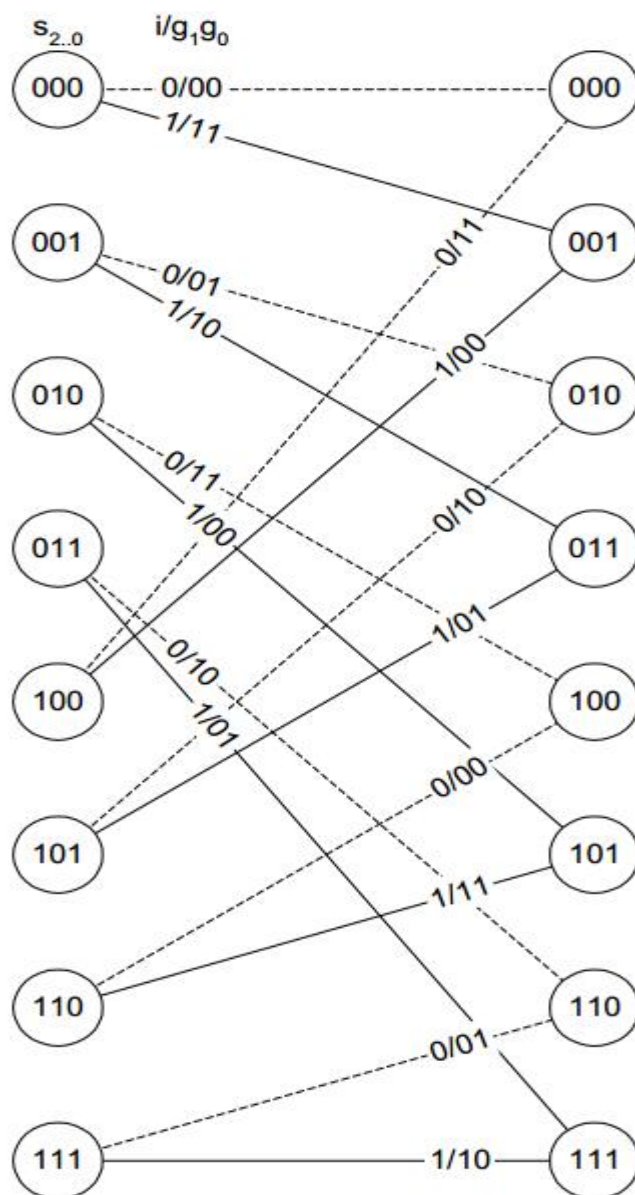


```

/*****
*
* Copyright 2014      Jr
* All right reserved
*
* 当前版本: V1.0
* 作    者: F06553
*
* 源代码说明: CC1101 所使用的 FEC 源码自述文件, 背景知识和源码解释。
*              后续参阅新的信息或错误后会更新此文档
*****/

```

阅读本文前请先行阅读(2,1,4)卷积编码{别名(4,1/2)卷积编码}的相关内容, 了解其硬件实现, 以及树状图(Trellis diagram)又称篱笆图的表述方式。



背景知识

FEC（前向纠错编码）常见的编码方式有：BCH 码、和卷积码 CC1101 使用的是 (2,1,4) 卷积码，A71XX 使用的是 RS 码(BCH 码一种)。

卷积译码分两类：

1.代数解码

(1)大数逻辑解码(又称门限解码)，代数解码的最主要一种方法。

2.概率解码(最大似然解码)

(1)Viterbi,算法复杂度随约束长度增加而呈指数增加，适用于约束度较小($N \leq 10$)卷积编码

(2)序贯解码,约束度较大使用此算法

Viterbi Algorithm 是高通创始人之一的 Viterbi 在 1967 提出的，维特比译码是一种最大似然卷积码译码。在维特比译码器中，卷积码编码的接收信号通常被噪声损坏，将可能的估计序列与接收到序列的汉明距离或欧氏距离度量进行比较，在树状图或网格图中选择一条最小路径的状态，从该状态中估计出输入最可能的序列编码。

网上 Viterbi 算法实现上大多数都是 FPGA 实现，用软件实现的较少。

关键的参数： (n,k,N)

k: 输入码元

n: 输出码元

m: 编码器中需要的寄存器个数(S2S1S0)

+1: i, i 参与到 output 的输出运算，故+1

N: m+1 编码过程中互相约束的码段个数

nN: $nN = (m+1)n$, 编码过程中互相约束的码元个数

1.码率: k/n

2.约束度: N

3.约束长度: nN

此源码中参数为(2,1,4)

Viterbi 译码过程主要有以下步骤：

1. 加比选

2. 回溯

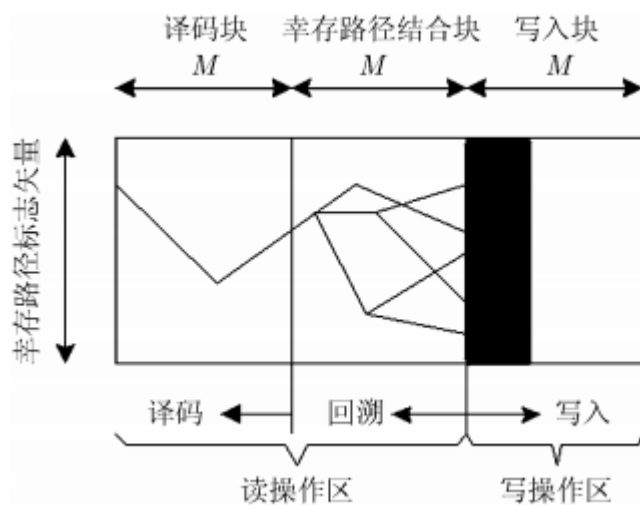
加：当前状态的汉明距离与上一状态的累积汉明距离相加，上一状态有两条路径到达当前路径。

比：比较当前状态的两条路径的累积汉明距离。

选：比的结果中选择一条累积汉明距离小的路径作为幸存路径。

回溯：利用 Viterbi 合并的特性，进行译码。根据 Viterbi 译码算法，随着译码的进行，经过一定的时间后，各个状态的幸存路径将开始结合，并最终结合于一条幸存路径。并根据这个特性，只要经过足够长的时间，根据任一状态向前回溯 M 时刻，会回到相同的状态点，M 为回溯深度，M 回溯深度的大小通常选定为约束长度 K 的 5~10 倍。（假设没有错码，最终幸存路径累积汉明距离为 0，此路径在加比选的过程会保存下来，而其他路径会被丢弃，经过多次加比选的过程，此路径经过多次排挤其他路径，会最终保存下来。故在当前时刻状态向前回溯 M 时刻，比回溯 M 时刻更早时刻是单一的路径），下图为 Viterbi 合并特性直观

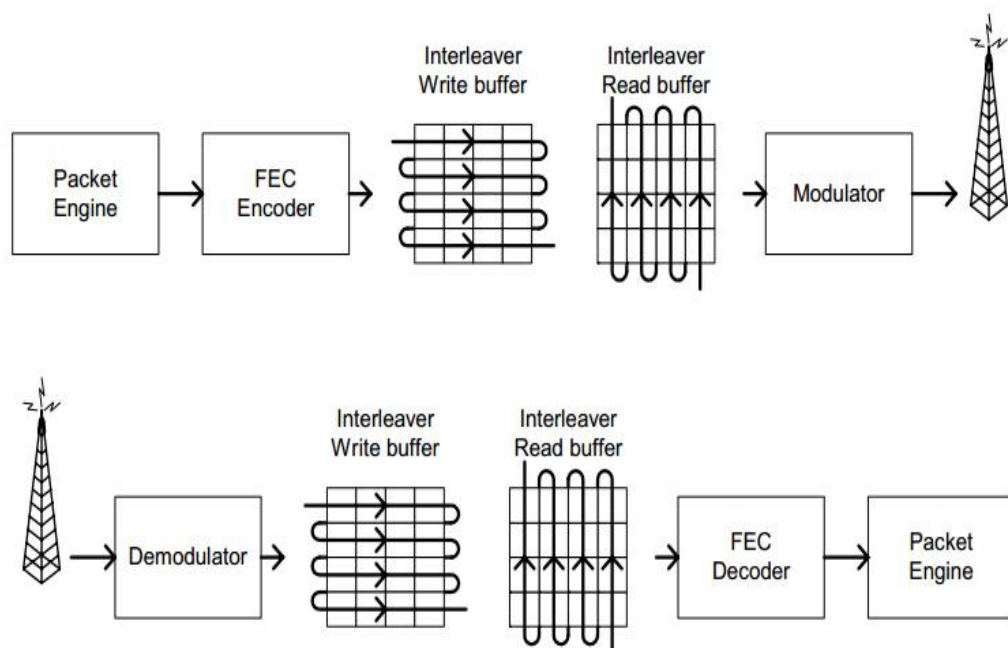
的表示:



尽管大多数无线收发芯片具有设置高灵敏度、GFSK(Gauss frequency shift keying)调制等功能,但突发的无线干扰还是会使数据发生一连串出错。卷积码的约束长度又限制了允许连续出错的信息位数,若连续出错的信息位数超出了约束长度,卷积译码将无法实现纠错。所以,在数据经卷积编码后还要进行交织,将连续出错的信息位在解交织时分解为分散的、不连续的信息位,以便卷积译码在约束长度内纠正连续错码。

交织分两类:

- 1.块交织,块交织适用于每次传输固定长度的数据帧或数据量较大的场合。
此源码 4*4 块交织,每个单元 2bit
- 2.卷积交织,卷积交织适用于每次传输长度不固定的数据帧,用于 Turbo 码



SecureCRT

Input: [62 bytes]

0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D

Appended CRC: [64 bytes]

0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x61 0x4D

Appended Trellis terminator: [66 bytes]

0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x61 0x4D
0x0B 0x0B

FEC encoder output: [132 bytes]

0x00 0x00 0x00 0x03 0x7C 0x0D 0xF0 0x0E 0x8C 0x37 0xC0 0x34 0xBC 0x3A 0x30 0x39
0x4C 0xDF 0x00 0xDC 0x7C 0xD2 0xF0 0xD1 0x8C 0xE8 0xC0 0xEB 0xBC 0xE5 0x30 0xE6
0x4F 0x7C 0x03 0x7F 0x7F 0x71 0xF3 0x72 0x8F 0x4B 0xC3 0x48 0xBF 0x46 0x33 0x45
0x4F 0xA3 0x03 0xA0 0x7F 0xAE 0xF3 0xAD 0x8F 0x94 0xC3 0x97 0xBF 0x99 0x33 0x9A
0x41 0xF0 0x0D 0xF3 0x71 0xFD 0xFD 0xFE 0x81 0xC7 0xCD 0xC4 0xB1 0xCA 0x3D 0xC9
0x41 0x2F 0x0D 0x2C 0x71 0x22 0xFD 0x21 0x81 0x18 0xCD 0x1B 0xB1 0x15 0x3D 0x16
0x42 0x8C 0x0E 0x8F 0x72 0x81 0xFE 0x82 0x82 0xBB 0xCE 0xB8 0xB2 0xB6 0x3E 0xB5
0x42 0x53 0x0E 0x50 0x72 0x5E 0xFE 0x5D 0x82 0x64 0xCE 0x67 0x86 0x33 0x4B 0x2B
0xBC 0xD1 0x8C 0xD1

TX Interleaver output: [132 bytes]

0xC0 0x00 0x00 0x00 0x84 0xCF 0x33 0x31 0x0C 0x47 0xCC 0x32 0x48 0x8B 0xFF 0x02
0x0C 0xCF 0x44 0xCD 0x48 0x03 0x77 0xFD 0xC0 0x8B 0x88 0xFE 0x84 0x47 0xBB 0xCE
0xF3 0xCF 0xCC 0x45 0xB7 0x03 0xFF 0x75 0x3F 0x8B 0x00 0x76 0x7B 0x47 0x33 0x46
0x3F 0x03 0x88 0x89 0x7B 0xCF 0xBB 0xB9 0xF3 0x47 0x44 0xBA 0xB7 0x8B 0x77 0x8A
0xD1 0x30 0xCC 0xCD 0x95 0xFC 0xFF 0xFD 0x1D 0x74 0x00 0xFE 0x59 0xB8 0x33 0xCE
0x1D 0xFC 0x88 0x01 0x59 0x30 0xBB 0x31 0xD1 0xB8 0x44 0x32 0x95 0x74 0x77 0x02
0xE2 0xFC 0x00 0x89 0xA6 0x30 0x33 0xB9 0x2E 0xB8 0xCC 0xBA 0x6A 0x74 0xFF 0x8A
0x2E 0x30 0x44 0x45 0x6A 0xFC 0x77 0x75 0xE2 0x74 0x88 0x76 0xFE 0xA1 0x8C 0x12
0x44 0x33 0x47 0xEE

FEC dec

0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x61 0x4D
0x0B 0x08

encoding time : 1074us

decoding time : 53555us