

一种基于 DSP 的新型纠错码的设计与实现

张京玲, 凌玉华, 廖力清

(中南大学信息与工程学院, 长沙 410083)

摘要: 在基于 DSP 的通信系统中, 由于纠错码的复杂性, 译码算法要占用 DSP 大量的时间和资源。针对此问题, 该文设计了一种新型的扭带纠错码——Tach 码。利用 DSP 移位指令, 通过左右移位进行编解码。介绍了扭带纠错码的编码和译码算法, 并与经典的 Hamming 码、BCH 码和 RS 码进行了纠错性能比较。仿真和分析表明扭带纠错码在与其他码性能相当的情况下, 译码简单, 不需要占用存储器去存储译码所需的错误图样表, 易于 DSP 实现。采用 TMS320vc5410 实现了该编解码器。

关键词: 信道编码; 纠错编解码; DSP

Design and Implementation of Novel Error-correcting Code Based on DSP

ZHANG Jing-ling, LING Yu-hua, LIAO Li-qing

(College of Information Science and Engineering, Central South University, Changsha 410083)

【Abstract】 Error-correcting code's decoding algorithm always takes up most time and memory of DSP, due to its complexity in the communication system. According to the problem, a novel tache code—Tach, as like tache counter, is presented, which is coded and decoded by right and left shifting, consequently need not store error charts, thus simplifies decoding algorithm and can be easily implemented based on DSP. The emulation and capability analysis, comparing with Hamming, BCH and RS codes, indicate that Tach code is an efficient and practicable error-correcting code based on DSP. The codec has been implemented using TMS320vc5410.

【Key words】 channel code; error-correcting coding and decoding; DSP

信道编码技术作为抗干扰技术的一种重要的手段, 在数字通信系统中显示出愈来愈重要的作用。人们不断提出新的编码来逼近香农极限, 如 Turbo 码、LDPC 码等。但是, 这些码的译码都非常复杂, 不便于工程上使用。就连 BCH 码、RS(Reed-Solomon)码的译码也要占用 DSP 大量的时间和资源。为了便于 DSP 的实现, 本文设计了一种新型的扭带纠错码——Tach 码(因其类似于扭带计数器, 故称之为扭带纠错码)。利用 DSP 移位指令, 通过循环的左右移位来进行编解码。该编码和译码都比较简单, 移位指令执行时间短, 节省 DSP 的时间; 同时不需要占用存储器去存储译码所需的错误图样表, 节约 DSP 资源, 便于 DSP 实现。

1 Tach 码编译系统

1.1 编码算法

利用 DSP 移位指令, 通过左右移位来进行编码。把信息码循环左移或右移得到的冗余码添加到信息码的后面, 进行编码。为了保证最小码距尽量大的原则, 采用信息码字中奇数个“1”右移编码, 而偶数个“1”左移编码, 并且信息码字中全 0 和全 1 的码字为违例码。具体编码算法如下:

算法定义

$p \ll$ 的符号代表循环左移, 且移出的高位补在最低位, P 代表移几位;

$p \gg$ 的符号代表循环右移, 且移出的低位补在最高位, P 代表移几位;

信息长度: $k \geq 3$

// k 为必须大于等于 3 的整数;

码字长度: $n = r \times k \quad 1 < r \leq k$

// r 为小于等于 k 的整数; r 称为节数

信息码多项式: $M(x)$

$$M(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \dots + mx + m_0$$

//除掉全 0 和全 1 的码字

编码后的多项式: $C(x)$

算法描述

(1) 计算 M 中“1”的个数, 为奇数转到(2), 偶数转到(3)

$$(2) C(x) = Mx^{(r-1)k} + 1 \ll Mx^{(r-2)k} + \dots + (r-1) \ll M$$

$$(3) C(x) = Mx^{(r-1)k} + 1 \gg Mx^{(r-2)k} + \dots + (r-1) \gg M$$

举例说明:

$M = [0001]$ 则 $k = 4$;

n 可以等于 $2 \times 4 = 8$ 或 $3 \times 4 = 12$ 或 $4 \times 4 = 16$

选 $n = 8$, 则为 Tach(8, 4)码

$C = [0001 \ 0010]$; C 为二节码;

选 $n = 16$, 则为 Tach(16, 4)码

$C = [0001 \ 0010 \ 0100 \ 1000]$;

C 为四节码。

1.2 译码算法

因 Tach 码编码的冗余码每一节都是信息码移位所得, 通过任何一节的冗余码都可译出信息码。在该码纠错能力的范围内, 总有一节码是正确的。本文的译码算法就是以接收到的码的每一节进行编码还原出原信息码, 与接收码比较, 选出码距最小的原信息码, 从而完成译码。具体译码算法如下:

算法定义

$p \ll$ 的符号代表循环左移, 且移出的高位补在最低位, P 代表移

作者简介: 张京玲(1978 -), 女, 硕士研究生, 主研方向: 无线通信和信道编码; 凌玉华、廖力清, 教授

收稿日期: 2006-09-13 **E-mail:** zhangjl_2005@126.com

几位；

$p \gg$ 的符号代表循环右移，且移出的低位补在最高位， P 代表移几位；

接收码多项式为： $R(x)$

译出后的多项式： $C(x)$

译出的信息码多项式： $M(x)$

信息长度： $k \geq 3$

// k 为必须大于等于 3 的整数；

码字长度： $n = r \times k \quad 1 < r \leq k$

// r 为小于等于 k 的整数； r 称为节数

所取的节数： $i \quad 1 < i \leq r$

// i 为小于等于 r 的整数；

算法描述

(1) 取 R 中第 i 节给 H 矩阵

(2) 计算 H 中“1”的个数，为奇数转到(3)，偶数转到(4)

(3) $C(x) = (i-1) \gg Hx^{(r-1)k} + (i-2) \gg Hx^{(r-2)k} + \dots$

$+ Hx^{(r-i)k} + 1 \ll Hx^{(r-1-i)k} + \dots + (r-i) \ll H$

(4) $C(x) = (i-1) \ll Hx^{(r-1)k} + (i-2) \ll Hx^{(r-2)k} + \dots$

$+ Hx^{(r-i)k} + 1 \gg Hx^{(r-1-i)k} + \dots + (r-i) \gg H$

(5) 计算 C 和 R 的码距 $d = \text{sum}(C \neq R)$

(6) 重复(1)~(5) r 次，直到 R 全部算完

(7) 选出最小码距 d_{\min} 所对应的原编码 C ， C 中的第 1 节为译出的信息码 M 。

举例说明：

$R = [001 \ 011]$ 则 $k = 3, r = 2$ ；

(1) 取 R 的第 1 节给 $H = [001]$

(2) $C = [001 \ 010 \ 100]$

(3) 计算 $d = 1$

(4) 取 R 的第 2 节给 $H = [011]$

(5) $C = [110 \ 011]$

(6) 计算 $d = 3$

(7) 译出 $M = [001]$

1.3 理论性能分析

1.3.1 Tach 码的归属

Tach(n, k) 码是把信息划成 k 个码元位一段(称为信息组)，通过编码器变成为 n 个码元的一组编码方式，所以属于分组码。但是，Tach 码不满足线性叠加原理，所以不属于线性码，为非线性码。

1.3.2 Tach 码的最小码距

当 Tach(n, k) 码中 n 等于 k 乘以 k 时，称此时的 Tach 码为完备码。下面分析 Tach 完备码的最小码距。

Tach 完备码可以根据信息码中“1”的个数分成 $k-1$ 组。每一组中都像扭带计数器的结构一样构成编码。由于每一组扭带内编码都是通过信息码左移或者右移实现的，因此每一组内的码距都为 $2 \times k$ 。

例如： $k = 3$

$[001 \ 010 \ 100]$ 与 $[010 \ 100 \ 001]$ 的码距 $d = 2 \times 3 = 6$

奇数个“1”的组之间的码距也一定大于等于 $2 \times k$ 。

例如： $k = 4$

$[0001 \ 0010 \ 0100 \ 1000]$

与 $[0111 \ 1110 \ 1101 \ 1011]$ 的码距 $d = 2 \times 4 = 8$

同理，偶数个“1”的组之间的码距也一定大于等于 $2 \times k$ 。

现在只剩下相邻的奇数个“1”和偶数个“1”的组。由于编码的时候采用奇数个“1”左移和偶数个“1”右移的方法，因此避免了奇数个“1”和偶数个“1”的组的码距为 k 的情况。奇数个“1”组中每一个的码字与偶数个“1”的码字的码距都为

$$d = 1 + 1 + 3 \times (k - 2)$$

例如： $k = 3$

$[001 \ 010 \ 100]$ 与 $[011 \ 101 \ 110]$ 的码距 $d = 1 + 1 + 3 = 5$

综上所述，Tach 完备码的最小码距 d_{\min} 为

$$d_{\min} = \begin{cases} 1 + 1 + 3 \times (k - 2) & k = 3 \\ 2 \times k & k \geq 4 \end{cases}$$

1.3.3 Tach 码的检错和纠错能力

根据上述最小码距的式子，可以得到 Tach 完备码能检出 $d_{\min} - 1$ 个比特的错误，能纠正 $k - 1$ 个比特的错误。这样在纠错能力范围内，总能保证接收码中有一节码是正确的。译码算法也正是利用了这一点。并且，由于全“0”和全“1”为违例码，接收码中任意一节出现全“0”或者全“1”就知道是错误的，因此不予处理。因为如何一节码中都含有信息码的全部信息，所以从原则上说只要一节码正确，就可以得到原信息码。正是因为这些特性，该码抗突发噪声比 Hamming 码要强，后面的仿真分析也证实了这一点。

如果 Tach (n, k) 码中 $n = r \times k$ ， r 为小于 k 的整数，称此时的 Tach 码为缩短码。Tach 缩短码在 $k \geq 4$ 的条件下能纠正 $r - 1$ 个比特的错误。缩短码的纠错性能比完备码差，但是它更灵活，码利用率更高，更有实际的利用价值。所以仿真分析中，都是使用 Tach 的缩短码与各种码进行比较。

2 仿真分析

所做的仿真是在不同的信噪比 E_b/N_0 下进行的蒙特卡洛仿真。仿真比较了 Tach 码与 Hamming 码、BCH 码和 RS 码在相同的信噪比下误码率的大小。仿真系统是先产生均匀随机数，用各种编码算法编码，再进行 BPSK 调制，加入高斯噪声或突发噪声后，进行 BPSK 解调和解码，得到的解码与原产生的随机数比较，计算误比特率，完成仿真过程。下面是两种信道下的仿真结果。

2.1 加性高斯白噪声信道仿真结果

在 AWGN 信道条件下，在不同的信噪比 E_b/N_0 下发送 10000 帧的二进制数，先进行编码，后进行 BPSK 调制，采用硬判决方法。主要选择了 3 对码进行比较。它们是 Tach(16, 8) 与 Hamming(15, 11) 比较；Tach(32, 16) 与 BCH(32, 21) 比较；Tach(64, 16) 与 RS(63, 32) 比较。得到的仿真结果如图 1 所示。

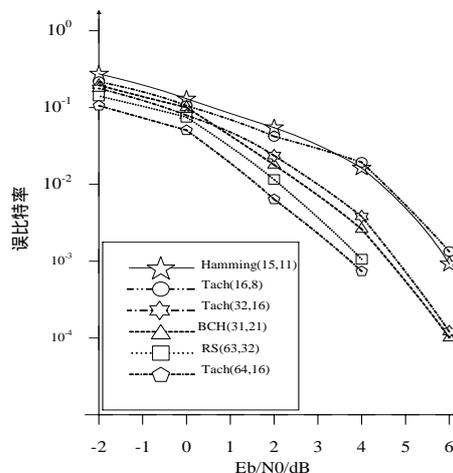


图 1 在 AWGN 信道下 Tach 码与 Hamming、BCH 和 RS 码比较

从仿真结果可以看出，在加性高斯白噪声信道下，在相同的信噪比和相同的纠错能力长度下，Tach 码的误码率与 Hamming、BCH 和 RS 码的相当。

2.2 突发噪声信道仿真结果

在突发噪声信道条件下,在不同的信噪比 E_b/N_0 下发送10 000帧的二进制数,先进行编码,后进行BPSK调制,采用硬判决方法。选择了Tach(16, 8)与Hamming(15, 11)比较;Tach(32, 16)与BCH(32, 21)比较;Tach(64, 16)与RS(63, 32)比较。得到的仿真结果如图2所示。

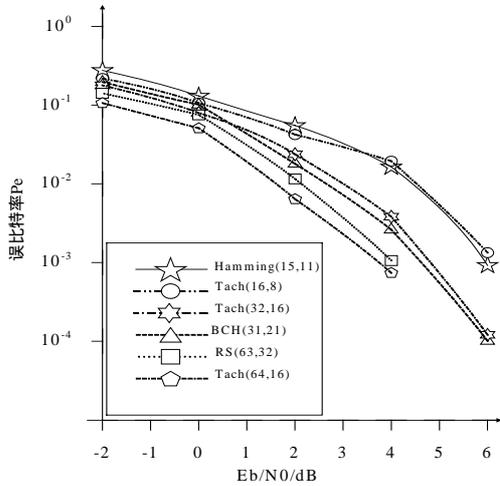


图2 在突发噪声信道下 Tach 码与 Hamming、BCH 和 RS 码比较

从仿真结果可以看出,在突发噪声信道下,在相同的信噪比和相同纠错能力长度下 Tach 码的误码率稍小于 Hamming 码,与 RS 码相当。正好和理论分析的一致。

3 DSP 实现与流程

为了便于DSP实现,一般选择信息码的长度为 2^n 个。例如信息码长度 k 为8、16或32等。笔者选择TMS320vc5410实现了该编译码系统。

3.1 DSP 编译码系统

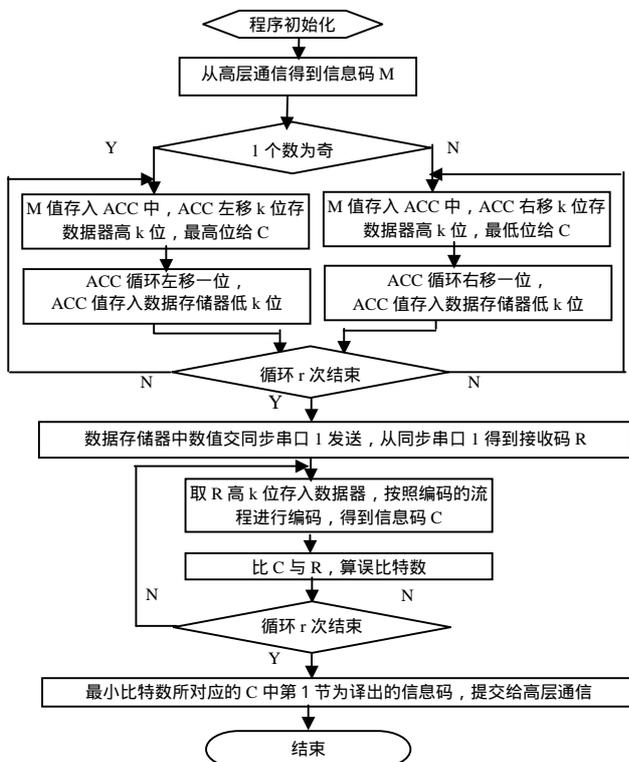


图3 Tach 编译码的程序流程

具体的编译码的 DSP 流程如图3所示。TMS320vc5410

有桶形移位寄存器,能把输入的数据进行0b~31b的左移和右移,只需要一个机器周期。使用ROL指令进行左移1位,使用ROR指令进行右移1位,所用的指令所需要的机器周期都是1。其外按时钟为12MHz,内部对其进行8倍频,则CPU时钟为96MHz,即一个机器周期为 $1/96\text{MHz}=10.42\text{ns}$ 。

同时,5410有3个同步串口和4个外部中断。同步串口1与低层通信,同步串口2和3用来与高层通信。外部中断1提供同步定时信息。主程序分为发送、接收和编码、译码4大模块。同步串口1产生中断后,首先判断是发送还是接收。如是发送,就把从高层通信得到的码进行编码后,发送出去。如是接收,就把接收的码进行译码后,传输给高层通信。

3.2 DSP 实现的部分源程序

主程序采取模块化的设计,分为发送、接收和编码、译码4大模块。程序在CCS2000的开发环境下进行编译,使用汇编语言编写。

下面给出Tach(64,32)编码和译码模块主要的源程序。

编码模块的主程序如下:

```

DLD #0400h, A
DST A, #0800h
BANZ LOOP *AR0
ROL A
DLD #0400h, A
ROL A
LOOP: ROR A
DLD #0400h, A
ROR A
DST A, #0801h
    
```

说明:从高层得到的信息码存在数据存储器0400H中,编好的码存入数据存储器0800H、0801H中;AR0为0则为信息码中奇数个“1”,否则为偶数个“1”。

译码模块的主程序如下:

```

DLD #0900h, A
ROL A
DLD #0900h, A
ROL A
XOR #0901h, A
LD #31, B
Xm1: SUB #1, B
ROL A
BCND JIAN1 NC
ADDM #1, T1
JIAN1: BANZ Xm1 B
LAR AR1, T1
CMPR AR7
BCND MI1 NTC
LAR AR7, T1
LAR AR6, A
MI1: DLD #0901h, A
ROR A
DLD #0901h, A
ROR A
XOR #0900h, A
LD #31, B
Xm2: SUB #1, B
ROL A
    
```

(下转第136页)