



中华人民共和国国家标准

GB/T 39348—2020

空间数据与信息传输系统 遥测同步与信道编码

Space data and information transfer systems—
TM (telemetry) synchronization and channel coding

(ISO 22641:2012, MOD)

2020-11-19 发布

2021-06-01 实施

国家市场监督管理总局 发布
国家标准化管理委员会

中 华 人 民 共 和 国
国 家 标 准
空间数据与信息传输系统
遥测同步与信道编码
GB/T 39348—2020

*

中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)

网址: www.spc.org.cn

服务热线: 400-168-0010

2020年11月第一版

*

书号: 155066 • 1-65971

版权专有 侵权必究

目 次

前言	V
1 范围	1
2 规范性引用文件	1
3 术语和定义、缩略语	1
3.1 术语和定义	1
3.2 缩略语	2
4 总则	3
4.1 体系结构	3
4.2 功能介绍	4
4.3 子层的内部组织	4
5 卷积码	6
5.1 概述	6
5.2 一般要求	6
5.3 基本卷积码	7
5.4 删余卷积码	8
6 RS 码(Reed-Solomon 码)	9
6.1 概述	9
6.2 一般要求	9
6.3 RS 码的特性	9
6.4 符号交织	13
7 级联码	13
7.1 级联的组合	13
7.2 内码和外码	13
7.3 级联信道编码系统框图	13
8 Turbo 码	14
8.1 概述	14
8.2 一般要求	14
8.3 标准 Turbo 码的特性	14
8.4 同步标识	18
9 LDPC 码传送帧	18
9.1 概述	18
9.2 一般要求	19
9.3 编码码率为 223/255 的 LDPC 码	19
9.4 编码码率为 1/2, 2/3 和 4/5 的 LDPC 码簇	22
9.5 同步标识	25

10 帧同步	25
10.1 概述	25
10.2 附加同步标识(ASM)	26
10.3 ASM 的比特图样	26
10.4 ASM 的位置	28
10.5 ASM 与 RS、Turbo 及 LDPC 码块和码字的关系	28
10.6 嵌入数据流中的 ASM	28
11 伪随机化	28
11.1 概述	28
11.2 伪随机化说明	28
11.3 伪随机化的同步与应用	29
11.4 伪随机序列	29
12 传送帧长度	29
13 管理参数	30
13.1 概述	30
13.2 编码选择	30
13.3 参数设置	31
附录 A (资料性附录) 本标准与 ISO 22641:2012 相比的结构变化情况	32
附录 B (资料性附录) 本标准与 ISO 22641:2012 的技术性差异及其原因	34
附录 C (规范性附录) 服务定义	36
附录 D (资料性附录) RS 码生成多项式系数的展开式	38
附录 E (资料性附录) Berlekamp 表示和常规表示间的变换	39
附录 F (资料性附录) 223/255 比率的 LDPC 码	45
附录 G (资料性附录) 循环冗余(CRC)检错码应用	47
图 1 TM 同步与信道编码和 OSI 层之间的关系	3
图 2 发送端同步和编码子层的内部流程	5
图 3 接收端同步和编码子层的内部流程	6
图 4 卷积码器原理图	7
图 5 删余编码器原理图	8
图 6 RS 码块结构	11
图 7 RS 交织编码框图	13
图 8 级联信道编码系统原理框图	14
图 9 Turbo 码器方框图	16
图 10 不同编码码率下的 Turbo 码字	16
图 11 数据排列说明	17
图 12 带有附加同步标识的 Turbo 码字	18
图 13 (8176,7156)码的校验矩阵	19
图 14 (8176,7154)码的生成矩阵	21
图 15 (8160,7136)码字示意图	22
图 16 无编码、卷积码、RS 码、级联码和 7/8 比率 LDPC 码的 ASM 比特码型	27

图 17	1/2 比率 Turbo 码和 1/2,2/3,4/5 比率 LDPC 码(应用到传送帧)的 ASM 比特码型	27
图 18	1/3 比率 Turbo 码的 ASM 比特码型	27
图 19	1/4 比率 Turbo 码的 ASM 比特码型	27
图 20	1/6 比率 Turbo 码的 ASM 比特码型	28
图 21	伪随机发生器配置	29
图 22	伪随机化逻辑框图	29
图 E.1	常规编码器与 Berlekamp 编码器间的等效变换关系	39
图 G.1	CRC 编码器简图	48
图 G.2	CRC 译码器简图	48
表 1	不同编码码率的删余卷积码的输出序列	8
表 2	不同编码码率的信息块长度及码字长度	15
表 3	不同信息块长度参数 k_1 和 k_2	17
表 4	循环矩阵规定	20
表 5	不同编码码率对应的码字长度	22
表 6	不同编码码率对应子矩阵大小 M	22
表 7	$\phi_K(0,M)$ 和 $\phi_K(1,M)$ 的定义	23
表 8	$\phi_K(2,M)$ 和 $\phi_K(3,M)$ 的定义	24
表 9	RS 码的管理参数表	26
表 10	卷积码的管理参数表	31
表 11	RS 码的管理参数表	31
表 12	Turbo 码的管理参数表	31
表 13	LDPC 码的管理参数表	31
表 14	帧同步的管理参数表	31
表 A.1	本标准与 ISO 22641:2012 的章条编号对照情况	32
表 B.1	本标准与 ISO 22641:2012 的技术性差异及其原因	34
表 D.1	$E=16$ 的 RS 码的生成多项式系数	38
表 D.2	$E=8$ 的 RS 码的生成多项式系数	38
表 E.1	常规基和对偶基符号对照表	41
表 F.1	生成矩阵的循环矩阵表	45

前 言

本标准按照 GB/T 1.1—2009 给出的规则起草。

本标准使用重新起草法修改采用 ISO 22641:2012《空间数据与信息传输系统 遥测同步与信道编码》。

本标准与 ISO 22641:2012 相比在结构上有较多调整,附录 A 中列出了本标准与 ISO 22641:2012 的章条编号对照一览表。

本标准与 ISO 22641:2012 相比存在技术性差异,这些差异涉及的条款已通过在其外侧页边空白位置的垂直单线(|)进行了标示,附录 B 中给出了相应技术性差异及其原因的一览表。

本标准还做了下列编辑性修改:

- 删除了 ISO 22641:2012 中第 1 章关于原理定义文档结构参考文献等部分内容;
- 将 ISO 22641:2012 中附录 A“缩略语”改为了正文第 4 章;
- 删除了 ISO 22641:2012 的资料性附录 B“安全、空间分配编号和专利因素”;
- 删除了 ISO 22641:2012 的资料性附录 E“参考文献”。

本标准由全国宇航技术及其应用标准化技术委员会(SAC/TC 425)提出并归口。

本标准起草单位:中国航天标准化研究所、北京空间飞行器总体设计部、北京航空航天大学、北京遥测技术研究所、西安空间无线电技术研究所、航天东方红卫星有限公司。

本标准主要起草人:周玉霞、徐勇、刘荣科、李炯卉、谢天娇、丁兴文、韩孟飞、刘栋梁、康登榜、王维嘉、许冬彦。

空间数据与信息传输系统 遥测同步与信道编码

1 范围

本标准规定了空间数据与信息传输系统中遥测同步与信道编码涉及的卷积码、RS 码、Turbo 码、LDPC 码、交织、伪随机化技术要求。

本标准适用于空-地链路以及空-空链路之间的信道编码的接口设计。可与 GB/T 39351 或 GB/T 39345 配合使用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 2900.54 电工术语 无线电通信:发射机、接收机、网络和运行(GB/T 2900.54—2002, IEC 60050-713:1998, IDT)

GB/T 39345 空间数据与信息传输系统 高级在轨系统空间数据链路协议(GB/T 39345—2020, ISO 22666:2016, MOD)

GB/T 39349 空间数据与信息传输系统 遥控同步与信道编码(GB/T 39349—2020, ISO 22642:2015, MOD)

GB/T 39351 空间数据与信息传输系统 遥测空间数据链路协议(GB/T 39351—2020, ISO 22645:2016, MOD)

3 术语和定义、缩略语

3.1 术语和定义

GB/T 2900.54 界定的以及下列术语和定义适用于本文件。

3.1.1

符号 symbol

进行变换和处理的基本数据单位,其具体含义视应用场合而异:

- a) 对标准卷积码,一个符号是指一个二元码位;
- b) 对 RS 码,其符号是指有限域 $GF(2^J)$ 中的元素,这时每个符号由 J 个比特位组成。

3.1.2

码字 codeword

一个 (n, k) 码的编码器对 k 个输入信息符号编码所产生的由 n 个符号组成的一个序列。

注:这个输出序列在译码时将作为一个整体来处理。

3.1.3

码块 codeblock

用于 RS 码, I 个码字的集合体,其中 I 是交织深度。在 I 为 1 是“码块”和“码字”等价。

3.1.4

编码码率 code rate

编码器输入信息符号的平均数与相应输出符号平均数之比。

3.1.5

约束长度 constraint length

在卷积码编码时,为确定任一个输出符号的值所需要的连续的输入信息比特位数。

3.1.6

生成多项式 generator polynomial

生成函数

编码器对输入进行运算,产生输出码字所采用的变换关系。

3.1.7

软判决 soft decision

软量化

对接收的二元符号的幅度,根据预置的公层值进行多分层量化处理的过程。

3.1.8

系统码 systematic code

编码器的输入信息序列不加改变的直接作为输出码字的一部分。

3.1.9

透明码 transparent code

如果一个码的编码器对两个相位互反的 PCM 序列编码后,其输出也为两个相位互反的 PCM 序列,则称此码为透明码,否则称为非透明码。

3.1.10

删余码 punctured code

通过删除传输前卷积码器所产生的校验位获得的码组。

注:删余码使带宽效率获得了提升。

3.1.11

外码 outer code

在级联编码的编码系统中,对数据流进行的第一步的编码。

3.1.12

内码 inner code

在级联编码的编码系统中,对数据流进行的最后一步的编码。

3.1.13

虚拟填充 virtual fill

在一个系统分组码 (n, k) 中,编码器输出的码字中包含有 k 个符号的信息部分和 $(n - k)$ 个符号的校验部分。当实际的信息符号数目 s 小于 k 时,为使编、译码器正常工作,在编码器和译码器中需预置具有 $(k - s)$ 个符号的常量字符串,附加到信息部分。

注:这 $(k - s)$ 个符号并不在信道中发送,因而称为“虚拟填充”。

3.1.14

循环矩阵 circulant

在 LDPC 码中,二进制方阵,每一行由前一行循环右移 1 位得到,最后一行由最后一行循环右移 1 位得到。

3.2 缩略语

下列缩略语适用于本文件。

- AOS——高级在轨系统(Advanced Orbiting Systems)
- ASM——附加同步标识(Attached Sync Marker)
- BPSK——二相频移键控(Binary Phase Shift Keying)
- CADU——信道访问数据单元(Channel Access Data Unit)
- CCSDS——空间数据系统咨询委员会(Consultative Committee for Space Data System)
- CRC——循环冗余校验(Cyclic Redundancy Check)
- CSM——码同步标识(Code Sync Marker)
- FECF——帧差错控制域(Frame Error Control Field)
- LDPC——低密度奇偶校验码(Low Density Parity Code)
- NRZ-L——非归零电平码(Non-Return-to-Zero-Level)
- NRZ-M——非归零传号码(Non-Return-to-Zero-Mark)
- OSI——开放系统互连(Open Systems Interconnection)
- PCM——脉冲编码调制(Pulse Code Modulation)
- QPSK——四相频移键控(Quadrature Phase Shift Keying)
- RS——瑞得-所罗门(Reed-Solomon)
- SMTF——同步标识传送帧(Sync-Marked Transfer Frame)

4 总则

4.1 体系结构

图 1 描述了本标准与开放系统互连参考模型(OSI)之间的关系。CCSDS 空间数据链路协议把数据链路层分为同步与信道编码子层和数据链路协议子层。

TM 和 AOS 空间数据链路协议位于数据链路协议子层,采用被称为传送帧的协议数据单元为数据提供传输功能。遥测同步和信道编码对固定长度的传送帧提供以恒定的帧速率在空间链路的物理信道上提供单向的传输服务,具体服务说明见附录 C。

同步和信道编码子层为传送帧在空间链路上的传送提供了必要的辅助功能,包括差错控制编/译码、传送帧定界/同步、和伪随机化的生成与去除。

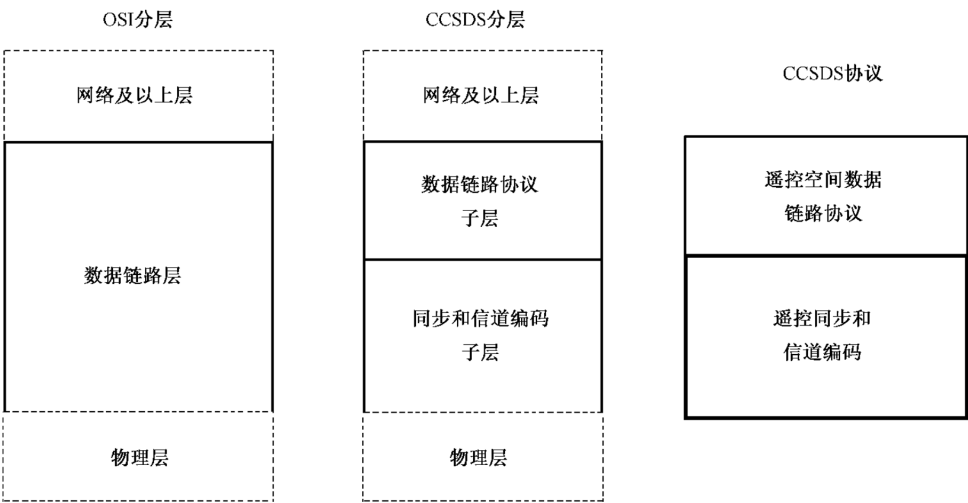


图 1 TM 同步与信道编码和 OSI 层之间的关系

4.2 功能介绍

4.2.1 功能

遥测同步与信道编码提供了如下功能：

- a) 差错控制编码,含帧验证;
- b) 同步;
- c) 伪随机化。

4.2.2 差错控制编码

本标准使用四种差错控制码,分别是:

- a) 卷积码:规定卷积码的编码码率为: $r=1/2$;约束长度为: $k=7$;
- b) RS(Reed-Solomon)码:规定(255,223)RS码和(255,239)标准RS码;
- c) Turbo码:编码码率可选: $r=1/2,1/3,1/4,1/6$;
- d) LDPC码:当选择编码码率223/255的LDPC码时,传送帧长度为1 024字节;当选择编码码率1/2、2/3或4/5的LDPC码时,允许的传送帧长度为1 024 bit、4 096 bit、16 384 bit。

4.2.3 帧校验

译码完成后,接收端的上层需要知道每个译码的传送帧是否能作为有效的数据单元使用,即需要接收某个指示来表示接收帧的质量。此功能称为帧校验。

RS和LDPC译码器能以很高的概率确认它们是否能正确地对传送帧译码。因此,RS码和LDPC码可用于帧有效性确认。当不采用RS码或LDPC码时,GB/T 39351或GB/T 39345定义的帧差错控制域用于帧验证。

4.2.4 同步

本标准规定了采用附加同步标识(ASM,见第8章)进行传送帧同步的方法。如果数据模糊(‘0’和‘1’)没有能够用物理层的调制方法解决,ASM也可用于解数据模糊。

本标准规定了一种码同步标识(CSM,见第8章)也用来解数据模糊,CSM不用于传送帧同步。

4.2.5 伪随机化

本标准规定了一种伪随机化发生器,以便改善遥测空间链路某些性能,如:辅助接收捕获、比特同步、卷积码同步以及适用的帧验证。

4.3 子层的内部组织

4.3.1 发送端

图2表示了发送端同步和编码子层的内部流程。该图指出了该层的功能,表示了这些功能之间的逻辑关系。该图并不表示实际系统中硬件和软件的配置。可选模块取决于实际任务,并不是所有的功能都出现在子层中。

在发送端,同步和信道编码子层接收来自数据链路协议子层的具有固定长度的传送帧,根据任务选择执行功能,将连续的、不间断的信道符号流传递到物理层。当使用SMTF流的LDPC码时,发送端的码块需要伪随机化和增加附加帧同步标识。

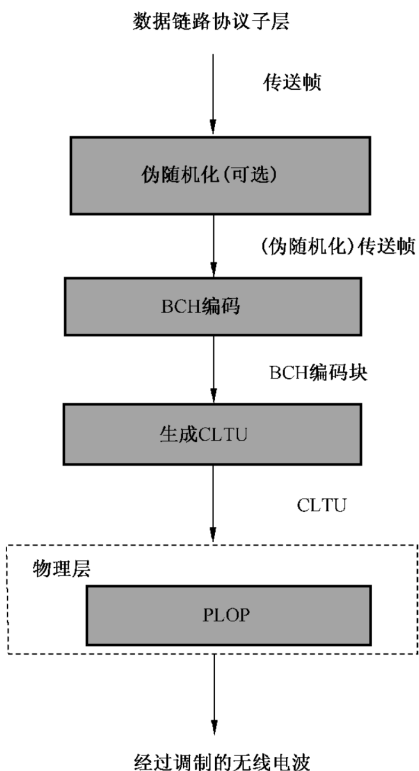


图 2 发送端同步和编码子层的内部流程

4.3.2 接收端

图 3 给出了接收端同步和编码子层的内部流程。该图显示了该层的功能,并描述了这些功能之间的逻辑关系。该图并不表示实际系统中硬件和软件的配置。可选模块取决于实际任务,不是所有的功能都出现在子层中。

在接收端,同步和信道编码子层接收来自物理层的一个连续的、不间断的信道符号数据流,根据任务选择执行功能,向数据链路协议子层发送传送帧。当使用 SMTF 流的 LDPC 码时,接收端进行 CSM 验证和码块去伪随机化操作,具体见 8.2。

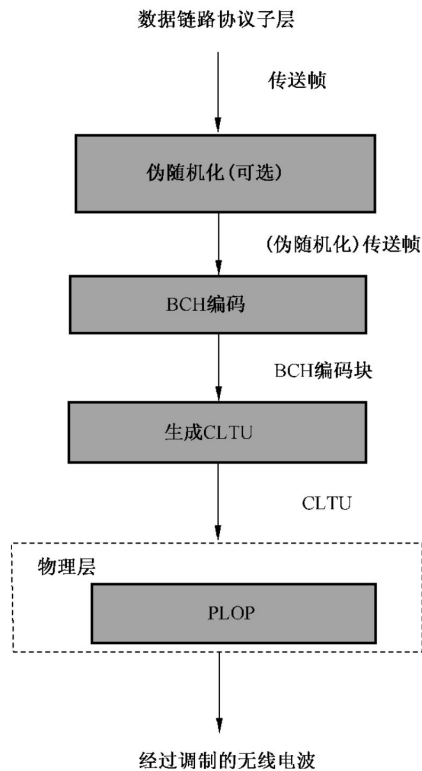


图3 接收端同步和编码子层的内部流程

5 卷积码

5.1 概述

基本卷积码是一种编码码率(r)为 $1/2$, 约束长度(k)为 7 的透明码, 它适合于典型的高斯噪声信道。5.3 对基本卷积码进行了定义。采用 5.4 的方法对基本码进行删减时, 尽管降低了纠错性能, 但却得到了较高的编码码率。

删余码允许从编码码率为 $2/3$ 、 $3/4$ 、 $5/6$ 、 $7/8$ 的四种方案中选择。在纠错性能和信息速率的匹配上从四种方案中可以找到一种与给定服务和数据率最接近的一种选项。

5.2 一般要求

5.2.1 附加同步标识(ASM)

与卷积码一同使用的附加同步标识, 一般采用 32 bit 码型的同步码, 且在卷积码前插入。

5.2.2 伪随机化

使用 BPSK 调制方式时, 基本卷积码的输出端应有一个反相器, 以便保证有足够的比特位翻转速率使接收机中的位同步器能够锁定。当出现下列情况时, 建议对数据进行伪随机化:

- a) 采用多路复用(如 QPSK 或删余卷积码)方案时, 单独的一个反相器不能保证足够的比特位翻转率;
- b) 一些数据码型的卷积码不能实现同步;
- c) 可用于辅助信号捕获, 还能使传输信号的频谱趋于均匀。

5.2.3 帧效验

使用 TM 帧或 AOS 帧传输信号时,应使用 GB/T 39351 和 GB/T 39345 中规定的帧差错控制域 (FECF),以便确认传送帧的有效性,否则就应采用卷积码与 RS 码级联的方案。

注:当误码超出级联码纠错能力时,译码输出中将有无法检测出的突发错误。

5.2.4 量化

当条件(如译码器的复杂度、实现性)允许时,应采用至少是 3 bit 量化的软判决译码。

5.3 基本卷积码

5.3.1 基本特性

基本卷积码是一种非系统码,具体特性如下:

- a) 使用最大似然概率译码(维特比)的卷积码;
- b) 编码码率(r): $1/2$;
- c) 约束长度(k):7 bit;
- d) 码类型:非系统码,透明码;
- e) 连接矢量: $G_1=111100$ (八进制 171)1, $G_2=1011011$ (八进制 133);
- f) 符号反相: G_1 产生的符号 C_1 直接输出, G_2 产生的符号 C_2 反相后输出。

5.3.2 卷积码器的原理框图

卷积码器原理方框图如图 4 所示。图中每输入 1 bit 信息,采样开关 S1 就要完成从位置 1 到位置 2 的一个切换循环,同时产生 2 个输出符号。当采样开关 S1 位于位置 1 时,产生与输入信息比特位相关的第一个输出符号。

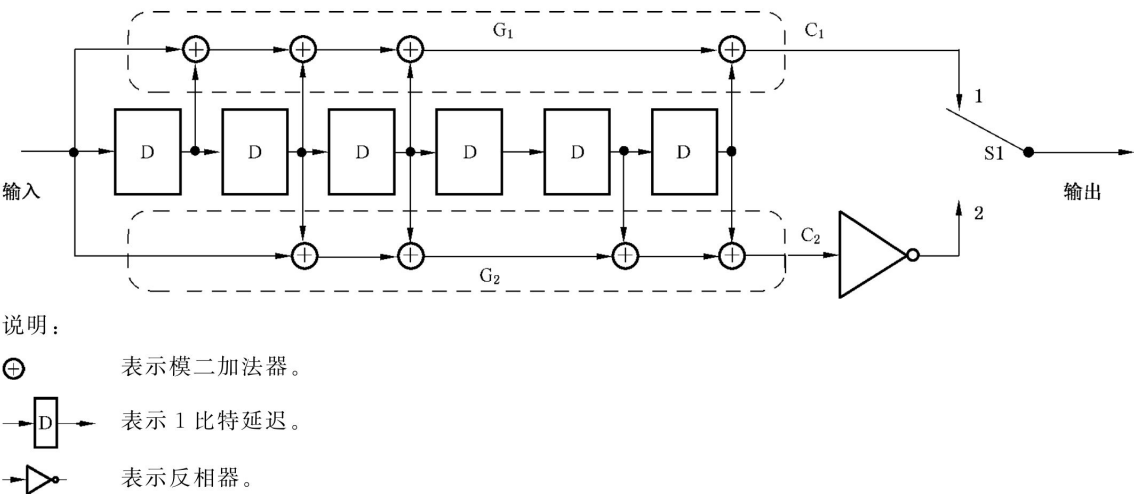


图 4 卷积码器原理图

5.3.3 编码器输入

卷积码器的输入 PCM 信息可以是非归零电平码 NRZ-L 或非归零传号码 NRZ-M;其他种类的 PCM 信号码型在使用卷积码的遥测系统中,不予采用。

5.3.4 编码器输出

卷积码器的输出是二元电平符号序列,即比特位流序列,每输入一个信息比特位,产生二个输出符号(即一个“符号对”,这个“符号对”在符号流中称为一节)。

符号对中的符号输出顺序如下:

- a) 第一输出符号: C_1 的输出;
- b) 第二输出符号: C_2 的反相输出。

具体表现为: $C_1(1), \overline{C_2(1)}, C_1(2), \overline{C_2(2)} \dots$ 。

5.4 删余卷积码

5.4.1 基本特性

删余卷积码应具有下述特性:

- a) 最大似然(维特比)译码的删余卷积码;
- b) 编码码率(r): $1/2$,删余到 $2/3$ 、 $3/4$ 、 $5/6$ 、 $7/8$;
- c) 约束长度(k): 7 bit;
- d) 连接矢量: $G_1=1111001$ (八进制 171), $G_2=1011011$ (八进制 133);
- e) 符号反相:无。

5.4.2 删余卷积码的原理框图

图 5 为删余卷积的编码原理框图,不同删余卷积码比率所对应的输出序列应满足表 1 的要求。

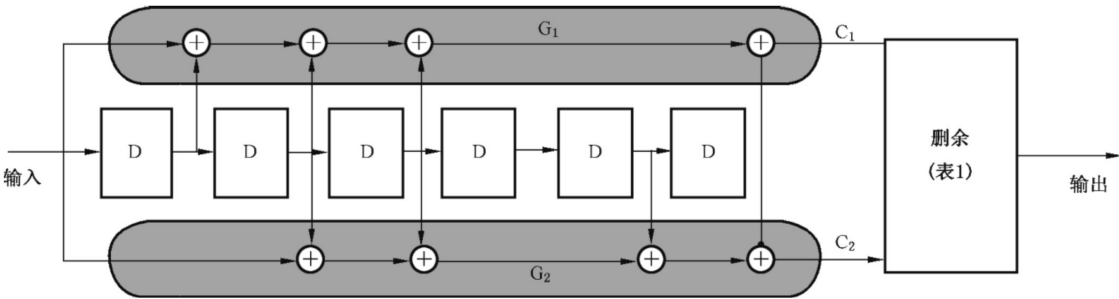


图 5 删余编码器原理图

表 1 不同编码码率的删余卷积码的输出序列

删余卷积码型 1=发送符号 0=不发送符号	编码码率	输出序列 $C_1(t), C_2(t)$ 表示:在比特位时刻 t 时的值
$C_1:10$ $C_2:11$	$2/3$	$C_1(1) C_2(1) C_2(2) \dots$
$C_1:101$ $C_2:110$	$3/4$	$C_1(1) C_2(1) C_2(2) C_1(3) \dots$

表 1 (续)

删余卷积码型 1=发送符号 0=不发送符号	编码码率	输出序列 $C_1(t), C_2(t)$ 表示: 在比特位时刻 t 时的值
$C_1: 1\ 0\ 1\ 0\ 1$ $C_2: 1\ 1\ 0\ 1\ 0$	5/6	$C_1(1)\ C_2(1)\ C_2(2)\ C_1(3)\ C_2(4)\ C_1(5)\cdots$
$C_1: 1\ 0\ 0\ 0\ 1\ 0\ 1$ $C_2: 1\ 1\ 1\ 1\ 0\ 1\ 0$	7/8	$C_1(1)\ C_2(1)\ C_2(2)\ C_2(3)\ C_2(4)\ C_1(5)\ C_2(6)\ C_1(7)\cdots$

6 RS 码(Reed-Solomon 码)

6.1 概述

RS 码对突发性传输错误的纠错很强。本章提供了两种纠错能力的 RS 码供选择使用,一种可纠 8 个 RS 符号错误的 RS(255,239),另一种则可纠 16 个 RS 符号错误的 RS(255,223)。

本标准规定的 RS 码是一种系统码,其中的信息序列部分则不予改变地出现在 RS 码字的信息序列域中。在实际应用中,为进一步增强突发错误的纠错能力,对待传送的数据可以交织分解为 I 路,并分别经过相应的 RS 码器编码后,再交织合成为一个 RS 码块向外传送。

6.2 一般要求

6.2.1 伪随机化

当推荐的 RS 码不能提供足够的传输位来确保接收端政策的同步锁定时,建议对数据进行伪随机化。主要是由于这些码字的伪周期特性而产生错误的码块同步,可能导致无法检测出的解码错误信息。同时伪随机化后的传输信号特征可以帮助信号快速识别和捕获。

6.2.2 帧校验

遥测空间数据链路协议和高级在轨空间数据链路协议中定义的帧差错控制域(FECF)为可选的,系统设计者可以选择使用该域来进行额外的码块有效性确认,特别是对 $E=8$ 的码。对某些应用来说, $E=16$ 的 RS 码具有极低的漏检测率,即使是 $E=8$ 的 RS 码的漏检测率也够用。因此 RS 码可以单独用于码块的确认,也可以单独包含在遥测传送帧或 AOS 传送帧内。

6.3 RS 码的特性

6.3.1 特性参数

RS 码的特性参数如下:

- a) 符号长度 J :即每个 RS 符号所包含的比特位数。每个 RS 符号长度规定为 $J=8$ 。
- b) 纠错能力 E :它表示一个 RS 码字可纠正的 RS 错误符号数目。 E 分别规定为 8 或 16。
- c) 交织深度 I :它表示进行符号交织分解时所选用的交织深度。
- d) 符号长度 J 、纠错能力 E 、交织深度 I 是相互独立的。
- e) 码字长度:它表示每个 RS 码字中包含的 RS 符号数目。RS 码字长度计算公式为: $n=2^J-1=255$ 。由于每个 RS 符号长度为 $J=8$,所以若以比特位为单位,RS 码字长度为 $(2J-1)\times J=$

255×8 bit。

- f) RS 码字内容包括两部分:信息符号部分、校验符号部分。
- g) 校验符号个数:长度为 n 的一个 RS 码字中,所包含的校验符号的个数为 $2E$ 。
- h) 信息符号个数:长度为 n 的一个 RS 码字中,可包含的信息部分的符号个数为 $k=n-2E$ 。

6.3.2 码字的格式

RS 码字格式为: $d_1 d_2 d_3 \cdots d_m \cdots d_k, p_1 p_2 p_3 \cdots p_n \cdots p_{2E}$ 。其中: d_m 为第 m 个信息符号(字节), p_n 为第 n 个校验符号。

6.3.3 码的符号

码的符号要求如下:

- a) RS 码符号是有限域 $GF(2^8)$ 中的元素;每个 RS 符号由 $J=8$ bit 构成,即 $GF(2)$ 上的 8 维行向量;
- b) 有限域 $GF(2^8)$ 在 $GF(2)$ 上的生成多项式见公式(1):

$$F(x) = x^8 + x^7 + x^2 + x + 1 \quad \cdots \cdots \cdots (1)$$

式中:

$F(x)$ ——域生成多项式;

x ——多项式变量。

6.3.4 码的生成多项式

RS 码的生成多项式见公式(2):

$$g(x) = \prod_{j=128-E}^{127+E} (x - \alpha^{11j}) = \sum_{i=0}^{2E} G_i x^i \quad \cdots \cdots \cdots (2)$$

式中:

α ——有限域 $GF(2^8)$ 上的生成元, $F(\alpha)=0$;

α^{11} ——有限域 $GF(2^8)$ 上的本原元;

G_i ——多项式系数, G_i 在常规基中的取值规定参见附录 D 的表 D.1。

$F(x)$ 和 $g(x)$ 共同规定了 $E=16$ 的(255,223)RS 码,以及 $E=8$ 的(255,239)RS 码的特性。

6.3.5 符号交织

多个 RS 码字的符号,逐个依次交织而产生 RS 码块。标准的交织深度规定为 $I=1, 2, 3, 4, 5$ 和 8。对于一项实际任务的物理信道来讲,其交织深度应是固定不变的。

6.3.6 最大码块长度

RS 码的最大码块长度,可用公式(3)表示:

$$L_{\max} = nI = (2^J - 1)I = 255I \quad \cdots \cdots \cdots (3)$$

当交织深度 $I=8$ 时, $L_{\max} = 255 \times 8 = 2\,040$ 个符号。

6.3.7 长度缩短的码块

当数据块(帧)长度小于最大码块长度即 $(n-2E)I$ 个符号时,应使用缩短的码块,但缩短的长度只能是整数个 8I bit。另外由于译码器总是对完整的码块进行译码,为此需引入“虚拟填充”。

kI 个 RS 符号进入“输入”端口后构成 1 个 RS 码块。在 S1 的作用下,每个编码器接收 k 个符号;与原始符号流相比,每个符号之间相隔 I 个符号。S2 按照与 S1 相同的方式排列“输出”端口输出的符

号。如果每个 RS 码字采用了 q 个虚拟填充符号,则使用 $k-q$ 代替以上讨论中的 k 。

6.3.8 RS 码块的结构和虚拟填充

6.3.8.1 RS 码块的结构

RS 码块的结构如图 6 所示。

RS 码块结构的使用规则为:

- 同步标识仅是为识别码块的起始和结束边界,它并不属于码块本身的组成部分。同步标识可以在遥测传输帧产生时插入,也可在进行 RS 码时插入。
- 分包遥测的每一个传输帧将通过 RS 码器构造为一个相应的 RS 码块。不包括同步标识,传输帧的最大长度为 $223 \times 5 \times 8 = 8\,920$ bit。
- 不包括同步标识,RS 码块的最大长度 $= 255 \times I \times 8$ bit; 当 $I = 5$ 时,码块最大长度为 10 200 bit。
- 由于 RS 译码器仅能对完整的 RS 码块正确译码,当实际的传输帧数据长度小于完整码块所要求的长度时,就要对相应的 RS 码器和译码器进行虚拟填充长度的设置。鉴于虚拟填充部分并不在信道上传送的情况,就导致了实际传送的物理码块与包括填充部分在内的完整码块间的差异。另外,包括填充部分在内的完整码块也称为逻辑码块,逻辑码块的长度为 $255 \times I \times 8$ bit。

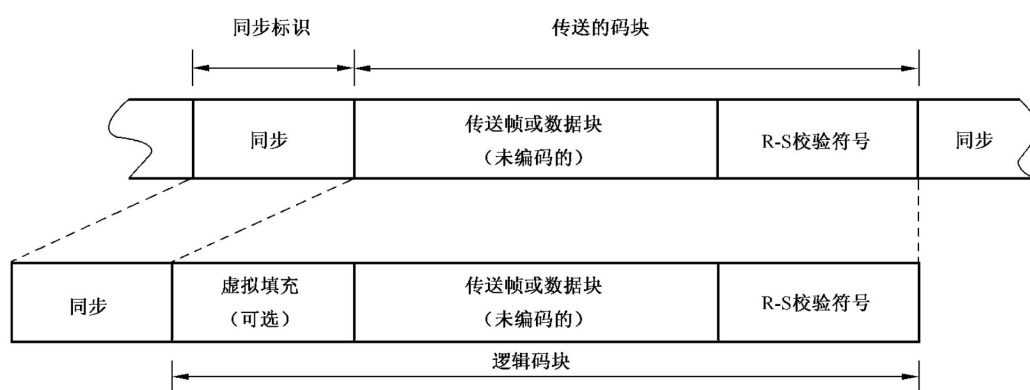


图 6 RS 码块结构

6.3.8.2 虚拟填充

虚拟填充只是在逻辑上完成 RS 码块,其使用规则如下:

- 由全“0”组成;
- 虚拟填充只是设置在编码器和译码器中,不在信道上发送,也不出现在编码器输入端和译码器的输出端;
- 译码器在跟踪过程中不改变虚拟填充的长度;
- 虚拟填充仅插入在码块同步标识与码块中的有效数据之间;
- 虚拟填充的长度只能是 $8I$ bit 的整数倍,且 $n - 2E - L / (8I) \geq 126$,即对每个码组而言,截短后的码组符号长度应大于或等于 $126 + 2E$ 。对于特定任务阶段的某一物理信道,虚拟填充长度应是固定的。

6.3.9 表示及发送顺序

6.3.9.1 在进行 RS 码的编码、译码时,所使用的坐标基有两种,即常规基 $\{1, \alpha^1, \alpha^2, \dots, \alpha^7\}$ 及其对偶基

$\{\ell_0, \ell_1, \ell_2, \dots, \ell_7\}$ 。

6.3.9.2 对偶基的特性为公式(4)：

$$\text{Tr}(\ell_i \beta^j) = \begin{cases} 1, i=j \\ 0, i \neq j \end{cases} \dots\dots\dots (4)$$

式中：

$\text{Tr}(\ell_i \beta^j)$ ——迹函数；

ℓ_i ——对偶基中的基矢量；

β^j ——常规基中的元素,且 $\beta = \alpha^{117}$ ；

$i, j = 0 \sim 7$ 的整数。

6.3.9.3 有限域 $\text{GF}(2^8)$ 上的迹函数定义为公式(5)：

$$\text{Tr}(Z) = \sum_{k=0}^7 Z^{2^k} \dots\dots\dots (5)$$

式中：

$\text{Tr}(Z)$ ——迹函数；

Z ——有限域 $\text{GF}(2^8)$ 中的任一元素。

6.3.9.4 有限域 $\text{GF}(2^8)$ 中任一元素在常规基中的多项式为公式(6)：

$$Z = u_7 \alpha^7 + u_6 \alpha^6 + \dots\dots + u_0 \alpha^0 \dots\dots\dots (6)$$

式中：

u_j ——多项式系数,取值为 0 或 1, j 是 $0 \sim 7$ 的整数；

α^i ——常规基中的基矢量, i 是 $0 \sim 7$ 的整数。

6.3.9.5 有限域 $\text{GF}(2^8)$ 中任一元素中对偶基中的多项式为公式(7)：

$$Z = z_0 \ell^0 + z_1 \ell^1 + \dots\dots + z_7 \ell^7 \dots\dots\dots (7)$$

式中：

z_j ——多项式系数,取值为 0 或 1, j 是 $0 \sim 7$ 的整数。

6.3.9.6 规定 RS 码器输出的编码后符号为对偶基中的 8 bit 矢量 $[z_0, z_1, \dots, z_7]$, 并且发送的先后顺序为 z_0, z_1, \dots, z_7 (z_0 首先发送)。

6.3.9.7 两种坐标基中,符号表达式间的变换关系表示如下：

a) 从常规基到对偶基的变换见公式(8)：

$$[z_0 \dots, z_7] = [u_7 \dots, u_0] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \dots\dots\dots (8)$$

b) 从对偶基到常规基的变换见公式(9)：

$$[u_7 \cdots, u_0] = [z_0 \cdots, z_7] \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \cdots \cdots (9)$$

两种坐标基符号表达式变换的详细信息参见附录 E 所述。

6.3.10 码块同步

为了区分不同的 RS 码块,在每个 RS 码块之前都应附加同步标识,同步标识的长度为 32 bit,二进制表示为:0001 1010 1100 1111 1111 1100 0001 1101,十六进制表示为:1ACFFC1D。

同步标识不包括在 RS 码块之内,另外,同步标识同时作为遥测传送帧与 RS 码块的同步识别标志。

6.4 符号交织

图 7 给出了 RS 交织编码的功能性示意图,对于一个由 (n, k) RS 码以交织深度 I 交织所产生的码块,其码块符号流顺序是由输入编码器的相继 kI 个数据符号,后随 $2EI$ 个校验符号构成。开关 S1 和 S2 同步地循环步进,并且在每个编码器位置上停留 1 个 RS 符号的持续时间。

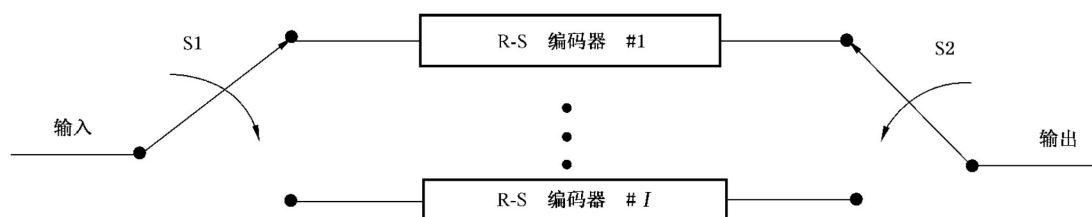


图 7 RS 交织编码框图

以 $I=5$, 对于 (n, k) RS 码交织所产生的初始码块符号流为:

$d_1^1 \cdots d_1^5 d_2^1 \cdots d_2^5 \cdots d_k^1 \cdots d_k^5 [2E \times 5 \text{space}]$, 其中: $k = n - 2E$; $E = 16$, 或 $E = 8$ 。在输出的相同 $[2E \times 5 \text{space}]$ 的区域填充的校验符号为: $p_1^1 \cdots p_1^5 \cdots p_{2E}^1 \cdots p_{2E}^5$ 。

7 级联码

7.1 级联的组合

级联码由 RS 码和某一种卷积码组合而成。

7.2 内码和外码

在级联码中,RS 码作为外码,卷积码作为内码。

7.3 级联信道编码系统框图

级联信道编码系统框图见图 8 所示。RS 码和卷积码并不能保证输出比特位流中含有足够的跳变沿频率,为保证同步信号的检出,图 8 中示出了两个可选的解决跳变频率的方框。对相位模糊问题,可

以采用不归零传号码,也可以采用伪随机化的方法解决。

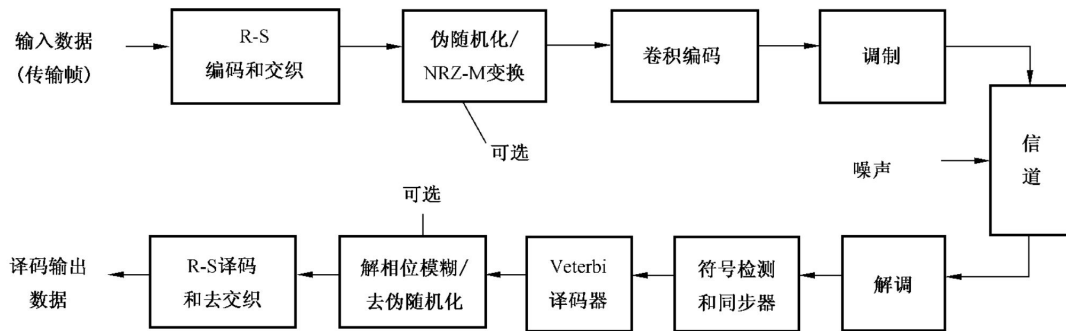


图 8 级联信道编码系统原理框图

8 Turbo 码

8.1 概述

Turbo 码能获得比级联码更大的编码增益,是具有较大码块(上百或上千比特位)的二进制分组码。

8.2 一般要求

8.2.1 数据伪随机化

Turbo 码本身不能保持接收机符号同步的锁定,但能提供足够的比特位转换密度。对数据进行伪随机化处理也有助于信号捕获,并且还能使传输信号的频谱趋于均匀。因此一般建议对数据进行伪随机化处理。

8.2.2 帧有效性确认

Turbo 码用于 TM 或 AOS 传送帧时,应采用 GB/T 39351 和 GB/T 39345 中分别规定的“帧差错控制域”(FECF),以便确认传送帧的有效性。

注: 尽管 Turbo 码具有显著的编码增益,但它仍会在编码输出中漏掉一些未能纠正的差错。

8.3 标准 Turbo 码的特性

8.3.1 标准 Turbo 码的特性参数

标准 Turbo 码的特性参数如下:

- a) 码型: 系统码、非透明码;
- b) 编码分组数目: 2;
- c) 每组码的类型: 递归卷积码;
- d) 每一个卷积分组码的状态数为: 16;
- e) 编码码率: $r=1/2, 1/3, 1/4, 1/6$ 。

8.3.2 信息块及码字长度

信息块长度 k 及相应的码字长度 $n=(k+4)/r$ 应符合表 2 的规定。

表 2 不同编码码率的信息块长度及码字长度

单位为 bit

信息块长度 k	码字长度 n			
	编码码率 1/2	编码码率 1/3	编码码率 1/4	编码码率 1/6
1 784	3 576	5 364	7 152	10 728
3 568	7 144	10 716	14 288	21 432
7 136	14 280	21 420	28 560	42 840
8 920	17 848	26 772	35 696	53 544

8.3.3 码字规范

8.3.3.1 图 9 为 Turbo 码器的实现原理框图,图中“in a”为顺序输入信息块,“in b”为经交织器打乱顺序后的信息块。具体说明如下:

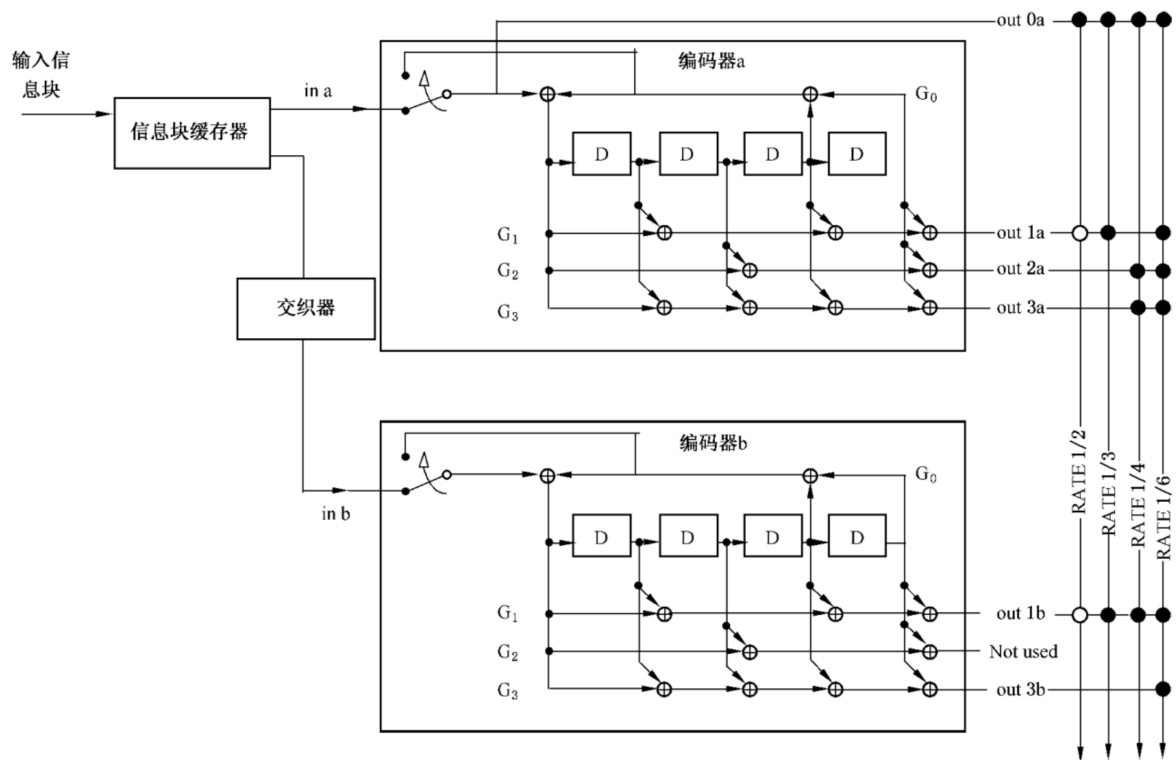
- a) 图 9 中的两个编码器的所有寄存器初始化为 0,运行 $k+4$ 个比特位周期,产生一个具有 $(k+4)/r$ 个编码符号的输出码字,其中 r 是编码码率;
- b) 对于前 k 个比特位周期,输入开关位于低位(图中所示)以接收输入数据;
- c) 对最后的 4 个比特位周期,输入开关切换到高位,从移位寄存器接收反馈信号;

注:该反馈信号使得最后 4 个比特位周期后,四个寄存器都为 0,称为“网格截尾”。

- d) 在“网格截尾”期间,编码器继续输出非零编码符号。对于系统的不编码输出部分(图中的 out 0a 行),包括了 k 比特位信息和 4 比特位反馈信号。

8.3.3.2 在图 9 中,在选定的编码码率下,输出行编码符号从上到下进行循环采样形成 Turbo 码字。具体说明如下:

- a) 对于编码码率为 1/3 的码,输出序列为(out 0a,out 1a,out 1b),该码输出序列重复 $(k-4)$ 次;
- b) 对于编码码率为 1/4 的码,输出序列为(out 0a,out 2a,out 3a,out 1b),该码输出序列重复 $(k-4)$ 次;
- c) 对于编码码率为 1/6 的码,输出序列为(out 0a,out 1a,out 2a,out 3a,out 1b,out 3b),该码输出序列重复 $(k-4)$ 次;
- d) 对于编码码率为 1/2 的码,输出序列是(out 0a,out 1a,out 0a,out 1b),该码输出重复 $(k+4)/2$ 次。



说明:

- ⊕ 表示异或运算
- 表示取每个符号
- 表示取每个其他符号
- D → 表示单比特延迟

图 9 Turbo 码器方框图

四种编码码率下,输出序列构造的 Turbo 码字如图 10 所示。

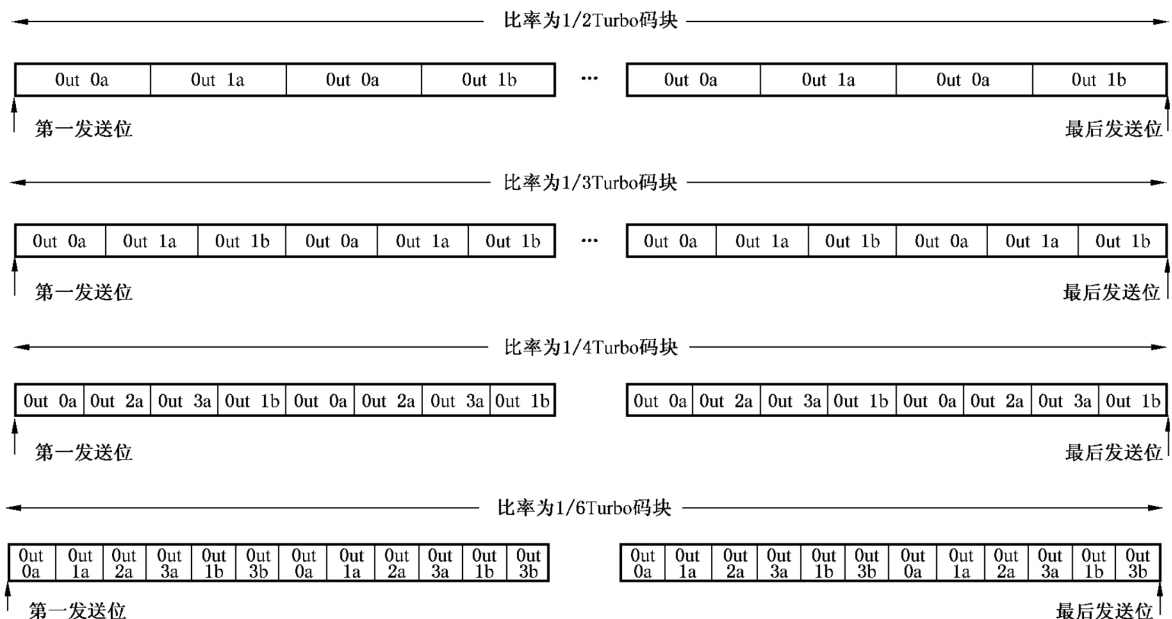


图 10 不同编码码率下的 Turbo 码字

8.3.4 交织器

交织器是 Turbo 编译码过程中的基本模块。Turbo 码的交织器对整个数据块进行逐位的交换。与 RS 码使用的按符号排列的矩形交织器不同, Turbo 码交织器打乱所有位。

对规定的信息块长度 k , Turbo 码对整数 $1, 2, \dots, k$ 按照特殊规则重新排序, k 按下述算法产生:

a) 令 $k = k_1 k_2$, 在规定信息块长度下, 参数 k_1 和 k_2 的取值见表 3。

表 3 不同信息块长度参数 k_1 和 k_2

信息块长度	k_1	k_2
1 784	8	223
3 568	8	223×2
7 136	8	223×4
8 920	8	223×5

b) 对 $s=1$ 到 $s=k$, 执行下述操作, 获得排列数 $\pi(s)$ 见公式(10):

$$p_1 = 31; p_2 = 37; p_3 = 43; p_4 = 47; p_5 = 53; p_6 = 59; p_7 = 61; p_8 = 67$$

$$m = (s-1) \bmod 2$$

$$i = \left\lfloor \frac{s-1}{2k_2} \right\rfloor$$

$$j = \left\lfloor \frac{s-1}{2} \right\rfloor - ik_2$$

$$t = (19i + 1) \bmod \frac{k_1}{2}$$

$$q = t \bmod 8 + 1$$

$$c = (P_q j + 21m) \bmod k_2$$

$$\pi(s) = 2(t + c \frac{k_1}{2} + 1) - m \quad \dots\dots\dots (10)$$

式中:

$\lfloor x \rfloor$ ——小于或等于 x 的表示最大整数。

图 11 给出了交织变换排列方法说明。

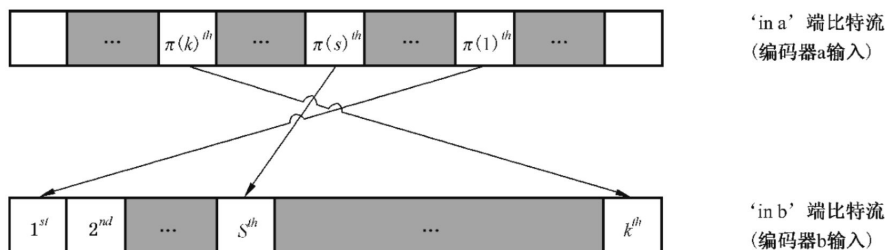


图 11 数据排列说明

8.3.5 反向与前向连接矢量

图 8 中的编码器 a 和编码器 b 各对应一个反向连接矢量 G_0 , 及一组正向连接矢量 G_1, G_2, G_3 , 规定如下:

- a) 所有编码码率下,两个编码器的反向连接矢量: $G_0=10011$;
- b) 编码码率为 $1/2$ 和 $1/3$ 时,两个编码器的前向连接矢量: $G_1=11011$;
- c) 编码码率为 $1/2$ 时,需要进行删余,删余后的符号输出见图 5;
- d) 编码码率为 $1/3$ 时,不进行删余;
- e) 编码码率为 $1/4$ 时,编码器 a 的前向连接矢量: $G_2=10101$, $G_3=11111$;编码器 b 的前向连接矢量: $G_1=11011$,不进行删余;
- f) 编码码率为 $1/6$ 时,编码器 a 的前向连接矢量: $G_1=11011$, $G_2=10101$, $G_3=11111$;编码器 b 的前向连接矢量: $G_1=11011$, $G_3=11111$,不进行删余。

8.4 同步标识

Turbo 码同步标识为识别 Turbo 码字的起始和结束边界,不属于码字本身的组成部分。同步标识可以在遥测传输帧产生时插入,也可在编码时插入;同步标识与编码码率相关联,置于 Turbo 码字之前。带有同步标识的 Turbo 码字见图 12。不同编码速率对应的同步标识信息如下:

- a) 编码码率为 $1/2$ 时,同步标识:034776C7272895B0
- b) 编码码率为 $1/3$ 时,同步标识:25D5C0CE8990F6C9461BF79C
- c) 编码码率为 $1/4$ 时,同步标识:034776C7272895B0 FCB88938D8D76A4F
- d) 编码码率为 $1/6$ 时,同步标识:25D5C0CE8990F6C9461BF79C DA2A3F31766F0936B9E40863



图 12 带有附加同步标识的 Turbo 码字

9 LDPC 码传送帧

9.1 概述

低密度奇偶校验(LDPC)码是一类码长较长的二进制分组码,码长达到几百或几千 bit,可获得比级联编码更高的编码增益。

LDPC 码可通过一个 $v \times w$ 维的奇偶校验矩阵 H 来定义, H 由 v 个线性无关的行向量组成。校验矩阵还可以包含额外的线性相关行,这不改变码字的校验关系。一个 w 位长的码字序列必定位于 H 的 $w-v$ 维对偶空间中,即它应满足与 H 的 v 行相对应的所有 v 个奇偶校验方程。LDPC 码还可以通过一个 $(w-v) \times w$ 维的生成矩阵来描述,此时,码字序列位于生成矩阵的 $w-v$ 维空间中。编码器将 k bit 的输入信息序列唯一地映射成 n bit 的码字序列。对 LDPC 码字进行缩短或删除时, $k < w-v$,此时多余的编码码字将不被使用。还可以对 LDPC 码进行扩展或删余,以使码长 n 大于或小于 w 。

LDPC 码的显著特征是校验矩阵 H 中“1”的密度很低。与此不同,生成矩阵通常不具有稀疏特性,在生成矩阵的非系统部分,“1”和“0”的密度相当。

9.2 一般要求

9.2.1 同步

在 9.3 中描述的(8160,7136)码,将与图 13 所示的 32 bit 地址同步字(ASM)一同使用。在 9.4 中定义的速率分别为 1/2、2/3 和 4/5 的 9 种码,将与图 14 所示的 64 bit 地址同步字一同使用。

注:差分编码和 LDPC 码一起使用不能带来益处,同步头可用来解决相位模糊。实际上,差分编码不能应用在 LDPC 码之前,因为本标准提供的 LDPC 码是非透明的;差分编码也不建议在 LDPC 码之后,这会造成相当明显的性能损失,而且还需进行差分检测,使其在软判决的情况下会更加复杂,即需在译码之前检测和解决相位模糊问题。

9.2.2 数据随机化

由于 LDPC 码本身不具有均衡的比特转换特性,因此无法保证接收端的符号时钟同步。由于这种码具有准循环特性,未检出的译码差错可能会引起错误的码字同步。伪随机码发生器也有助于捕获信号,并使得发送信号中的离散谱线趋于均匀。因此一般建议对数据进行伪随机化处理。

9.2.3 帧效验

LDPC 译码器可单独用于检测码字的有效性,因此可验证码字中所包含的 TM 传送帧或 AOS 传送帧的有效性。帧差错控制域(FECT)是可选项,系统设计时可以用它作为额外的帧有效性检测手段。

对于任意工作情况下的信噪比,LDPC 码不能检测出帧或比特位的差错率位于相应可检测差错率的几个数量级以下。

9.3 编码码率为 223/255 的 LDPC 码

9.3.1 概述

本标准规定的(8160,7136)码是在(8176,7156)码基础上进行删余、缩短和扩展后得到的。(8160,7136)码的码率为 223/255,码字长度和信息位长度均与 GB/T 39349 中交织深度为 $I=4$ 的 RS(255,223)码相同。(8176,7156)码为透明码,其缩短形式的(8160,7136)码包含填充比特位,为非透明码。

9.3.2 (8176,7156)码校验矩阵

(8176,7156)码的校验矩阵是由以 511×511 的循环矩阵为元素构成的 2×16 矩阵产生的。校验矩阵的结构如图 13 所示。校验矩阵大小为 1022×8176 ,秩为 1020。

$$\begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} & A_{1,6} & A_{1,7} & A_{1,8} & A_{1,9} & A_{1,10} & A_{1,11} & A_{1,12} & A_{1,13} & A_{1,14} & A_{1,15} & A_{1,16} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} & A_{2,5} & A_{2,6} & A_{2,7} & A_{2,8} & A_{2,9} & A_{2,10} & A_{2,11} & A_{2,12} & A_{2,13} & A_{2,14} & A_{2,15} & A_{2,16} \end{bmatrix}$$

图 13 (8176,7156)码的校验矩阵

$A_{i,j}$ 是 511×511 的循环矩阵。每个循环矩阵的行重都为 2,即每行有 2 个“1”。因此,校验矩阵每行的行重为 $2 \times 16 = 32$ 。表 4 的第 2 列给出了每个循环矩阵第 1 行中“1”的位置,下边的每一行由上一行循环右移 1 位得到,“1”有 511 个可能的位置,位置数从 0 到 510;表 4 的第 3 列给出了校验矩阵中“1”的绝对位置,有 8176 个可能的位置,位置数从 0 到 8175。

表 4 循环矩阵规定

循环矩阵	循环矩阵第 1 行“1”的位置	循环矩阵第 1 行“1”在校验矩阵中
$A_{1,1}$	0,176	0,176
$A_{1,2}$	12,239	523,750
$A_{1,3}$	0,352	1022,1374
$A_{1,4}$	24,431	1557,1964
$A_{1,5}$	0,392	2044,2436
$A_{1,6}$	151,409	2706,2964
$A_{1,7}$	0,351	3066,3417
$A_{1,8}$	9,359	3586,3936
$A_{1,9}$	0,307	4088,4395
$A_{1,10}$	53,329	4652,4928
$A_{1,11}$	0,207	5110,5317
$A_{1,12}$	18,281	5639,5902
$A_{1,13}$	0,399	6132,6531
$A_{1,14}$	202,457	6845,7100
$A_{1,15}$	0,247	7154,7401
$A_{1,16}$	36,261	7701,7926
$A_{2,1}$	99,471	99,471
$A_{2,2}$	130,473	641,984
$A_{2,3}$	198,435	1220,1457
$A_{2,4}$	260,478	1793,2011
$A_{2,5}$	215,420	2259,2464
$A_{2,6}$	282,481	2837,3036
$A_{2,7}$	48,396	3114,3462
$A_{2,8}$	193,445	3770,4022
$A_{2,9}$	273,430	4361,4518
$A_{2,10}$	302,451	4901,5050
$A_{2,11}$	96,379	5206,5489
$A_{2,12}$	191,386	5812,6007
$A_{2,13}$	244,467	6376,6599
$A_{2,14}$	364,470	7007,7113
$A_{2,15}$	51,382	7205,7536
$A_{2,16}$	192,414	7857,8079

9.3.3 (8176,7154)LDPC 码及其生成矩阵

本标准规定的 LDPC 码校验矩阵的大小为 1022×8176 , 其对应的基本生成矩阵的大小为 $(8176, 7156)$, 因为该 LDPC 码校验矩阵的秩为 1020, 所以会导致有 2 bit 的信息位于准循环编码器的结构之外, 这就增加了获得生成矩阵的复杂度, 通过删余 2 bit 的信息位后可得到其系统形式的子码 $(8176, 7154)$, 如图 14 所示, 该生成矩阵可以完全由循环矩阵来构造, 子码生成矩阵 $(8176, 7154)$ 的左边部分是一个 7154×7154 的单位矩阵, 这里将其表示为分块矩阵的形式, 其中 I 表示 511×511 的单位矩阵, 0 表示 511×511 的全 0 矩阵。矩阵的右边为 2 列 511×511 的循环矩阵, 其构造方法如下:

- $D = \begin{bmatrix} A_{1,15} & A_{1,16} \\ A_{2,15} & A_{2,16} \end{bmatrix}$, 由图 14 和表 4 定义, 是一个 1022×1022 的矩阵;
 - 定义 $u = (1\ 0\ 0\ 0 \cdots 0)$ 为包含 511 个元素的单位数组, 其中最左侧位置为‘1’, 其他位置为‘0’;
 - $z_i = (b_{i,1}\ b_{i,2})$, 其中 $i = 1, 2, \dots, 14$, $b_{i,j}$ 是循环矩阵 $B_{i,j}$ 的第 1 行;
 - $M_i = \begin{bmatrix} A_{1,i} \\ A_{2,i} \end{bmatrix}$, 其中 $i = 1, 2, \dots, 14$, 校验矩阵可表示为 $[M_1\ M_2 \cdots M_{14}\ D]$;
 - z_i 的第 511 和第 1 022 个元素设为 0, 求解方程 $M_i u^T + D z_i^T = 0$ 得到 z_i , 其中, $i = 1, 2, \dots, 14$, T 表示矩阵转置;
- 注: 由于 D 的秩是 1 020 而不是 1 022, 因此 D 中有两列可用其他列线性表示。这两列可取为第 511 和第 1 022 列。
- $b_{i,j}$ 可从 z_i 中提取出来。 $b_{i,j}$ 的值参见附录 F。

$$\begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{1,1} & B_{1,2} \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{2,1} & B_{2,2} \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{3,1} & B_{3,2} \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{4,1} & B_{4,2} \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{5,1} & B_{5,2} \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{6,1} & B_{6,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{7,1} & B_{7,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & B_{8,1} & B_{8,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & B_{9,1} & B_{9,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & B_{10,1} & B_{10,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & B_{11,1} & B_{11,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & B_{12,1} & B_{12,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & B_{13,1} & B_{13,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & B_{14,1} & B_{14,2} \end{bmatrix}$$

图 14 (8176,7154)码的生成矩阵

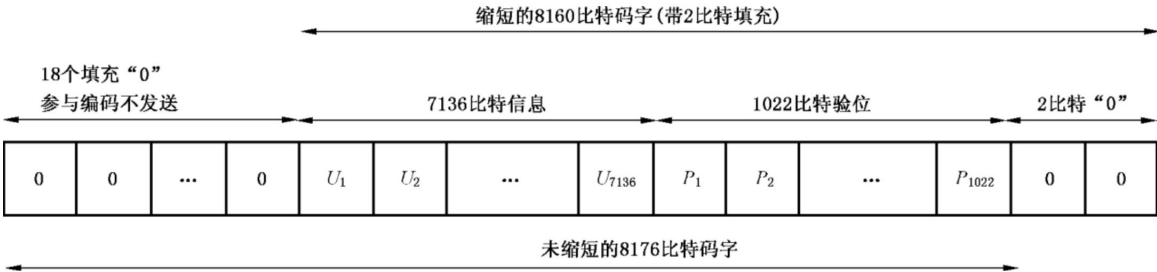
9.3.4 编码

(8176,7154)码按照如下方式缩短、扩展后得到一个参数是 32 倍数的 $(8160, 7136)$ 码, 编码码率为 $223/255$, 具体说明如下:

- 编码器接收长度为 7 136 bit 的遥测传送帧作为输入;
- 将 18 个“0”放置在待编码的 7 136 bit 信息之前, 产生一个 7 154 bit 元素的行矢量;
- 这个矢量乘以 9.3.3 定义的生成矩阵, 产生了一个 8 176 bit 的矢量, 这个矢量由 18 个 0、7 136 个

系统的信息符号和 1 022 个校验符号组成；

- d) 丢弃这个矢量中的 18 个前导“0”，在末端附加 2 个“0”，就产生了一个 8 160 bit 的码字。
18 个填充比特位、7136 信息比特位、1022 效验比特位和 2 个填充位“0”的格式如图 15 所示。



9.4 编码码率为 1/2, 2/3 和 4/5 的 LDPC 码簇

9.4.1 概述

本标准规定的 LDPC 码簇信息长度 k 包括 1 024 bit, 4 096 bit, 16 384 bit, 编码码率 r 包括 1/2, 2/3 以及 4/5, 共 9 种码。这些参数及其对应的码长度 $n = k/r$ 见表 5。

表 5 不同编码码率对应的码字长度 单位为 bit

遥测传送帧或信息块长度 k	码字长度 n		
	编码码率 1/2	编码码率 2/3	编码码率 4/5
1 024	2 048	1 536	1 280
4 096	8 192	6 144	5 120
16 384	32 768	24 576	20 480

9.4.2 校验矩阵

码簇的校验矩阵由 $M \times M$ 的子矩阵构造, 不同编码码率对应的子矩阵大小见表 6。

表 6 不同编码码率对应子矩阵大小 M

信息位长度 k	子矩阵大小 M		
	编码码率 1/2	编码码率 2/3	编码码率 4/5
1 024	512	256	128
4 096	2 048	1 024	512
16 384	8 192	4 096	2 048

具体说明如下：

- a) 编码码率为 1/2 的 LDPC 码, 校验矩阵应满足公式(11)：

$$H_{1/2} = \begin{bmatrix} 0_M & 0_M & I_M & 0_M & I_M \oplus \Pi_1 \\ I_M & I_M & 0_M & I_M & \Pi_1 \oplus \Pi_3 \oplus \Pi_4 \\ I_M & \Pi_5 \oplus \Pi_6 & 0_M & \Pi_7 \oplus \Pi_8 & I_M \end{bmatrix} \quad \dots\dots\dots (11)$$

b) 编码码率为 2/3 的 LDPC 码, 校验矩阵应满足公式(12):

$$H_{2/3} = \left[\begin{array}{cc|c} 0_M & 0_M & \\ \hline \Pi_9 \oplus \Pi_{10} \oplus \Pi_{11} & I_M & \\ I_M & \Pi_{12} \oplus \Pi_{13} \oplus \Pi_{14} & \end{array} \right] H_{1/2} \quad \dots\dots\dots (12)$$

c) 编码码率为 4/5 的 LDPC 码, 校验矩阵应满足公式(13):

$$H_{4/5} = \left[\begin{array}{cccc|c} 0_M & 0_M & 0_M & 0_M & \\ \hline \Pi_{21} \oplus \Pi_{22} \oplus \Pi_{23} & I_M & \Pi_{15} \oplus \Pi_{16} \oplus \Pi_{17} & I_M & \\ I_M & \Pi_{24} \oplus \Pi_{25} \oplus \Pi_{26} & I_M & \Pi_{18} \oplus \Pi_{19} \oplus \Pi_{20} & \end{array} \right] H_{2/3} \quad \dots\dots\dots (13)$$

式中:

I_M —— $M \times M$ 的单位矩阵;

0_M —— $M \times M$ 的 0 矩阵;

$\Pi_1 \sim \Pi_{26}$ ——置换矩阵。

置换矩阵 Π_K 在第 i 行和第 $\pi_K(i)$ 列应为非 0 项。其中: $i \in \{0, \dots, M-1\}$, $\pi_K(i) = \frac{M}{4}((\theta_K + \lfloor 4i/M \rfloor) \bmod 4) + (\phi_K(\lfloor 4i/M \rfloor, M) + i) \bmod \frac{M}{4}$, θ_K 和 $\phi_K(j, M)$ 在表 7 及表 8 中定义。

表 7 $\phi_K(0, M)$ 和 $\phi_K(1, M)$ 的定义

k	θ_k	$\phi_K(0, M)$								$\phi_K(1, M)$							
		$M=2^7 \dots 2^{13}$								$M=2^7 \dots 2^{13}$							
1	3	1	59	16	160	108	226	1	148	0	0	0	0	0	0	0	0
2	0	22	18	103	241	126	618	2	032	27	32	53	182	375	767	1	822
3	1	0	52	105	185	238	404	249		30	21	74	249	436	227	203	
4	2	26	23	0	251	481	32	1	807	28	36	45	65	350	247	882	
5	2	0	11	50	209	96	912	485		7	30	47	70	260	284	1	989
6	3	10	7	29	103	28	950	1	044	1	29	0	141	84	370	957	
7	0	5	22	115	90	59	534	717		8	44	59	237	318	482	1	705
8	1	18	25	30	184	225	63	873		20	29	102	77	382	273	1	083
9	0	3	27	92	248	323	971	364		26	39	25	55	169	886	1	072
10	1	22	30	78	12	28	304	1	926	24	14	3	12	213	634	3	54
11	2	3	43	70	111	386	409	1	241	4	22	88	227	67	762	1	942
12	0	8	14	66	66	305	708	1	769	12	15	65	42	313	184	446	
13	2	25	46	39	173	34	719	532		23	48	62	52	242	696	1	456
14	3	25	62	84	42	510	176	768		15	55	68	243	188	413	1	940
15	0	2	44	79	157	147	743	1	138	15	39	91	179	1	854	1	660

表 7 (续)

k	θ_k	$\phi_K(0,M)$							$\phi_K(1,M)$						
		$M=2^7 \dots 2^{13}$							$M=2^7 \dots 2^{13}$						
16	1	27	12	70	174	199	759	965	22	11	70	250	306	544	1 661
17	2	7	38	29	104	347	674	141	31	1	115	247	397	864	587
18	0	7	47	32	144	391	958	1 527	3	50	31	164	80	82	708
19	1	15	1	45	43	165	964	505	29	40	121	17	33	1009	1 466
20	2	10	52	113	181	414	11	1 312	21	62	45	31	7	437	433
21	0	4	61	86	250	97	413	1 840	2	27	56	149	447	36	1 345
22	1	19	10	1	202	158	925	709	5	38	54	105	336	562	867
23	2	7	55	42	68	86	687	1 427	11	40	108	183	424	816	1 551
24	1	9	7	118	177	168	752	989	26	15	14	153	134	452	2 041
25	2	26	12	33	170	506	867	1 925	9	11	30	177	152	290	1 383
26	3	17	2	126	89	489	323	270	17	18	116	19	492	778	1 790

表 8 $\phi_K(2,M)$ 和 $\phi_K(3,M)$ 的定义

k	θ_k	$\phi_K(2,M)$							$\phi_K(3,M)$						
		$M=2^7 \dots 2^{13}$							$M=2^7 \dots 2^{13}$						
1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	12	46	8	35	219	254	318	13	44	35	162	312	285	1 189
3	1	30	45	119	167	16	790	494	19	51	97	7	503	554	458
4	2	18	27	89	214	263	642	1 467	14	12	112	31	388	809	460
5	2	10	48	31	84	415	248	757	15	15	64	164	48	185	1 039
6	3	16	37	122	206	403	899	1 084	20	12	93	11	7	49	1 000
7	0	13	41	1	122	184	328	1 630	17	4	99	237	185	101	1 265
8	1	9	13	69	67	279	518	64	4	7	94	125	328	82	1 223
9	0	7	9	92	147	198	477	689	4	2	103	133	254	898	874
10	1	15	49	47	54	307	404	1 300	11	30	91	99	202	627	1 292
11	2	16	36	11	23	432	698	148	17	53	3	105	285	154	1 491
12	0	18	10	31	93	240	160	777	20	23	6	17	11	65	631
13	2	4	11	19	20	454	497	1 431	8	29	39	97	168	81	464
14	3	23	18	66	197	294	100	369	22	37	113	91	127	823	461
15	0	5	54	49	46	479	518	352	19	42	92	211	8	50	844
16	1	3	40	81	162	289	92	1 177	15	48	119	128	437	413	392
17	2	29	27	96	101	373	464	836	5	4	74	82	475	462	922
18	0	11	35	38	76	104	592	1 572	21	10	73	115	85	175	256
19	1	4	25	83	78	141	198	348	17	18	116	248	419	715	1 986
20	2	8	46	42	253	270	856	1 040	9	56	31	62	459	537	19
21	0	2	24	58	124	439	235	779	20	9	127	26	468	722	266
22	1	11	33	24	143	333	134	476	18	11	98	140	209	37	471
23	2	11	18	25	63	399	542	191	31	23	23	121	311	488	1 166
24	1	3	37	92	41	14	545	1 393	13	8	38	12	211	179	1 300
25	2	15	35	38	214	277	777	1 752	2	7	18	41	510	430	1 033
26	3	13	21	120	70	412	483	1 627	18	24	62	249	320	264	1 606

9.4.3 生成矩阵

本标准规定的码簇支持比率为 $K/(K+2)$ 的 LDPC 码,当编码码率为 $1/2$ 时 $K=2$,编码码率为 $2/3$ 时 $K=4$,编码码率为 $4/5$ 时 $K=8$ 。如果包含删余列,生成矩阵大小为 $MK \times M(K+3)$;如果去掉删余列,则生成矩阵大小为 $MK \times M(K+2)$ 。

生成矩阵按如下方式构造:

- P 为 H (不同编码码率的校验矩阵 H 定义见 9.3.2) 的 $3M \times 3M$ 的子矩阵,由 H 的最后 $3M$ 列构成; Q 为 H 的 $3M \times MK$ 子矩阵,由 H 的起始 MK 列构成;
- 运算 $W = (P^{-1}Q)^T$,其中运算为模 2 算法;
- 构造矩阵 $G = [IMK, W]$,其中 IMK 是 $MK \times MK$ 的单位矩阵, W 是 $MK \times 3M$ 的循环移位矩阵;
- 删除生成矩阵的最后 M 列。

9.4.4 对遥测数据的 $1/2$ 、 $2/3$ 、 $4/5$ 比率编码

编码步骤如下:

- 编码器接收如表 5 所示的长度为 k 的遥测传送帧作为输入;
- 遥测传送帧与生成矩阵相乘。

9.5 同步标识

9.5.1 LDPC 码同步标识是为识别 LDPC 码字的起始和结束边界,它并不属于 LDPC 码字本身的组成部分。

9.5.2 编码码率为 $223/255$ 的 LDPC 码同步标识:1ACFFC1D。

9.5.3 编码比率 $1/2$ 、 $2/3$ 、 $4/5$ 的 LDPC 码同步标识:034776C7272895B0。

10 帧同步

10.1 概述

10.1.1 同步

帧或码块同步对于 RS 码、Turbo 码和 LDPC 码的正确译码,以及随后的传送帧处理是必要的。另外,同步对伪随机序列发生器的同步以及卷积译码器的节点同步也是需要的。

一般要求如下:

- 对于采用 5.3 规定的基本卷积码的编码系统,附加同步标识(ASM)可以在信道符号域提取(即任何译码之前),也可在卷积译码器译码后的比特位域提取。
- 对于 RS 码与 5.3 定义的基本卷积码的级联系统,ASM 可在信道符号域提取(即任何译码之前),或在内码译码后的比特位域(即 RS 码的符号域)中提取。
- 对于采用 5.4 定义的删余卷积码的编码系统,ASM 仅能在卷积译码后的比特位域中提取,而不能在信道符号域中(即译码前)提取。
- 对于 RS 码与 5.4 定义的删余卷积码相级联的编码系统中,ASM 仅能在内码译码的比特位域中(即 RS 码的符号域)提取,即它不能在信道符号域(任何译码前)提取。
- 对于 Turbo 或 LDPC 码系统,ASM 仅能在信道符号域(即任何译码前,在 Turbo 或 LDPC 码的码符号域)提取。

10.1.2 信道访问数据单元

由 ASM 与传送帧(若物理信道上没有 RS、Turbo 或 LDPC 码时)组成的数据单元,或 ASM 与 RS 码块、或 Turbo 或 LDPC 码字(若物理信道上有 RS 或 Turbo 或 LDPC 码)组成的数据单元都称为信道访问数据单元(CADU)。其中 CADU 中的传送帧、码字或码块可以是随机化的,或是没有随机化的。不同编码选择的 CADU 的组成见表 9。

表 9 RS 码的管理参数表

应用的编码方式	信道访问数据单元(CADU)
卷积码	ASM+传送帧
RS 码	AM+RS 码块
级联码	AM+RS 码块
Turbo 码	ASM + Turbo 码字
传送帧的 LDPC 码	ASM + LDPC 码字
SMTF 数据流的 LDPC 码	CSM + LDPC 码块

10.2 附加同步标识(ASM)

10.2.1 RS 码块或 Turbo 或 LDPC 码字的同步(或传送帧,如果物理信道不是 RS 码或 Turbo 码或 LDPC 码),是采用一串固定长度的码块或码字(或传送帧)流,并在它们之间附加同步标志(ASM),而进行同步。

注:接收端同步的提取是通过识别物理信道中数据流的附加同步标志(ASM)的特定比特图样,通过进一步验证而实现的。

10.2.2 如果采用卷积码,则 ASM 应是经过卷积码的。

10.2.3 如果卷积码为内码、RS 为外码的级联系统,则 ASM 应由内码编码,而不能用外码编码(见第 5 章)。

10.3 ASM 的比特图样

10.3.1 对于没有使用 Turbo 或 LDPC 码的数据,ASM 应由图 16 所示的 32 bit(4 字节)标志码型构成。

10.3.2 对于编码比率为 1/2、1/3、1/4 或 1/6 的 Turbo 码数据,ASM 应为图 17~图 20 所示的 $32/r$ bit ($4/r$ 字节)的标志码型。

10.3.3 对于编码比率为 7/8 的 LDPC 码数据,ASM 应为图 16 所使得 32 bit 码型构成。

10.3.4 对于编码比率为 1/2、2/3 或 4/5 的 LDPC 码数据,ASM 应为图 17 所示的 64 bit 码型构成。ASM 的比特位码型以 16 进制数表示如下:

- a) 未编码数据、卷积、RS、级联和 7/8 比率 LDPC 码数据的 ASM:1ACFFC1D;
- b) 1/2 比率 turbo 和 1/2、2/3、4/5 比率的 LDPC 码数据的 ASM:034776C7272895B0;
- c) 1/3 比率 Turbo 码数据 ASM:25D5C0CE8990F6C9461BF79C;
- d) 1/4 比率 Turbo 码数据 ASM:034776C7272895B0 FCB88938D8D76A4F;
- e) 1/6 比率 Turbo 码数据 ASM:25D5C0CE8990F6C9461BF79C DA2A3F31766F0936B9E40863。

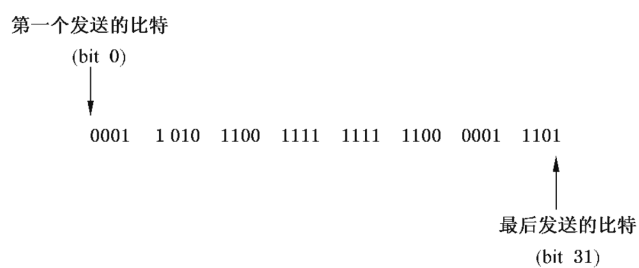


图 16 无编码、卷积码、RS 码、级联码和 7/8 比率 LDPC 码的 ASM 比特码型

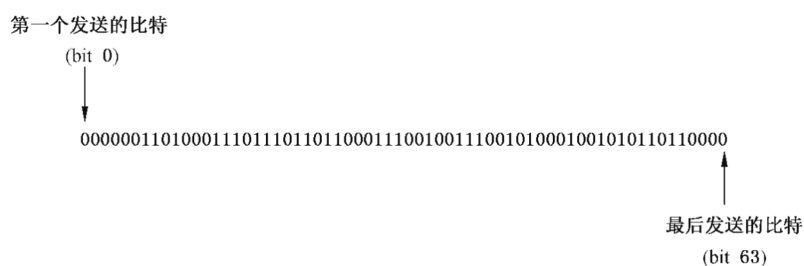


图 17 1/2 比率 Turbo 码和 1/2, 2/3, 4/5 比率 LDPC 码(应用到传送帧)的 ASM 比特码型

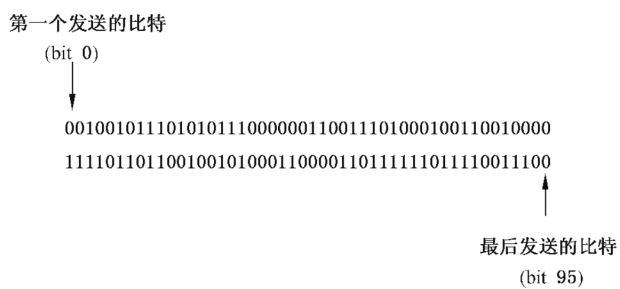


图 18 1/3 比率 Turbo 码的 ASM 比特码型

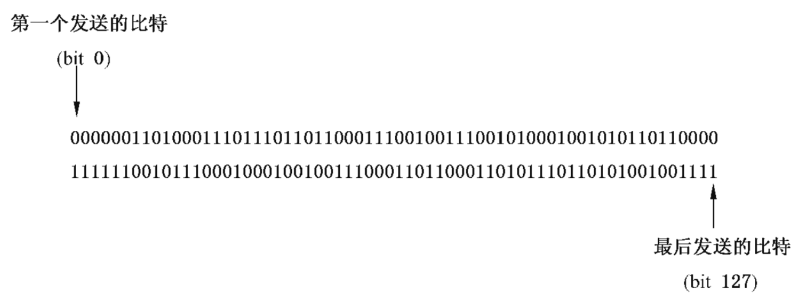


图 19 1/4 比率 Turbo 码的 ASM 比特码型

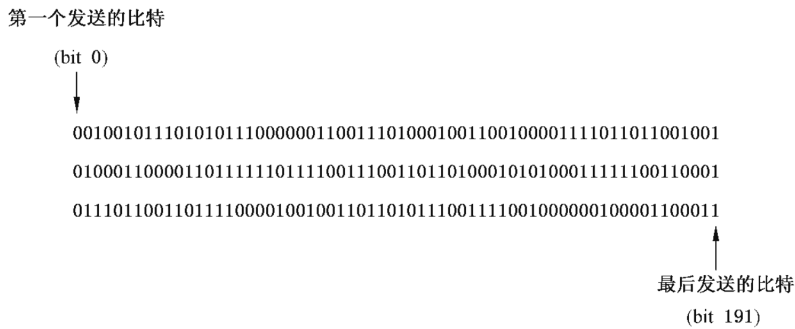


图 20 1/6 比率 Turbo 码的 ASM 比特码型

10.4 ASM 的位置

- 10.4.1 ASM 应直接添加到码块(RS 码)、码字(Turbo 或 LDPC 码)或传送帧(卷积码或未编码)之前。
- 10.4.2 ASM 应立即跟随在前边的码块(若为 RS 码)、码字(若为 Turbo 或 LDPC 码)或传送帧(若仅是卷积码或未编码的)之后;即 ASM 之前不应出现干扰比特位。

10.5 ASM 与 RS、Turbo 及 LDPC 码块和码字的关系

- 10.5.1 ASM 不是 RS 码块的编码数据空间的一部分,它也不应提供给 RS 码器或解码器的输入端。可防止编码器在某些相关数重复出现的条件下,在校验符号域程序性的产生另一个类似的标识(例如,其中的测试码型 01010101010),从而避免引起对接收端同步的干扰。图 6 描述了 ASM、RS 码块和传送帧间的关系。
- 10.5.2 ASM 不出现在 Turbo 码器或解码器的输入端。它应直接添加在 Turbo 码字前。
- 10.5.3 ASM 不出现在 LDPC 码器或解码器的输入端。它应直接添加在 LDPC 码字前。

10.6 嵌入数据流中的 ASM

为了与主数据流同步标志相区别,嵌入数据流的 ASM 应由如下的 32 位(4 字节)标志构成,其比特位码型为:00110101 00101110 11111000 01010011(bit 0~bit 31),其十六进制表示为:352EF853。

11 伪随机化

11.1 概述

为保证传送的二进制比特位流中有足够的电平跳变密度,可以对 RS 码块、传送帧等信道编码进行伪随机化,但对同步标识不能伪随机化。

11.2 伪随机化说明

实现伪随机化的方法是将码块、码字、或传送帧重的每一个比特位与标准的伪随机序列进行异或运算。如果使用伪随机发生器,在发送端就应对完成 RS、turbo、或 LDPC 码后的码块、码字、或传送帧进行随机化处理,发送端的配置如图 21 所示。

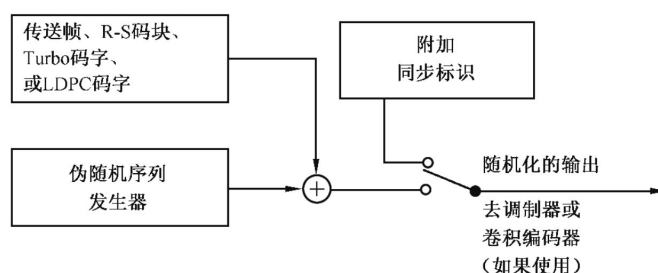


图 21 伪随机发生器配置

11.3 伪随机化的同步与应用

11.3.1 附加同步标志(ASM)应用于伪随机化的同步。

11.3.2 伪随机序列应在码块、码字或传送帧的第 1 个比特位的起始处就开始使用。在发送端码块、码字和传送帧的随机化,是将码块、码字或传送帧的第 1 比特位与伪随机序列的第 1 比特位进行异或运算,紧接着将码块、码字或传送帧的第 2 比特位与伪随机序列的第 2 比特位进行异或运算,依次类推。

11.3.3 接收端,初始码块、码字或传送帧使用同样的伪随机序列进行重构(去随机化)。

11.3.4 对接收数据流中的 ASM 定位后,应直接对跟随在 ASM 后的数据进行去伪随机化。

11.4 伪随机序列

伪随机序列的生成多项式为 $h(x) = x^8 + x^7 + x^5 + x^3 + 1$ 。该伪随机序列的长度为 255 bit。伪随机序列从码块或传送帧的第 1 个比特位开始, 经过一个伪随机序列周期的循环(255 bit)后继续重复循环, 直到码块或传送帧的结束为止。

伪随机序列的第1个比特位与码块或传送帧的第1个比特位进行异或运算,伪随机序列的第2个比特位与码块或传送帧的第2个比特位进行异或运算,依次类推。

伪随机化的逻辑框图如图 22 所示。伪随机序列发生器开始输出第 1 个比特位之前,反馈移位寄存器的各级初始化为全“1”状态。

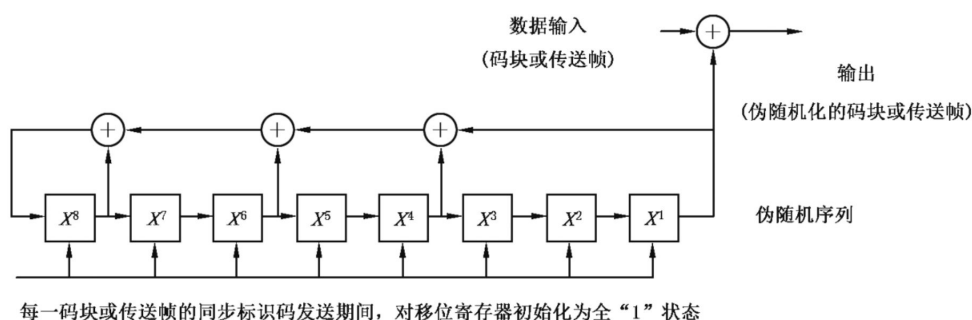


图 22 伪随机化逻辑框图

12 传送帧长度

遥测数据链路协议和高级在轨数据链路协议中都规定了传送帧的长度,传送帧长度的选择与所选的编码类型相关。一般传送帧的长度在某次任务阶段一旦选定后,其值即为一个固定值。

对应不同编码方式,传送帧长度的最大设定值为:

- a) 不使用信道编码时,其最大值为 2 048 字节。
- b) 选择卷积码时,其最大值为 2 048 字节。
- c) 选择 RS 码时,一般传送帧长度可用公式(14)来计算得出,如果考虑 32 位运行系统的兼容性,传送帧长度可用公式(14)计算得出,也可以为 4 倍的 $(255-q)I$ 。

$$L = (255 - 2E - q)I \quad \dots\dots\dots (14)$$

式中:

L ——传送帧长度;

E ——纠错能力;

Q ——RS 码块中的虚拟填充字节数值;

I ——交织深度。

- d) 选择以卷积码为内码,以 RS 码为外码的级联码时,帧长度值与 RS 码相同。
- e) 选择 Turbo 码时,传送帧长度应与所选的 Turbo 码的码块长度相匹配,本标准推荐的可选的码块长度为 223,446,892 和 1 115 四种选择。
- f) LDPC 码作为传送帧时,其最大值应符合如下设定:
 - 使用编码比率为 7/8 的 LDPC 时,其值为 892 字节;
 - 使用编码比率为 1/2、2/3 或 4/5 的 LDPC 时,其值可设为 128 字节、512 字节或者 2 048 字节中的一种。
- g) LDPC 码作为 SMTF 数据流时,其最大值为 2 048 字节。

13 管理参数

13.1 概述

为了保护空间链路的带宽,同步和信道编码相关的参数推荐由管理者控制,而不是由线路上的通信协议直接处理。这些参数是一些长时间保持不变的参数,改变这些参数一般表示与特定任务相关的协议实体的主要配置发生了改变。通过使用管理系统,管理者可以将需要的信息传递给协议实体。

13.2 编码选择

遥测信道编码方法可以采用下述任一方式:

- a) 不使用信道编码;
- b) 卷积码;
- c) RS 码;
- d) 以卷积码为内码,以 RS 码为外码的级联码;
- e) Turbo 码;
- f) LDPC 码作为传送帧;
- g) LDPC 码作为 SMTF 数据流。

若卷积码单独使用时,一旦信道传输差错超出了卷积码的纠错能力,那么在译码器的输出端就会产生一连串差错。因此在这种情况下一般应采用 CRC 检错码,并将 CRC 的 16 bit 校验字作为遥测传送帧的最后两个字节,以便在接收端确认各帧接收数据的正确性。

若在系统中没有使用信道编码,也应对遥测传送帧采用 CRC 检错码,以便在接收端确认各帧接收数据的正确性。

关于 CRC 检错应用的进一步信息参见附录 G。

13.3 参数设置

各种类型遥测信道编码的参数管理见表 10～表 14，卷积码的管理参数及其设置见表 10，RS 码的管理参数及其设置见表 11，Turbo 码的管理参数及其设置见表 12，LDPC 码的管理参数及其设置见表 13，帧同步的管理参数及其设置见表 14。

表 10 卷积码的管理参数表

管理参数	设置
编码码率(r)	1/2, 2/3, 3/4, 7/8

表 11 RS 码的管理参数表

管理参数	设置
差错检测能力(E , symbols)	8, 16
交织深度(I)	1, 2, 3, 4, 5, 8
虚拟填充长度(Q , symbols)	整数型

表 12 Turbo 码的管理参数表

管理参数	设置
编码码率(r)	1/2, 1/3, 1/4, 1/6
信息块长度 $[k, \text{bits}]$	1 784, 3 568, 7 136, 8 920

表 13 LDPC 码的管理参数表

管理参数	设置
码率(r)	1/2, 2/3, 4/5, 7/8
码片长度(如信息块长度 $[k, \text{bits}]$)	1 024, 4 096, 16 384 (if $r=1/2, 2/3$, or $4/5$), 7 136 (if $r=7/8$)
LDPC 码块数(码字)	$m = 1, 2, 3, 4, 5, 6, 7, 8$

表 14 帧同步的管理参数表

管理参数	设置
传送帧长度(bit)	整数型

附 录 A
(资料性附录)

本标准与 ISO 22641:2012 相比的结构变化情况

本标准与 ISO 22641:2012 相比在结构上有较多调整,具体章条编号对照情况见表 A.1。

表 A.1 本标准与 ISO 22641:2012 的章条编号对照情况

本标准章条编号	对应的 ISO 22641:2012 章条编号
1	1.1,1.2
—	1.3~1.7
2	1.8
3	附录 D
4.1	2.1
4.2	2.2
4.3	2.3
5.1	3.1
5.2	3.2
5.3	3.3
5.4	3.4
6.1	4.1
6.2	4.2
6.3	4.3
6.4	4.4
7.1	5.1
7.2	5.2
7.3	—
8.1	6.1
8.2	6.2
8.3	6.3
8.4	—
9.1	7.1
9.2	7.2
9.3	7.3
9.4	7.4
9.5	—
10.1	8.1
10.2	8.2

表 A.1 (续)

本标准章条编号	对应的 ISO 22641:2012 章条编号
10.3	8.3
10.4	8.4
10.5	8.5
10.6	8.6
11.1	9.1
11.2	9.2
11.3	9.3
11.4	9.4,9.5
12	10.1~10.8
13.1	11.1
13.2	11.2,11.3
13.3	11.4~11.8
附录 A	—
附录 B	—
附录 C	附录 A
附录 D	附录 G
附录 E	附录 F
附录 F	附录 C
附录 G	—
—	附录 B、附录 E

附 录 B
(资料性附录)

本标准与 ISO 22641:2012 的技术性差异及其原因

表 B.1 给出了本标准与 ISO 22641:2012 的技术性差异及原因。

表 B.1 本标准与 ISO 22641:2012 的技术性差异及其原因

本标准章条编号	技术性差异	原因
1	删除了 1.3,1.4,1.5,1.6,1.7 中本标准不规定的具体内容	对于不规定的内容只在标准的编制说明中进行了说明
2	关于规范性引用文件,本标准做了具有技术性差异的调整,调整的情况集中反映在第 2 章“规范性引用文件”中,具体调整如下: ——增加引用了 GB/T 2900.54(IEC 60050-713:1998,IDT)(见第 3 章); ——用修改采用国际标准的 GB/T 39345 代替 ISO 22666:2016(见 4.2.3、5.2.3、8.2.2); ——用修改采用国际标准的 GB/T 39349 代替 ISO 22642:2015(见 9.3.1); ——用修改采用国际标准的 GB/T 39351 代替 ISO 22645:2016(见 4.2.3、5.2.3、8.2.2)	强调采用国际标准时按照 GB/T 1.1 的规定编写,确保技术内容的确定和文本结构的协调统一,适应我国的标准编写和版式要求
3	补充了本标准需要的术语和定义	国际标准中的附录 D 给出了本标准需要用到的术语,只是给出了从理解本标准角度,需要了解术语词条,并没有给出具体定义。从便于本标准理解的角度,应给出本标准直接应用到的术语及其定义
4.2.2	删除了国际标准中注释部分以及正文中说明解释性的内容	按照我国标准的编写要求与版式,不属于规范性内容,因此删除
5.2.2	将国际标准中注释部分的调整为正文的内容,并用列项进行分类描述	按照我国标准编写要求,本部分内容包括很多要求性和原则性的技术内容,因此直接放入正文中
6.2.1,6.2.2	将国际标准中注释的内容调整为正文的内容	注释的内容为介绍图 1 的具体信息,属于规范性内容,因此做了调整
6.3.1	补充了 RS 码特性参数的信息说明	为了加深理解,以及具体实现的可操作性,便于标准的执行
6.3.7	删除了国际标准 4.3.7 中注释部分的内容	按照我国标准的编写要求与版式,不属于规范性内容,因此删除
6.3.9.2,6.3.9.3	补充了两种坐标基表征的基础信息	为了确保内容全面理解,以及具体实现的可操作性,便于标准的执行

表 B.1 (续)

本标准章条编号	技术性差异	原因
6.3.10	将国际标准中注释的内容调整为正文的内容	按照我国标准编写要求,本部分内容包括很多要求性和原则性的技术内容,因此直接放入正文中
6.4	将原国际标准中 4.4 讨论的内容中规范性内容进行了保留,探讨以及说明性文字进行了删除	为了确保内容全面理解,以及具体实现的可操作性,对内容进行了调整,便于标准的执行
7.3	增加了级联码的物理实现原理框图技术内容	增加可操作性,便于标准的执行
8.2.1,8.2.2	删除了国际标准中注释部分 6.2.1 和 6.2.2 的内容	按照我国标准的编写要求与版式,不属于规范性内容,因此删除
8.3	将国际标准中注释部分有要求的内容调整为正文的内容,说明性文字进行了删除	注释的内容为介绍图 2 的具体信息,属于规范性内容,因此做了调整
8.4	将国际标准中的 6.3 注释部分中关于 Turbo 码同步标识相关的内容作为新的一条	方便技术内容的理解,增加可操作性,便于标准的执行
9.3.3	将国际标准中的 7.3.2 和 7.3.3 的内容进行整合成一条	为了确保内容全面理解,以及具体实现的可操作性,对内容进行了调整,便于标准的执行
9.3.4	将国际标准中的 7.3.4 和 7.3.5 的内容进行整合成一条	为了确保内容全面理解,以及具体实现的可操作性,对内容进行了调整,便于标准的执行
9.4.3	将国际标准的 7.4 注释性内容中规定的生成矩阵的技术内容作为单独一条进行规定	方便技术内容的理解,增加可操作性,也便于标准的执行
9.5	为了本条内容的完整性,将国际标准中第 10 章描述的 LDPC 码的同步标识信息单独作为一节进行规定	方便技术内容的理解,增加可操作性,也便于标准的执行
12	将国际标准中的 8 条信息用列项的形式进行描述	精简文字,便于理解
13.2	将国际标准中用表描述的信息修改为列项形式进行描述,补充了 CRC 检错码的相关信息内容	方便阅读,便于理解

附录 C

(规范性附录)

服务定义

C.1 概述

本附录以原语的形式提供了服务定义,它给出了服务提供者和用户之间对数据和控制信息进行逻辑交换的抽象模型。原语的定义与特定的实现方法无关。

原语参数以抽象的意义定义,它规定了原语用户可利用的信息。本标准并不限定获取这些信息的特定实现方法。除了本附录规定的参数外,一个实现也可能对服务用户提供其他参数,例如控制服务的参数、监视性能的参数、利用诊断的参数等。

C.2 服务综述

遥测同步和信道编码对固定长度的遥测和高级在轨数据传送帧,以恒定的帧速率在空间链路的物理信道上提供单向的传输服务,检错和纠错功能可选。

一个物理信道上只有一个用户能使用该服务,而不同用户的传送帧不能一起在一个物理信道内多路复用。

C.3 服务参数

C.3.1 帧

帧参数是本服务的服务数据单元。TM 传送帧在 GB/T 39351 中定义,AOS 传送帧在 GB/T 39345 中定义。一个物理信道上的任一个传送帧的长度应是相同的,并由管理建立。

C.3.2 质量指示

质量指示参数用于通知接收端的服务用户:接收的传送帧中存在一个无法纠正的错误。

C.3.3 顺序指示

顺序指示参数用于通知接收端的服务用户:物理信道上的一个或多个传送帧由于帧同步失锁而丢失。

C.4 服务原语

C.4.1 ChannelAccess.request 信道访问请求原语

C.4.1.1 功能:由发送端的服务用户向服务提供者传递的,它要求将一个传送帧通过物理信道传送到接收端的用户。

C.4.1.2 语义:提供 1 个参数:ChannelAccess.request(帧)。

C.4.1.3 产生时机:当需要处理和发送传送帧时,将 ChannelAccess.request 原语提交给服务提供者。

C.4.1.4 接受的影响:受到 ChannelAccess.request 原语,使得服务提供者执行 4.3.1 描述的功能,并传

送所产生的信道符号。

C.4.2 ChannelAccess.indication 信道访问指示原语

C.4.2.1 功能:ChannelAccess.indication 是这项服务的服务指示原语。

C.4.2.2 语义:提供 1 个参数:ChannelAccess.indication(帧,质量指示,顺序指示)。

C.4.2.3 产生时机:ChannelAccess.indication 原语由服务提供者提交给接收端的服务用户,以交付一个帧。

C.4.2.4 接收效果:服务用户接收到 ChannelAccess.indication 时的响应没有定义。

附录 D
(资料性附录)

RS 码生成多项式系数的展开式

为了有助于实现 RS 码系统,本附录给出了 RS 码生成多项式系数 G_i ,其中表 D.1 给出了 $E=16$ 的 RS 码生成多项式的系数展开式;表 D.2 给出了 $E=8$ 的 RS 码生成多项式的系数展开式。

表 D.1 $E=16$ 的 RS 码的生成多项式系数

生成多项式 $G(x)$ 的系数 G_i	G_i 的 α 多项式							
	α^7	α^6	α^5	α^4	α^3	α^2	α^1	α^0
$G_0 = G_{32} = \alpha^0$	0	0	0	0	0	0	0	1
$G_1 = G_{31} = \alpha^{249}$	0	1	0	1	1	0	1	1
$G_2 = G_{30} = \alpha^{59}$	0	1	1	1	1	1	1	1
$G_3 = G_{29} = \alpha^{66}$	0	1	0	1	0	1	1	0
$G_4 = G_{28} = \alpha^4$	0	0	0	1	0	0	0	0
$G_5 = G_{27} = \alpha^{43}$	0	0	0	1	1	1	1	0
$G_6 = G_{26} = \alpha^{126}$	0	0	0	0	1	1	0	1
$G_7 = G_{25} = \alpha^{251}$	1	1	1	0	1	0	1	1
$G_8 = G_{24} = \alpha^{97}$	0	1	1	0	0	0	0	1
$G_9 = G_{23} = \alpha^{30}$	1	0	1	0	0	1	0	1
$G_{10} = G_{22} = \alpha^3$	0	0	0	0	1	0	0	0
$G_{11} = G_{21} = \alpha^{213}$	0	0	1	0	1	0	1	0
$G_{12} = G_{20} = \alpha^{50}$	0	0	1	1	0	1	1	0
$G_{13} = G_{19} = \alpha^{66}$	0	1	0	1	0	1	1	0
$G_{14} = G_{18} = \alpha^{170}$	1	0	1	0	1	0	1	1
$G_{15} = G_{17} = \alpha^5$	0	0	1	0	0	0	0	0
$G_{16} = \alpha^{24}$	0	1	1	1	0	0	0	1
注: $G_3 = G_{29} = G_{13} = G_{19}$								

表 D.2 $E=8$ 的 RS 码的生成多项式系数

生成多项式 $G(x)$ 的系数 G_i	G_i 的 α 多项式							
	α^7	α^6	α^5	α^4	α^3	α^2	α^1	α^0
$G_0 = G_{16} = \alpha^0$	0	0	0	0	0	0	0	1
$G_1 = G_{15} = \alpha^{30}$	1	0	1	0	0	1	0	1
$G_2 = G_{14} = \alpha^{230}$	0	1	1	0	1	0	0	1
$G_3 = G_{13} = \alpha^{49}$	0	0	0	1	1	0	1	1
$G_4 = G_{12} = \alpha^{235}$	1	0	0	1	1	1	1	1
$G_5 = G_{11} = \alpha^{129}$	0	1	1	0	1	0	0	0
$G_6 = G_{10} = \alpha^{81}$	1	0	0	1	1	0	0	0
$G_7 = G_9 = \alpha^{76}$	0	1	1	0	0	1	0	1
$G_8 = \alpha^{173}$	0	1	0	0	1	0	1	0

附录 E (资料性附录)

Berlekamp 表示和常规表示间的变换

E.1 等效变换

本附录给出了常规基和对偶基中符号表示间的变换,并且给出了使用两种符号表示的编码器间变换的等效性。

Berlekamp RS 码器所输入的信息符号 I 以及产生的校验符号 C 可表示为: $[z_0, z_1, \dots, z_7]$, 其中, z_i 分别为矢量 $z_0\ell_0 + z_1\ell_1 + \dots + z_7\ell_7$ 在相应对偶基的基矢量 ℓ_i 方向的系数。

常规 RS 码器所输入的信息符号 I' 以及产生的校验符号 C' 可表示为: $[u_7, u_6, \dots, u_0]$, 其中, u_j 分别为矢量 $u_7\alpha^7 + u_6\alpha^6 + \dots + u_0\alpha^0$ 在相应常规基的基矢量 α^j 方向的系数。

由于 RS 码器输出的信道符号定义为对偶基中符号(8 bit 矢量 $[V_0, V_1, \dots, V_7]$), 因此当使用常规编码器完成本部分所规定的编码时, 就须分别在常规编码器的前、后进行相应的符号表达式变换。如图 E.1 所示。

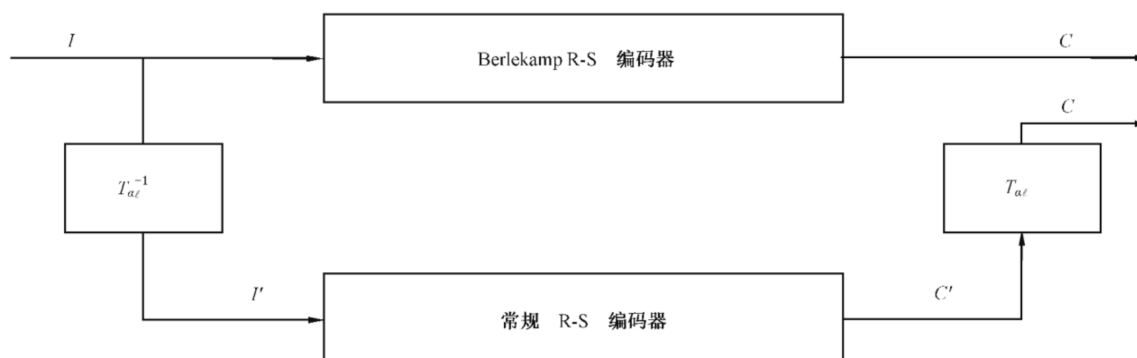


图 E.1 常规编码器与 Berlekamp 编码器间的等效变换关系

E.2 具体说明

假定常规的和 Berlekamp 类型的 $(255, k)$ RS 码器具有相同的自反多项式。与常规编码器有关的符号以“ α ”的多项式形式的表达式; 对“ α ”的每一个多项式也都存在一个与 Berlekamp 编码器的符号有关的对偶基上的表达式, 见表 E.1。两种坐标基上各符号元素相对应的矢量表达式为公式(E.1)。

给定:

$$\alpha^i = u_7\alpha^7 + u_6\alpha^6 + \dots + u_0 \quad \dots\dots\dots (E.1)$$

其中, $(0 \leq i < 255) \cup (i = *)$

若相应的对偶基的元素为: $z_0\ell_0 + z_1\ell_1 + \dots + z_7\ell_7$, 则为公式(E.2)

$$[z_0, z_1, \dots, z_7] = [u_7, u_6, \dots, u_0] T_{\alpha\ell} \quad \dots\dots\dots (E.2)$$

并且 $T_{\alpha\ell}$ 可用公式(E.3)表示:

$$T_{ae} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \dots\dots\dots (E.3)$$

其中 T_{ae} 的第 1 行、第 2 行、直到第 8 行分别表示常规基矢量 $\alpha^7, \alpha^6, \dots, \alpha^0$ 在对偶基中的对应表达式。

T_{ae} 的逆矩阵见公式(E.4)：

$$T_{ae}^{-1} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \dots\dots\dots (E.4)$$

其中 T_{ae}^{-1} 的第 1 行、第 2 行、直到第 8 行分别表示对偶基矢量 $\ell_0, \ell_1, \dots, \ell_7$ 在常规基中的对应表达式。

这样 $[z_0, z_1, \dots, z_7] \cdot T_{ae}^{-1} = [u_7, u_6, \dots, u_0]$ 。

示例 1：给定信息符号 $I = [z_0, z_1, \dots, z_7] = [10111001]$

$$\text{则: } I' = [u_7, u_6, \dots, u_0] = I \cdot T_{ae}^{-1} = [10111001] \begin{bmatrix} 11000101 \\ 00101010 \\ 00101110 \\ 11111101 \\ 11110000 \\ 01111001 \\ 10101100 \\ 11001100 \end{bmatrix} = [00101010]$$

示例 2：给定校验符号 $C' = [\alpha_7, \alpha_6, \dots, \alpha_0] = [01011001] = \alpha^{152}$

$$\text{则: } C = [z_0, z_1, \dots, z_7] = C' T_{ae} = [01011001] \begin{bmatrix} 10001101 \\ 11101111 \\ 11101100 \\ 10000110 \\ 11111010 \\ 10011001 \\ 10101111 \\ 01111011 \end{bmatrix} = [11101000]$$

表 E.1 常规基和对偶基符号对照表

α 的 幂次	常规基表达式 α 的多项式	Berlekamp 表示法 $\ell_{0,1,2,3,4,5,6,7}$	α 的 幂次	常规基表达式 α 的多项式	Berlekamp 表示法 $\ell_{0,1,2,3,4,5,6,7}$
*	00000000	00000000	32	00011101	10011110
0	00000001	01111011	33	00111010	00111111
1	00000010	10101111	34	01110100	00011100
2	00000100	10011001	35	11101000	01110100
3	00001000	11111010	36	01010111	00100100
4	00010000	10000110	37	10101110	10101101
5	00100000	11101100	38	11011011	11001010
6	10100000	11101111	39	00110001	00010001
7	10000000	10001101	40	01100010	10101100
8	10000111	11000000	41	11000100	11111011
9	10001001	00001100	42	00001111	10110111
10	10010101	11101001	43	00011110	01001010
11	10101101	01111001	44	00111100	00001001
12	11011101	11111100	45	01111000	01111111
13	00111101	01110010	46	11110000	00001000
14	01111010	11010000	47	01100111	01001110
15	11110100	10010001	48	11001110	10101110
16	01101111	10110100	49	00011011	10101000
17	11011110	00101000	50	00110110	01011100
18	00111011	01000100	51	01101100	01100000
19	01110110	10110011	52	11011000	00011110
20	11101100	11101101	53	00110111	00100111
21	01011111	11011110	54	01101110	11001111
22	10111110	00101011	55	11011100	10000111
23	11111011	00100110	56	00111111	11011101
24	01110001	11111110	57	01111110	01001001
25	11100010	00100001	58	11111100	01101011
26	01000011	00111011	59	01111111	00110010
27	10000110	10111011	60	11111110	11000100
28	10001011	10100011	61	01111011	10101011
29	10010001	01110000	62	11110110	00111110
30	10100101	10000011	63	01101011	00101101
31	11001101	01111010	64	11010110	11010010

表 E.1 (续)

α 的 幂次	常规基表达式 α 的多项式	Berlekamp 表示法 $\ell_{0,1,2,3,4,5,6,7}$	α 的 幂次	常规基表达式 α 的多项式	Berlekamp 表示法 $\ell_{0,1,2,3,4,5,6,7}$
65	00101011	11000010	98	11000010	11001101
66	01010110	01011111	99	00000011	11010100
67	10101100	<u>00000010</u>	100	00000110	00110110
68	11011111	01010011	101	00001100	01100011
69	00111001	11101011	102	00011000	01111100
70	01110010	00101010	103	00110000	01101010
71	11100100	00010111	104	01100000	00000011
72	01001111	01011000	105	11000000	01100010
73	10011110	11000111	106	00000111	01001101
74	10111011	11001001	107	00001110	11001100
75	11110001	01110011	108	00011100	11100101
76	01100101	11100001	109	00111000	10010000
77	11001010	00110111	110	01110000	10000101
78	00010011	01010010	111	11100000	10001110
79	00100110	11011010	112	01000111	10100010
80	01001100	10001100	113	10001110	01000001
81	10011000	11110001	114	10011011	00100101
82	10110111	10101010	115	10110001	10011100
83	11101001	00001111	116	11100101	01101100
84	01010101	10001011	117	01001101	11110111
85	10101010	00110100	118	10011010	01011110
86	11010011	00110000	119	10110011	00110011
87	00100001	10010111	120	11100001	11110101
88	01000010	<u>01000000</u>	121	01000101	00001101
89	10000100	00010100	122	10001010	11011000
90	10001111	00111010	123	10010011	11011111
91	10011001	10001010	124	10100001	00011010
92	10110101	00000101	<u>125</u>	11000101	<u>10000000</u>
93	11101101	10010110	126	00001101	00011000
94	01011101	01110001	127	00011010	11010011
95	10111010	10110010	128	00110100	11110011
96	11110011	11011100	129	01101000	11111001
97	01100001	01111000	130	11010000	11100100

表 E.1 (续)

α 的 幂次	常规基表达式 α 的多项式	Berlekamp 表示法 $\ell_{0,1,2,3,4,5,6,7}$	α 的 幂次	常规基表达式 α 的多项式	Berlekamp 表示法 $\ell_{0,1,2,3,4,5,6,7}$
131	00100111	10100001	164	01111101	10011101
132	01001110	00100011	165	11111010	01011101
133	10011100	01101000	166	01110011	01010001
134	10111111	01010000	167	11100110	10111000
135	11111001	10001001	168	01001011	11000001
136	01110101	01100111	169	10010110	00111101
137	11101010	11011011	170	10101011	01001111
138	01010011	10111101	171	11010001	10011111
139	10100110	01010111	172	00100101	00001110
140	11001011	01001100	173	01001010	10111010
141	00010001	11111101	174	10010100	10010010
142	00100010	01000011	175	10101111	11010110
143	01000100	01110110	176	11011001	01100101
144	10001000	01110111	177	00110101	10001000
145	10010111	01000110	178	01101010	01010110
146	10101001	11100000	179	11010100	01111101
147	11010101	00000110	180	00101111	01011011
148	00101101	11110100	181	01011110	10100101
149	01011010	00111100	182	10111100	10000100
150	10110100	01111110	183	11111111	10111111
151	11101111	00111001	184	01111001	<u>00000100</u>
152	01011001	11101000	185	11110010	10100111
153	10110010	01001000	186	01100011	11010111
154	11100011	01011010	187	11000110	01010100
155	01000001	10010100	188	00001011	00101110
156	10000010	00100010	189	00010110	10110000
157	10000011	01011001	190	00101100	10001111
158	10000001	11110110	191	01011000	10010011
159	10000101	01101111	192	10110000	11100111
160	10001101	10010101	193	11100111	11000011
161	10011101	00010011	194	01001001	01101110
162	10111101	11111111	195	10010010	10100100
<u>163</u>	11111101	<u>00010000</u>	196	10100011	10110101

表 E.1 (续)

α 的 幂次	常规基表达式 α 的多项式	Berlekamp 表示法 $\ell_{0,1,2,3,4,5,6,7}$	α 的 幂次	常规基表达式 α 的多项式	Berlekamp 表示法 $\ell_{0,1,2,3,4,5,6,7}$
197	11000001	00011001	<u>226</u>	00101110	<u>00100000</u>
198	00000101	11100010	227	01011100	00001010
199	00001010	01010101	228	10111000	00011101
200	00010100	00011111	229	11110111	01000101
201	00101000	00010110	230	01101001	10000010
202	01010000	01101001	231	11010010	01001011
203	10100000	01100001	232	00100011	00111000
204	11000111	00101111	233	01000110	11011001
205	00001001	10000001	234	10001100	11101110
206	00010010	00101001	235	10011111	10111110
207	00100100	01110101	236	10111001	01100110
208	01001000	00010101	237	11110101	11101010
209	10010000	00001011	238	01101101	00011011
210	10100111	00101100	239	11011010	10110001
211	11001001	11100011	240	00110011	10111110
212	00010101	01100100	241	01100110	00110101
213	00101010	10111001	<u>242</u>	11001100	<u>00000001</u>
214	01010100	11110000	243	00011111	00110001
215	10101000	10011011	244	00111110	10100110
216	11010111	10101001	245	01111100	11100110
217	00101001	01101101	246	11111000	11110010
218	01010010	11000110	247	01110111	11001000
219	10100100	11111000	248	11101110	01000010
220	11001111	11010101	249	01011011	01000111
221	00011001	00000111	250	10110110	11010001
222	00110010	11000101	251	11101011	10100000
223	01100100	10011010	252	01010001	00010010
224	11001000	10011000	253	10100010	11001110
225	00010111	11001011	254	11000011	10110110
注：表中带下划线者为对偶基中的基矢量。					

附 录 F
(资料性附录)

223/255 比率的 LDPC 码

生成矩阵的循环矩阵表见表 F.1, 其中第 2 列表示每一个循环矩阵第 1 行的十六进制表示法。因为只有 511 个可能位置, 为了满足 128 个 16 进制数, 最左边的比特位需要用 0 填充。即每一个循环矩阵规范中最左边的数应理解为八进制数(3 bit), 其他数字则应理解为 16 进制(4 bit)。与表 3 所示的校验矩阵的循环矩阵不同, 生成矩阵的循环矩阵中的“1”不是低密度的。循环矩阵的后一行相对于前一行右移 1 bit。

表 F.1 生成矩阵的循环矩阵表

循环矩阵 第 1 行标号	循环矩阵第 1 行
$b_{1,1}$	55BF56CC55283DFEEFEA8C8CFF04E1EBD9067710988E25048D67525426939E206 8D2DC6FCD2F822BEB6BD96C8A76F4932AAE9BC53AD20A2A9C86BB461E43759C
$b_{1,2}$	6855AE08698A50AA3051768793DC238544AF3FE987391021AAF6383A6503409C3 CE971A80B3ECE12363EE809A01D91204F1811123EAB867D3E40E8C652585D28
$b_{2,1}$	62B21CF0AEE0649FA67B7D0EA6551C1CD194CA77501E0FCF8C85867B9CF679C18 CF7939E10F8550661848A4E0A9E9EDB7DAB9EDABA18C168C8E28AACDDEAB1E
$b_{2,2}$	64B71F486AD57125660C4512247B229F0017BA649C6C11148FB00B70808286F1A 9790748D296A593FA4FD2C6D7AAF7750F0C71B31AEE5B400C7F5D73AAF00710
$b_{3,1}$	681A8E51420BD8294ECE13E491D618083FFBBA830DB5FAF330209877D801F92B5 E07117C57E75F6F0D873B3E520F21EAFD78C1612C6228111A369D5790F5929A
$b_{3,2}$	04DF1DD77F1C20C1FB570D7DD7A1219EAECEA4B2877282651B0FFE713DF338A63263BC0E324A 87E2DC1AD64C9F10AAA585ED6905946EE167A73CF04AD2AF9218
$b_{4,1}$	35951FEE6F20C902296C9488003345E6C5526C5519230454C556B8A04FC0DC642 D682D94B4594B5197037DF15B5817B26F16D0A3302C09383412822F6D2B234E
$b_{4,2}$	7681CF7F278380E28F1262B22F40BF3405BFB92311A8A34D084C086464777431D BFDD2E82A2E6742BAD6533B51B2BDEE0377E9F6E63DCA0B0F1DF97E73D5CD8
$b_{5,1}$	188157AE41830744BAE0ADA6295E08B79A44081E111F69BBE7831D07BEEBF7623 2E065F752D4F218D39B6C5BF20AE5B8FF172A7F1F680E6BF5AAC3C4343736C2
$b_{5,2}$	5D80A6007C175B5C0DD88A442440E2C29C6A136BBCE0D95A58A83B48CA0E7474E9476C92E33D 164BFF943A61CE1031DFF441B0B175209B498394F4794644392E
$b_{6,1}$	60CD1F1C282A1612657E8C7C1420332CA245C0756F78744C807966C3E13264388 78BD2CCC83388415A612705AB192B3512EEF0D95248F7B73E5B0F412BF76DB4
$b_{6,2}$	434B697B98C9F3E48502C8DBD891D0A0386996146DEBEF11D4B833033E05EDC28 F808F25E8F314135E6675B7608B66F7FF3392308242930025DDC4BB65CD7B6E

表 F.1 (续)

循环矩阵 第 1 行标号	循环矩阵第 1 行
$b_{7,1}$	766855125CFDC804DAF8DBE3660E8686420230ED4E049DF11D82E357C54FE256E A01F5681D95544C7A1E32B7C30A8E6CF5D0869E754FFDE6AEFA6D7BE8F1B148
$b_{7,2}$	222975D325A487FE560A6D146311578D9C5501D28BC0A1FB48C9BDA173E869133 A3AA9506C42AE9F466E85611FC5F8F74E439638D66D2F00C682987A96D8887C
$b_{8,1}$	14B5F98E8D55FC8E9B4EE453C6963E052147A857AC1E08675D99A308E7269FAC5 600D7B155DE8CB1BAC786F45B46B523073692DE745FDF10724DDA38FD093B1C
$b_{8,2}$	1B71AFFB8117BCF8B5D002A99FEEA49503C0359B056963FE5271140E626F6F8FC E9F29B37047F9CA89EBCE760405C6277F329065DF21AB3B779AB3E8C8955400
$b_{9,1}$	0008B4E899E5F7E692BDCE69CE3FAD997183CFAEB2785D0C3D9CAE510316D4BD65A2A06CBA7 F4E4C4A80839ACA81012343648EEA8DBBA2464A68E115AB3F4034
$b_{9,2}$	5B7FE6808A10EA42FEF0ED9B41920F82023085C106FBBC1F56B567A14257021BC 5FDA60CBA05B08FAD6DC3B0410295884C7CCDE0E56347D649DE6DDCEEBC0C95E
$b_{10,1}$	5E9B2B33EF82D0E64AA2226D6A0ADCD179D5932EE1CF401B336449D0FF775754C A56650716E61A43F963D59865C7F017F53830514306649822CAA72C152F6EB2
$b_{10,2}$	2CD8140C8A37DE0D0261259F63AA2A420A8F81FECB661DBA5C62DF6C817B4A61 2BC1F068A50DFD0EA8FE1BD387601062E2276A4987A19A70B460C54F215E184
$b_{11,1}$	06F1FF249192F2EAF063488E267EEE994E7760995C4FA6FFA0E4241825A7F5B65 C74FB16AC4C891BC008D33AD4FF97523EE5BD14126916E0502FF2F8E4A07FC2
$b_{11,2}$	65287840D00243278F41CE1156D1868F24E02F91D3A1886ACE906CE741662B40B 4EFDFB90F76C1ADD884D920AFA8B3427EEB84A759FA02E00635743F50B942F0
$b_{12,1}$	4109DA2A24E41B1F375645229981D4B7E88C36A12DAB64E91C764CC43CCEC188E C8C5855C8FF488BB91003602BEF43DBEC4A621048906A2CDC5DBD4103431DB8
$b_{12,2}$	2185E3BC7076BA51AAD6B199C8C60BCD70E8245B874927136E6D8DD527DF0693DC10A1C8E51B 5BE93FF7538FA138B335738F4315361ABF8C73BF40593AE22BE4
$b_{13,1}$	228845775A262505B47288E065B23B4A6D78AFBDDDB2356B392C692EF56A35AB4A A27767DE72F058C6484457C95A8CCDD0EF225ABA56B7657B7F0E947DC17F972
$b_{13,2}$	2630C6F79878E50CF5ABD353A6ED80BEACC7169179EA57435E44411BC7D566136 DFA983019F3443DE8E4C60940BC4E31DCEAD514D755AF95A622585D69572692
$b_{14,1}$	7273E8342918E097B1C1F5FEF32A150AEF5E11184782B5BD5A1D8071E94578B0A C722D7BF49E8C78D391294371FFBA7B88FABF8CC03A62B940CE60D669DFB7B6
$b_{14,2}$	087EA12042793307045B283D7305E93D8F74725034E77D25D3FF043ADC5F8B5B1 86DB70A968A816835EFB575952EAE7EA4E76DF0D5F097590E1A2A978025573E

附 录 G

(资料性附录)

循环冗余(CRC)检错码应用

G.1 概述

本附录介绍了 CCSDS 建议书中有关遥测传送帧差错控制方面使用 CRC(Cyclic Redundancy Code)检错码时的编码和译码方案。

下面所介绍的 CRC 检错码是一种二元分组码,它用于检测码块中的传输差错。其检错能力如下:

- a) 能检测出含有奇数个比特位差错的所有错误序列;
- b) 能检出码块中随机分布的 2 bit 差错;
- c) 对码块中大于或等于 4 bit 的随机差错,其不能检出的概率约 2^{-15} (或 3×10^{-5});
- d) 能检出长度不大于 16 bit 的单个突发差错。

G.2 编译码程序

G.2.1 编码方式

编码程序接受一个长度为 $(n-16)$ bit 的数据块,然后将经过运算所得到的一个 16 bit 的帧校验序列 FCS(Frame Check Sequence)附加在数据块之后,这样就生成了一个 $(n, n-16)$ 的二元分组码。16 bit 的帧校验序列 FCS 也可以插入在传送帧尾部的帧差错控制字的位置上。

FCS 帧校验序列的运算见公式(G.1)~公式(G.3)

$$\text{FCS} = [X^{16} \cdot M(X) \oplus X^{(n-16)} \cdot L(X)] \cdot \text{modulo} \cdot G(X) \dots\dots\dots (\text{G.1})$$

$$LX = \sum_{i=0}^{15} X^i \quad (15 \text{ 阶全 '1' 多项式}) \quad \dots\dots\dots (\text{G.2})$$

$$G(X) = X^{16} + X^{12} + X^5 + 1 \quad \dots\dots\dots (\text{G.3})$$

公式(G.1)~公式(G.3)中:

FCS —— 帧校验序列;

X —— 多项式变量;

$M(X)$ —— 以二进制数为系数的 X 的多项式,它用以表示待编码的 $(n-16)$ bit 的消息;

\oplus —— 模二加法器(异或);

n —— 分组码中的总比特位数;

$L(X)$ —— 预置多项式;

$G(X)$ —— 生成多项式。

注: $X^{n-16} \cdot L(X)$ 所起的作用是:在编码开始之前,将编码器的移位寄存器预置为全 '1' 状态。这一点是与一般的循环分组编码器的不同之处。

G.2.2 译码方式

差错检测的伴随式 $S(X)$ 如公式(G.4)所示:

$$S(X) = [X^{16} \cdot C^*(X) \oplus X^n \cdot L(X)] \cdot \text{modulo} \cdot G(X) \dots\dots\dots (\text{G.4})$$

式中:

$S(X)$ —— 伴随多项式。如果接收码块中没有检出差错时 $S(X)$ 为 0;而如果接收码块中有差错检出时 $S(X)$ 应为非 0;

GB/T 39348—2020

$C^*(X)$ ——以多项式表示的接收码块。

G.3 CRC 编译码器的一种实现方法

G.3.1 编码器

图 G.1 给出了一种利用移位寄存器实现的编码器简图。编码开始时,将移位寄存器初始化为全“1”状态,门 A 和门 B 接通,门 C 断开。然后移位时钟控制 $(n-16)$ 个数据比特位顺序移入编码器,与此同时 $(n-16)$ 个数据比特位出现在编码器输出端。 $(n-16)$ 个数据比特位全部移入之后,门 A 的输出被嵌位到 0,门 B 禁止,门 C 接通,移位时钟再驱动移位寄存器中的 16 比特校验位顺序移出,并出现在编码器的输出端,这样就完成了 $(n, n-16)$ 的 CRC 编码。

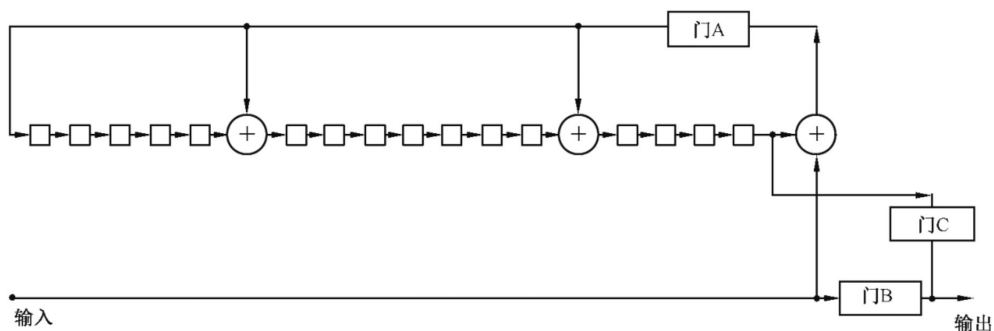


图 G.1 CRC 编码器简图

G.3.2 译码器

图 G.2 给出了一种利用移位寄存器实现的译码器简图。译码开始时,将移位寄存器初始化为全“1”状态,门 B 接通。然后移位时钟控制接收块的 $n \text{ bit}[(n-16)$ 个数据比特位以及 16 个校验比特位] 顺序移入移码器,当第 $(n-16)$ 个数据比特位移入之后,全部数据已出现在译码器输出端,这时将门 B 禁止,16 效验比特位再顺序移入移位寄存器。最后检查移位寄存器的内容,若接收块没有差错,移位寄存器的内容应为全“0”状态;否则为非全“0”状态。

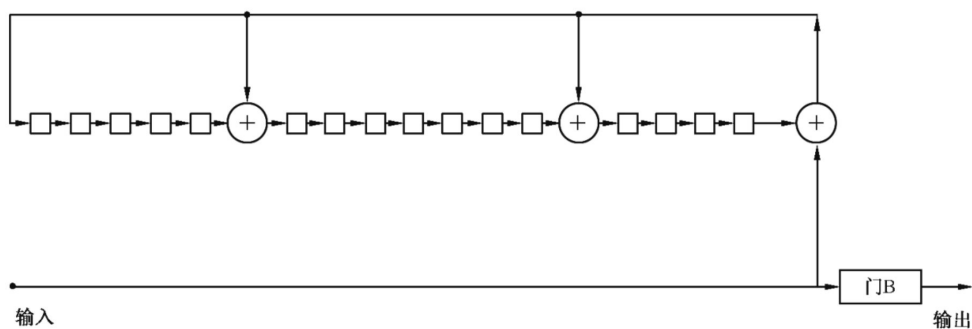


图 G.2 CRC 译码器简图



GB/T 39348-2020

版权专有 侵权必究

*

书号:155066 · 1-65971