号（+1 或 -1）同极性的符号。显然，这样做会破坏“极性交替反转”的规律，这样符号就称为破坏符号，表示方式为 V 符号（即+1 记为+V，-1 记为-V）。为使附加 V 符号后序列不破坏“极性交替反转”而形成的无直流特性，还必须保证其相邻 V 符号也应该产生极性交替。但当相邻 V 符号之间有奇数个非“0”符号时，这个是可以的；而当有偶数个非“0”符号时，则无法保证，这时再将该小段的第 1 个“0”变换成+B 或-B，B 符号的极性与前一非“0”符号的相反，并让后面的非“0”符号从 V 符号开始再交替变化。

其编码规则如下：

(1) 当信码的连“0”个数不超过 3 时，仍按 AMI 码的规则编，即传号极性交替。

(2)当连“0”个数超过3时,则将第4个“0”改为非“0”脉冲,记为+V或-V,称为破坏脉冲,V的极性与前一非零码的极性相同。相邻V码的极性交替出现,确保编好的码中无直流。

(3)当V码之间有奇数个非“0”符号时,能保证相邻V码的极性交替;否则,当V码之间有偶数个(或零个)非0符号时,不能保证相邻V码的极性交替,将该4个连“0”的第一个“0”更改为B,B的符号与B的前一非0符号极性相反,V码与B码极性相同。

(4)破坏脉冲之后的传号码极性也要交替。例如:表格 1.2 编码示例

代码:	1000	0	1000	0	1	1	000	0	1	1
AMI 码:	-1000	0	+1000	0	-1	+1	000	0	-1	+1
HDB3 码:	-1000	-V	+1000 +V	-1	+1	-B	00	-V	+1	-1

其中的±V脉冲、±B脉冲与±1脉冲波形相同,用V或B符号的目的是为了示意将原信码的“0”变换成“1”码。HDB3码的编码规则比较复杂,但是译码却比较简单,因为从上述原理不难看出,HDB3码也是1B/1T码,而每一个破坏符号V总与前一非0符号相同极性(包括B在内)。

HDB3码还原规则:从收到的符号序列中找到与前一非“0”符号同号的,这使得可以更加轻松的找到破坏点V,于是也可以决定了V符号及其前面的3个符号必是连“0”符号。将V还原为0,然后恢复4个连“0”码;若V前3位也是非“0”符号,即存在B,则将B也还原为0;再将所有-1和+1变成1后便得到原消息代码。

HDB3码的特点:HDB3码保持了AMI码无直流的优点,同时还将连“0”码限制在3个以内,便于传输和位定时信号的提取。HDB3码是应用最为广泛的码型,在实际中A律PCM4次群以下的接口码型均为HDB3码,也是CCITT推荐码之一。

1.1. 3PST 码

PST码的全称是成对选择三进制码,它的编码过程是:先将二进制的代码划分成两个码元为一组的码组序列,然后再把每一码组编码成两个三进制数字(+0)。因为两位三进制数字共有9种状态,故可灵活地选择其中的4种状态,将00和11分别用-+和+-表示,将10和01分别用+0和0+ (+模式)或-0和0- (-模式)表示。通常为了防止PST码产生直流漂移,我们应交替变换这两个模式。

例如:

表格 1.3 PST码转换

代码:	10	00	11	01	10	11	00	11
取十模式开始时:	+0	-+	+-	0-	+0	+-	-+	+-
取一模式开始时:	-0	-+	+-	0+	-0	+-	-+	+-

1.1.4 双相码(又称 Manchester 码, 即曼彻斯特码)

双相码是对每个二进制代码分别用两位二进制新码去取代的码, 而它的编码规则之一是:

0→01 (零相位的一个周期的方波)

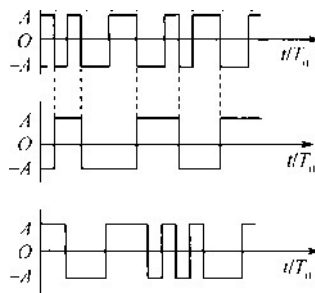
0→10 (π 零相位的一个周期的方波) 例如:

表格 1.4 双相码转换

代码:	1	1	0	0	1	0
双相码:	10	10	01	01	10	01

(c) CMI 码双相码的特点是只使用两个正、负电平, 而不像前面的 3 种码具有 3 个电平。这种码的编码过程简单, 是因为其既无直流漂移, 又能够提供足够的定时分量, 但这种码的带宽要比较上述的编码方式来说要宽一些。上述码又称为绝对双相码, 还有一种码与其相对应, 被称为差分双相码。差分双相码原理是先把输入的 NRZ (非归零) 波形变换成差分波形, 再用差分波形实行绝对双相

双相码图 1.1 码波形



码编码, 此时的输出码相对于输入 NRZ 波形, 被称为差分双相码。该码在本地局域网中常被 (b) 密勒码使用。

双相码的码波形如图 1.1 (a) 所示。

1.1.5 密勒 (Miller) 码

密勒码又被称延迟调制码, 是双相码的一种变形。编码规则如下: “1” 码用码元持续时间中心点出现跃变来表示, 即用 “10” 或 “01” 表示; “0” 码分两种情况处理: 对于单个 “0” 时, 在码元持续期间内不出现电平跃变, 且与相邻码元的边界处也不跃变; 对于连 “0” 时, 在两个 “0” 码的边界处出现电平跃变, 即 “00” 与 “11” 交替。由图 1.1 (b) 是代码序列为 11010010 时密勒码和双相码的波形, 若两个 “1” 码中间有一个 “0” 码时, 密勒码流的波形最大宽度为 $2T_s$, 即两个码元周期。我

们通常利用这一性质来完成误码检测。比较图 1.1(a)和(b)两个波形不难看出，双相码的下降沿正好对应于密勒码的跃变沿，因此用双相码的下降沿去触发双稳电路，即可输出密勒码。密勒码最初用于气象卫星和磁记录，现在也用于低速基带数传机中。

1.1.6CMI 码

CMI 码是传号反转码的简称,与双相码类似,它也是一种双极性二电平码。编码规则是:“1”码交替用“11”和

“00”两位码表示;“0”码固定地用“01”表示,其波形图如图 1.1(c)所示。

可以看出 CMI 码的电平跃变较多,包含的定时信息非常丰富。由于存在禁用码组 10,所以不会出现 4 个以及 4 个以上的连码,利用这个特点我们可以实现宏观检错功能。由于 CMI 码易于实现,且具有上述特点,因此是 CCITT 推荐的

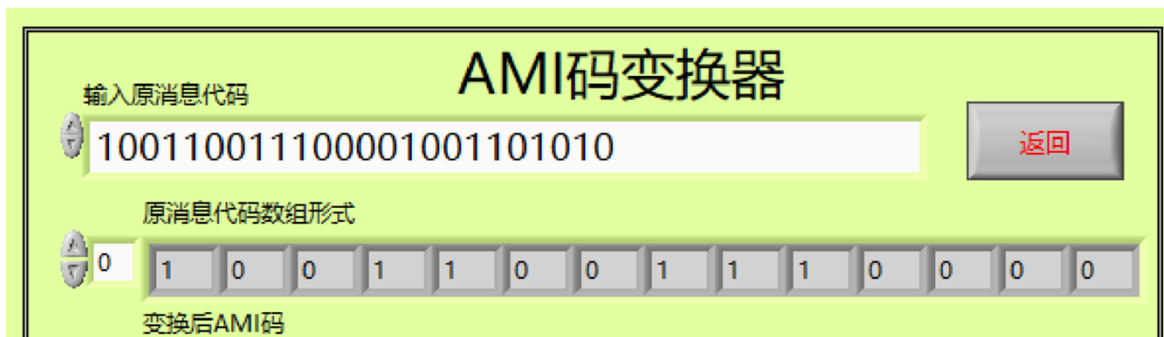
PCM 高次群采用的接口码型,在速率低于 8.448Mbps 的光纤传输系统中有时也用作线路传输码型。CMI 码的码波形如图

1.1(c)所示。

在上述三个码中,每个原二进制信码都用一组两位的二进制码表示,因此这类码又称为 1B/2B 码。

1.2AMI 码的仿真

在设计常用码型变换器时,都采用了统一的前面板设计风格,如 AMI 码、HDB3 码、PST 码、双相码、Miller 码和 CMI 码等。图 1.2 就是 AMI 码变换器的前面板。该前面板基本上包括 5 个控件,其中有一个输入控件、两个输出控件、一个显示控件和一个返回按钮。



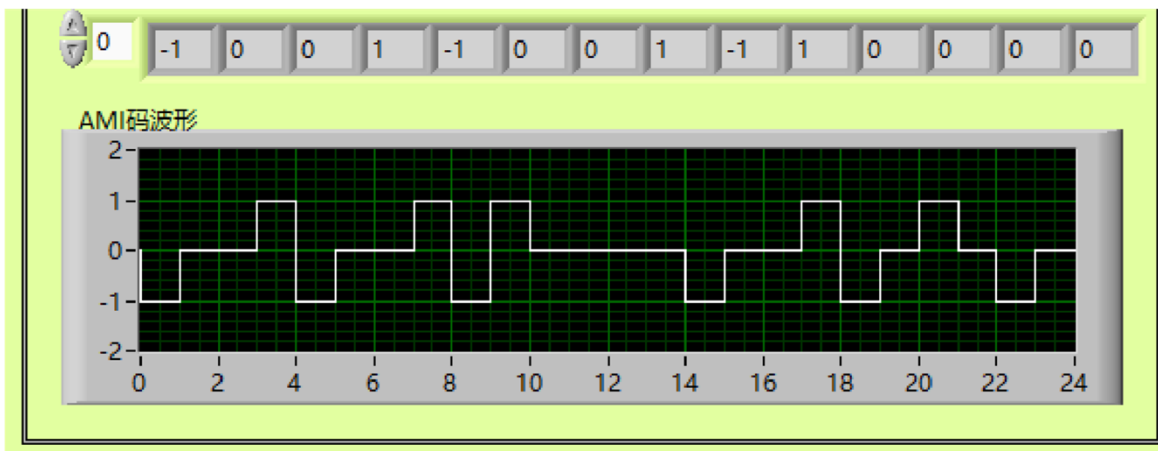


图 1.2 AMI 码变换器的前面板

输入控件是“输入原消息代码”这一组件，它的数据类型是 32 位无符号的二进制数，程序运行过程中，我们应先对其输入要进行转换的二进制数。在输入控件下方就是“原消息代码数组形式”和“变换后 AMI 码”这两个输出控件，前一个的数据类型是 8 位无符号的二进制数，后一个是 8 位有符号数。它们主要是将位与位分开，对照显示变换前和变换后的数据。“AMI 码波形”是一个显示控件，功能是显示代码转换后的输出波形。

AMI 码变换器后面板编程思路为：

1. 将代码转化为布尔型数组；
2. 从数组的第 31 位索引值开始提取布尔量；
3. 将第一次布尔量为“真”的索引值加 1；
4. 这个值就是代码的有效位数。

“消息代码”为 32 位无符号二进制数，利用“Number To Boolean Array”函数将其转化成 32 个元素的布尔型数组，这个数组中的数据与原代码是倒序排列的，在输入数值不够 32 位的时候，数组索引值较大的部分用“False”来赋值。在程序的循环部分，用 $31-i$ 来确定索引值，从数组的第 31 个元素开始判断是否为“True”，当遇到第一个为“True”值时，说明之后的所有元素均为有效代码，这个索引值加 1 便是有效位数，这个程序在同一时间结束。

“数转数组.vi”的后面板功能是首先将“Number To Boolean Array”函数将消息代码转化成 32 个元素的布尔型数组，从这个数组的第 0 索引值开始提取，共提取有效位个数，将提取的这个子数组用“Boolean To(0, 1)”函数转换成数值型数组，再将该数组倒序排列，得到二进制数的数组形式。

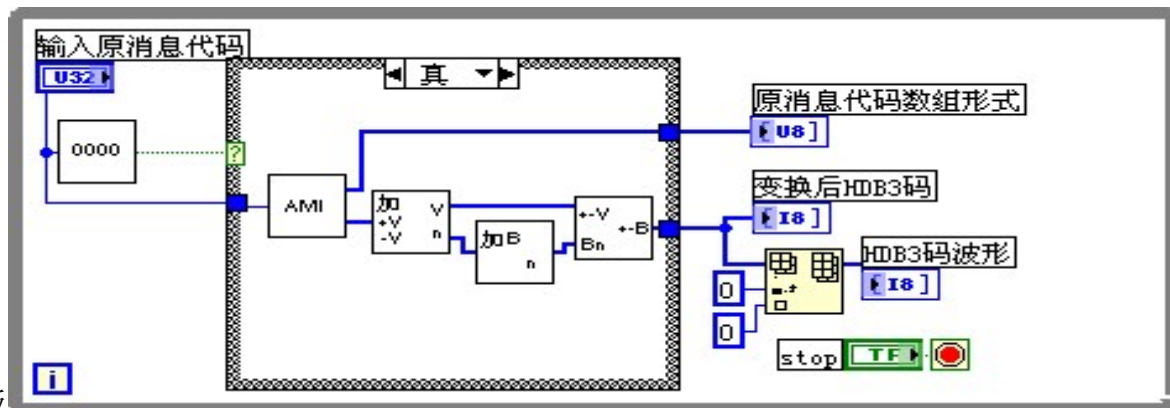
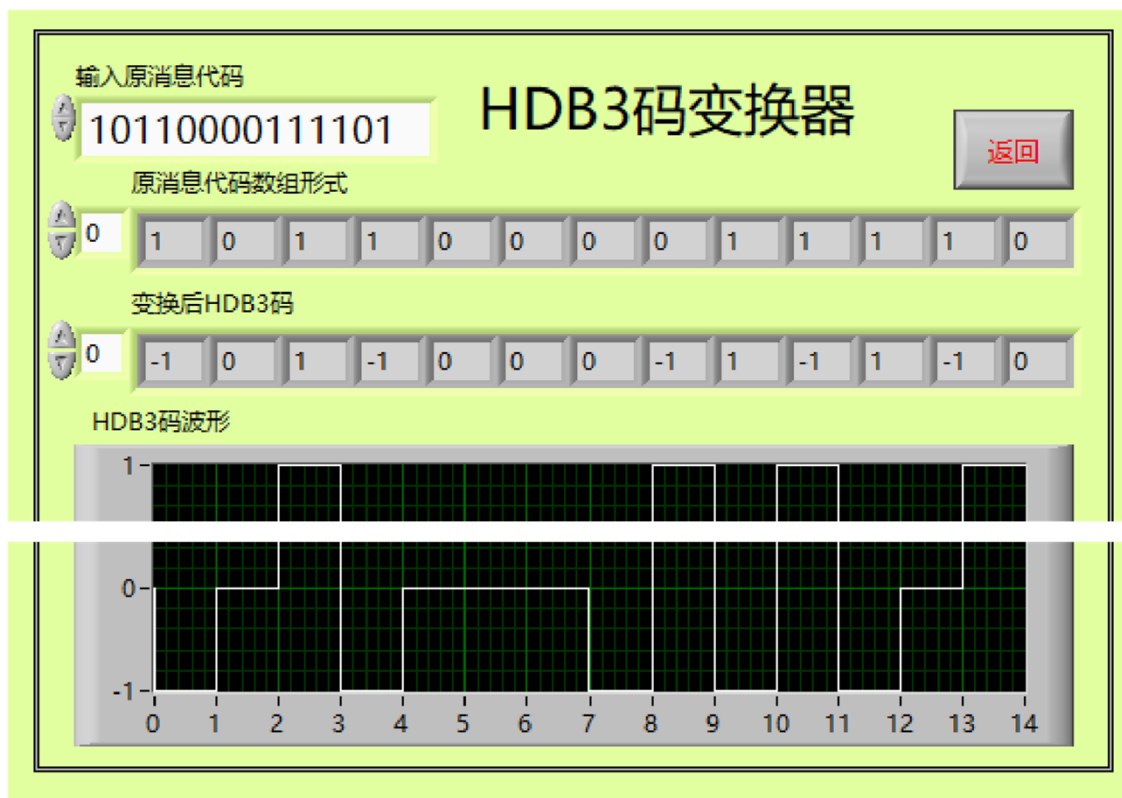
判断是否被 X 整除.vi 后面板里面的“除数”和“被除数”均为浮点型数据，在前面板输入两个数相除。

“Round To-Infinity” 节点为取输入浮点数的最小整数，当除得的商与这个最小整数相等时，即为整除, 输出为

“True”。

1. 3HDB3 码的仿真

图 1. 3(a) 、(b)是 HDB3 码变换器的前、后面板。HDB3 码变换器的输入、输出控件与 AMI 码基本相同，所以这里不再对 HDB3 码变换器的前面板进行描述。



(a) 前面板

(b) 后面板

图 .3 HDB3 码变换器

HDB3 码变换器的整体设计要比 AMI 码复杂得多。首先要判断输入的消息代码是否有 4 个或 4 个以上的连。出现(用 “判断是否四连零.vi” 实现), 如果没有, 则通过 Case 语句中的 “False”, 按照 AMI 码的规则输出; 如果为真, 则要
通过图中的 4 个子 VI 来确定: “AMI.vi”、“加+V-V.vi”、“加 B 索引值数组.vi”、“加+B-B.vi”。

- (1) “AML.vi”: 将 AMI 码变换器去掉循环, 制作成子程序, 功能是将输入原消息代码变换成 AMI 码。
- (2) “加+V-V.vi”: 功能是在 AMI 码中加入破坏符号 V, 并确定加+V、-V 处的索引值。
- (3) “加 B 索引值数组.vi”: 功能是通过加 V 处的索引值来确定加 B 处的索引值。
- (4) “加+B-B.vi”: 功能是将 B 加入代码中, 并确定其符号。

HDB3 码中包括的子程序分别介绍如下:

“加+V-V.vi” 的后面板如图 1.1.4 所示。

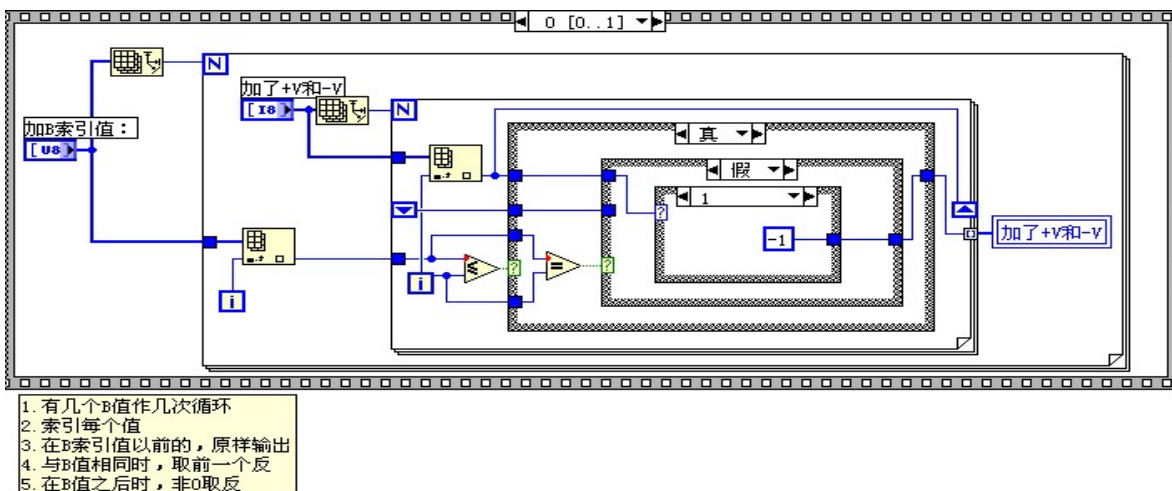


图 1.4 加+V-V.vi

“判断是否整除.vi”, 输出为布尔量。若能整除, 输出为 “真” (True); 否则, 输出为 “否” (False)。

“加 B 索引值数组.vi” 后面板。在外层 For 语句中, 循环次数由 “加 B 索引值” 数组的个数确定。里层 For 语句对加了 V 之后的代码进行逐一判断输出, 当 $i < B$ 时, 代码原样输出; 当 $i = B$ 时, 与前一码元相反; 当 $i > B$ 时, 非 0 符号取反输出。在该程序的第二页, 只是将局部变量 “加了+V 和-

V”赋给“加了+B 和-B（即+1 和-1）”输出控件。

加+B—B. vi 后面板编程思路为:

- (1)有几个 B 值做几次循环;
- (2)索引每个值;
- (3)在 B 索引值以前的原样输出;
- (4)与 B 值相同时,取前一个反;
- (5)与 B 值不相同,非 0 取反。

判断是否四连零. vi 后面板编程思路为从 0 索引值开始提取,每次取 4 个元素,并于常量 4 个元素的 0 值数组进行判断:当 4 个元素都相等时,经“Boolean Array To Number”变成一整数,如果等于 15(即二进制数 1111)为真,输出布尔量“真”;反之,输出布尔量“否”

1. 4PST 码的仿真

PST 码变换器的前面板如图 1.5 所示。PST 码是先将二进制代码划分成两个码元为一组的码组序列,在这里是变成两列的二维数组,每次从一行数组中分别取出一个元素来确定是何种组合,当码元为“00”时取“-”;当码元为“11”时取“+”;当码元为“01”或“10”时,+模式与-模式交替出现,用局部变量“模式交替”是否被 2 整除来判断。最后再将二维数组变成一维数组输出。

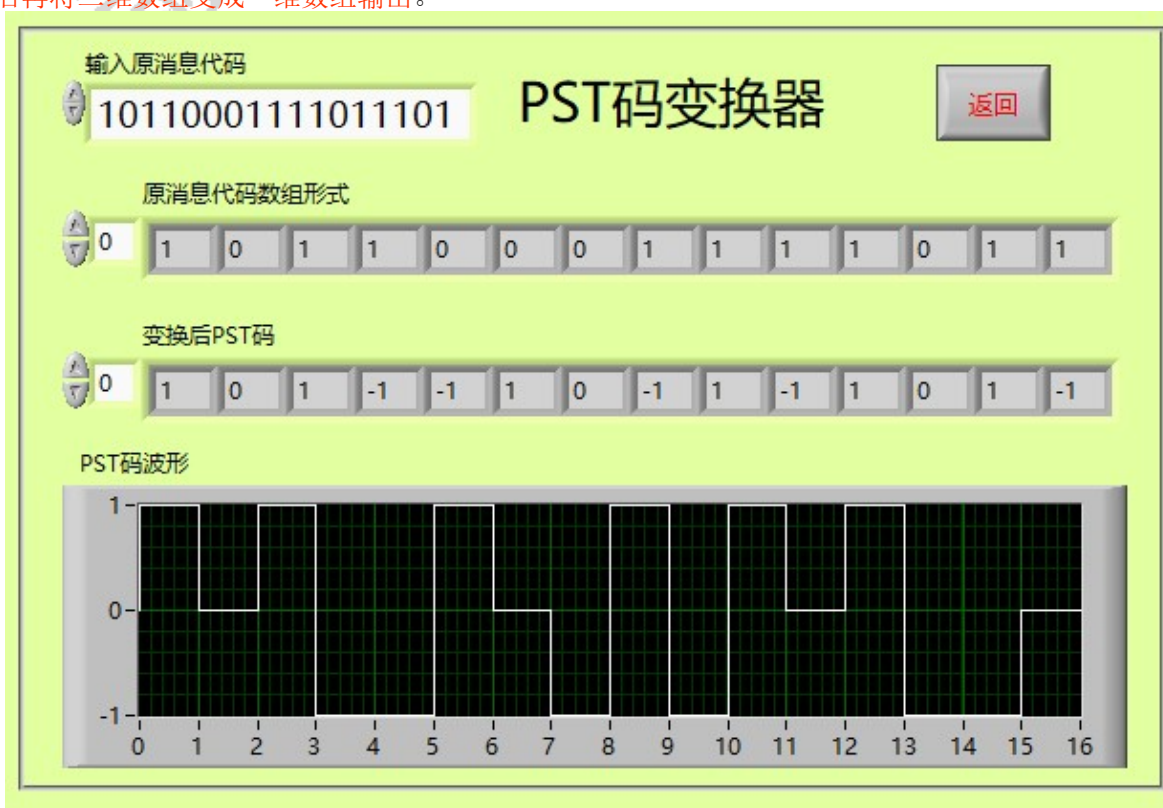
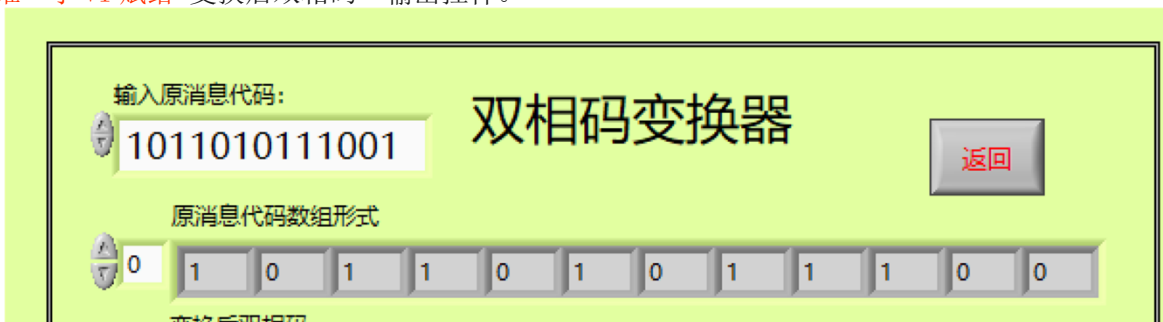


图 1.5 PST 码变换器前面板

1.5 双相码的仿真

双相码变换器的前面板如图 1.6 所示。在 For 语句中,由“二进制位数”来确定循环次数 N,每次从数组中提取一个元素来进行判断,当元素为“1”时,生成“10”两个码元;当元素为“0”时,生成“01”两个码元。再经“二维数组转一维”子 VI 赋给“变换后双相码”输出控件。



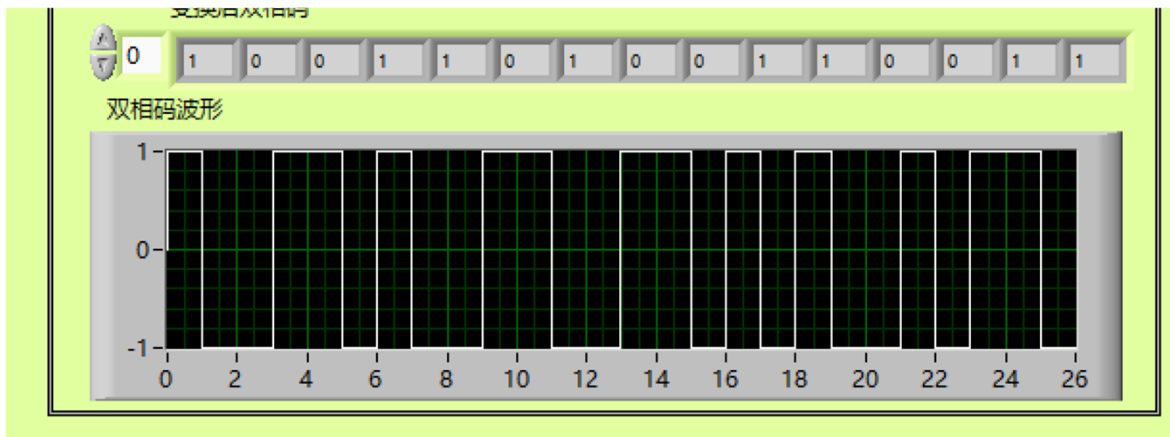


图 1.6 双相码变换器程序中:

- (1) 输出原代码数组形式;
- (2) 以二进制位数为次数进行循环;
- (3) 当为“1”时, 输出“10”; 当为“0”时, 输出“01”;
- (4) 这里所有数据都是“无符号二进制数”。

“1、0 码元”和“+1、-1 码元”均为一维数组, 一个是输入控件, 另一个是输出控件。利用 For 语句从数组中依次取出元素, 判断当是“0”时, 输出“-1”; 当是“1”时, 原样输出。这个子 VI 主要是单极性码变换成双极性码。

1.6 Miller 码的仿真

Miller 码变换器的前面板如图 1.7 所示。在 For 语句中有两个等于判断, 第一个用来对输入的每一个元素进行是否为“1”进行判断, 如果为真, 则要用“0 判 8 无 2 未数字”子 VI 来判断前一输出码元是“1”还是“0”, 当为“1”时, 输出“10”; 当为“0”时, 输出“01”。如果当前元素不为“1”时, 就要用到第二个判断, 来确定前一输入码元为何值, 为“0”时输出“11”; 为“1”时输出“00”。这里还有两个移位寄存器, 一个是保存前一输入码元, 另一个是保存前一输出码元的。

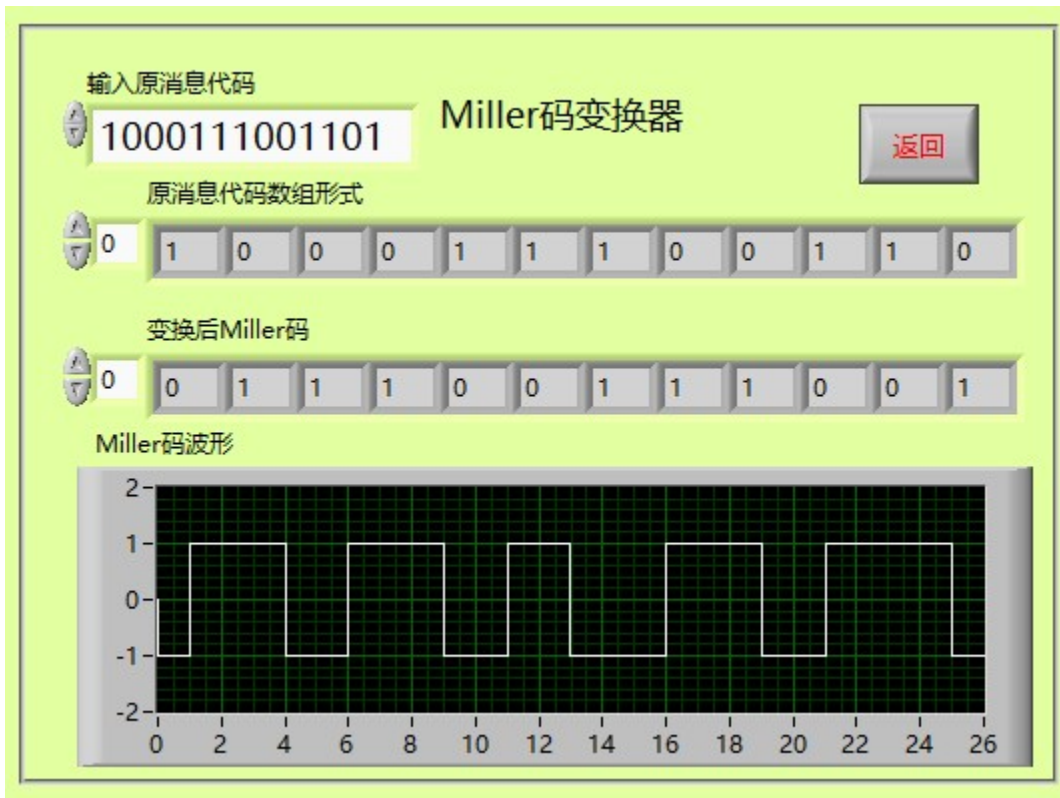


图 1.7 Mille 码变换器 “判 8 无 2 未数字.vi” 子程序编程思路为：“输入值”是一个有两个元素的 8 位无符号二进制数数组，程序中取出第二个元素判断是否为“1”，如果不等于 0, 即为真，灯亮。

1.7 CMI 码的仿真

CMI 码变换器前、后面板如图 1.8 所示，输入部分同其他变换器一样。在这个 For 语句中，循环还是由“二进制位数”确定，循环时每次取一个元素，并判断它是否为“1”，如果为假, 则输出“01”；为真时，则交替输出“11”和“00”，局部变量“计数”就是用来查看是否被 2 整除，来使“11”和“00”交替输出的。

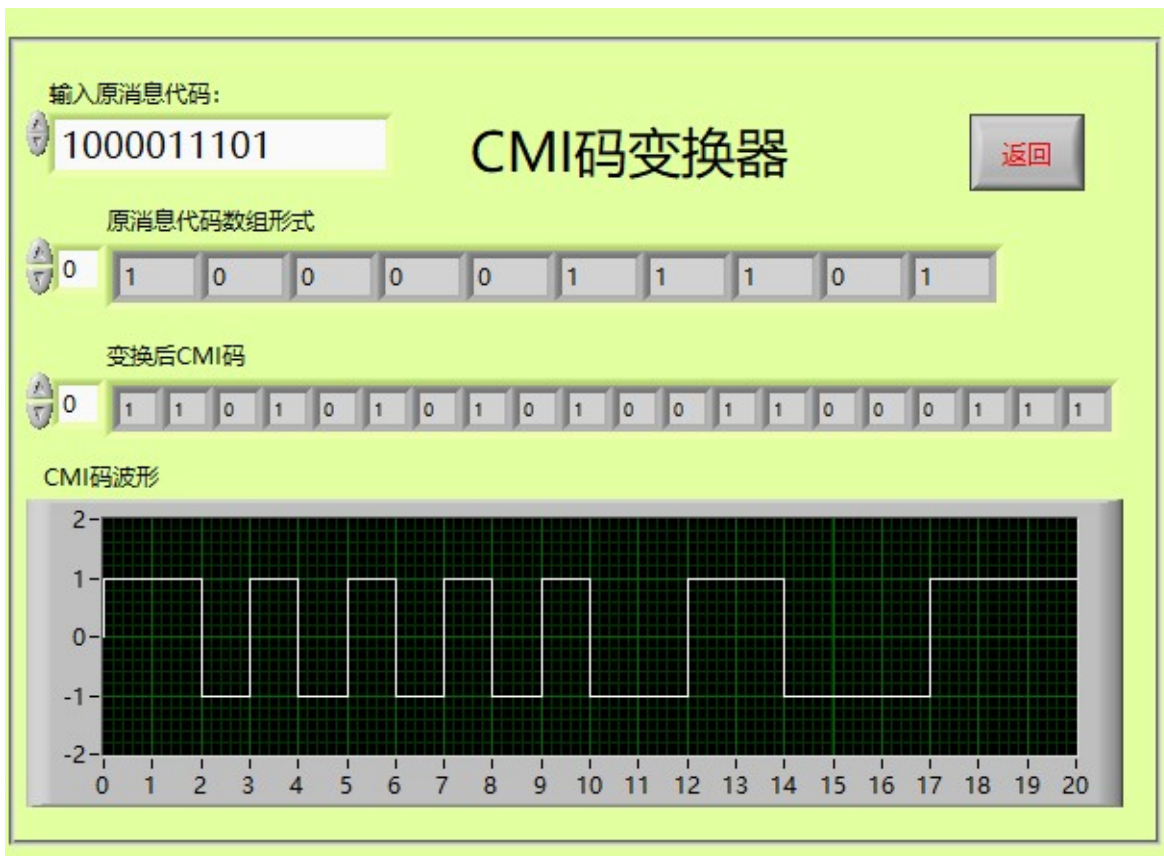


图 1.8 CMI 码变换器

2 奈奎斯特准则的仿真无码间干扰的基带传输系统应满足奈奎斯特准

$$\sum_i H\left(\omega + \frac{2\pi i}{T_s}\right), \quad |\omega| \leq \frac{\pi}{T_s} \quad (2.1)$$

当系统函数为升余弦函数时，表达式为

$$H(\omega) = \begin{cases} \frac{T_s}{2} \left(1 + \cos \frac{\omega T_s}{2}\right), & |\omega| \leq \frac{2\pi}{T_s} \\ 0, & |\omega| > \frac{2\pi}{T_s} \end{cases} \quad (2.2)$$

图 2.1 演示了当码元速率 $R = 1/T_s$ 时式 (4.2.2) 中 $H(\omega)$ 的奈奎斯特准则验证过程: 移位-相加-判断。 $H(\omega)$ 的验证程序设计思路是通过对比式 (2.1) 进行编译得出的，图中唯一的输入控件就是“T”，而其他的 5 个均为图形显示输出控件。在图 2.1(b) 中，For 循环的次数定为 101，当 i 为 50 时刚好设为中点，这样做是为了控制输出的采样点数。

“左→右”的 if 条件为 $\omega \leq -\pi/T_s \ \&\& \ \omega \geq -2\pi/T_s,$

“右→左”的 if 条件为 $\omega \leq 2\pi/T_s \ \&\& \ \omega \geq 2\pi/T_s,$