ANALOG DEVICES

AN-1292
应用笔记

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

里德-所罗门前向纠错和ADF7023

作者：Stephen Hinchy和Kalim Khan

简介

本应用笔记描述供ADF7023收发器使用的里德-所罗门(RS)固件模块。该固件模块包含RS前向纠错和高级加密标准(AES)加密文件**rom_ram_7023_2_2_RS_AES.dat**，可从下列FTP服务器下载：ftp://ftp.analog.com/pub/RFL/FirmwareModules/ADF7023。

在里德-所罗门编码中，校验符号附加到传输数据。接收到数据后，用这些符号检测是否存在错误并对接收数据进行纠错。里德-所罗门码以RS(n, k)表示，n和k如图1所示。可通过RS解码纠错的字节数为t = (n − k)/2。

ADF7023固件模块可灵活部署RS编码与解码，用户可以选择n和k的数值，实现数据包中最多5字节内容的纠错。RS编码数据包对突发错误和随机错误均具有较高的抵抗力，并具有编码增益，可实现更佳的链路裕量。t = 5时，不同载荷长度下的RS码率如表1所示。

表1. RS码率

编码字节数(n)	数据字节数(k)	开销 (m = n − k)	可纠错字节数 (t = m/2)	码率 (R = k/n)
38	28	10	5	0.74
128	118	10	5	0.92
198	188	10	5	0.95

命令和数据包随机存取存储器寄存器位置

提供数据包随机存取存储器(RAM)寄存器(地址范围：0x010至0x0D5)，存储里德-所罗门编码数据。因此，可能的最大编码长度(n)为198字节。所有其他数据包RAM寄存器都由ADF7023或里德-所罗门固件予以保留。

表2显示用户必须在RS编码或解码之前进行初始化的寄存器。这些寄存器定义针对该固件模块，不适用于ADF7023的正常操作。

表2. 需在RS编码或解码之前进行初始化的寄存器位置

寄存器地址[1]	寄存器名称	说明
0x0D7	VAR_RSPCKTADR	指向用于编码的K数据字节或用于解码的n数据字节及校验符号的序列开端数据包RAM地址
0x0D8	VAR_RS_K	K值，也就是数据字节数
0x0D9	VAR_RS_N	N值，也就是数据字节加校验符号数

[1] 这些寄存器定义针对该固件模块，不适用于ADF7023的正常操作。

表3中显示初始化RS编码、执行RS编码或执行RS解码所必需的命令。有关RS编码和RS解码程序的更多信息，请参考本应用笔记的“RS程序”部分和“RS解码程序”部分。

表3. 里德-所罗门命令

命令	代码	说明
CMD_RS_ENCODE_INIT	0xD1	初始化里德-所罗门编码所需的内部变量
CMD_RS_ENCODE	0xD0	计算里德-所罗门校验字节并将其追加到包RAM中存储的发送有效载荷数据。
CMD_RS_DECODE	0xD2	对包RAM中存储的接收有效载荷数据执行里德-所罗门纠错

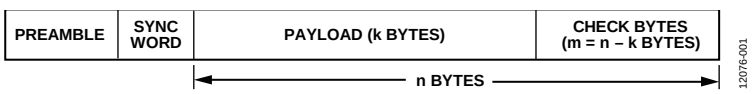

图1. 里德-所罗门编码数据包

目录

简介 1
命令和数据包随机存取存储器寄存器位置 1
修订历史 2
RS程序 3
向ADF7023写入RS固件模块 3
确定完成里德-所罗门命令的时间 3
RS编码程序 3
RS解码程序 4
发送和接收RS编码数据 5
RS编码和解码时间 6
编码增益 7

修订历史

2014年2月—修订版0：初始版

RS程序

向ADF7023写入RS固件模块

使用RS固件模块之前，用户必须将其写入ADF7023的程序RAM中。下列步骤详细解释如何向程序RAM写入RS固件模块：

1. 确保ADF7023处于PHY_OFF状态。
2. 发出CMD_RAM_LOAD_INIT命令(地址0xBF)。
3. 使用SPI存储器块写入命令向程序RAM写入模块(0x1E00 [固件模块]；参考ADF7023数据手册中的“块写入”部分)。
4. 发出CMD_RAM_LOAD_DONE命令(地址0xC7)。

固件模块将被存储到程序RAM中。

确定完成里德-所罗门命令的时间

CMD_FINISHED中断可用来确定CMD_RS_ENCODE_INIT、CMD_RS_ENCODE和CMD_RS_DECODE命令何时完成。如需使能该中断，可置位INTERRUPT_ MASK_1寄存器(地址0x101[0])中的位0 (CMD_FINISHED)。置位该屏蔽位后，ADF7023的中断引脚将在完成任何命令后置位。向INTERRUPT_SOURCE_1(地址0x337[0])中的位0写入逻辑1可清零中断。有关使用中断的更多信息，请参见ADF7023数据手册中的“中断产生”部分。

RS编码程序

在下列示例中，对28个数据字节进行编码，从而最多可纠错5个字节(RS(38, 28))。这28个字节的数据位于数据包RAM中，从地址0x0A0处开始。

```
// Initialize RS encoding.
180200 // Clear this register prior to RS encoding. (SPI_MEM_WR to set packet RAM Address 0x002
       // to Address 0x00)

// Set RS Packet Address 0x0D7.
18D7A0 // Set address of the start of the packet RAM data to encode to Address 0x0A0.

//Set value of k (number data bytes) to 28.
18D81C // k = 0x1C = 28.

// Set value of n (number of encoded bytes = k + number of check bytes).
18D926 // n = 0x26 = 38     (t = (n – k)/2 = (38 – 28)/2 = 5 = number of bytes that can be
       // corrected).

D1     // Issue CMD_RS_ENCODE_INIT to initialize the internal variables required for the
       // Reed-Solomon encoding.
       // Wait until CMD_FINISHED interrupt is generated.

// Perform an RS encode.
D0     // Issue CMD_RS_ENCODE to perform a Reed-Solomon encode of the data bytes.
       // Wait until CMD_FINISHED interrupt is generated (or until Address 0x0D6 = 0x00).
```

RS解码程序

下例中，对RS(38, 28)编码数据包进行解码，并回读错误的字节数。

```
//Initialize RS decoding.
180200 // It is necessary to clear this register prior to RS decoding.

// Set RS Packet Address 0x0D7.
18D7A0 // Set address of the start of the packet RAM data to decode 0x0A0.

//Set value of k (number data bytes) to 28.
18D81C // k = 0x1C = 28

// Set value of n (number of encoded bytes = k + number of check bytes).
18D926 // n = 0x26 = 38     (t = (n – k)/2 = (38 – 28)/2 = 5 = number of bytes that can be
       // corrected).

// Perform an RS decode. There is no need for a separate initialization command.
D2     // Issue CMD_RS_DECODE to perform a RS error correction of the received payload data in
       // packet RAM.
       // Wait until CMD_FINISHED interrupt is generated.

// Read back Address 0x0D6 to give result of the RS Decode. Refer to Table 4.
```

表4. RS解码回读值的含义

0x0D6数值	回读值的含义
0x00	数据和校验符号表示不存在错误。
0x01	解码器纠正1字节错误。
0x02	解码器纠正2字节错误。
0x03	解码器纠正3字节错误。
0x04	解码器纠正4字节错误。
0x05	解码器纠正5字节错误。
0xFF	存在6字节或更多错误，字节无法纠正。

发送和接收RS编码数据

ADF7023支持固定长度和可变长度数据包；然而，发送RS编码数据时建议使用固定长度数据包。必须设置PACKET_LENGTH_MAX(地址0x127)，以匹配编码数据长度(n)。设置TX_BASE_ADR(地址0x124)，使其指向编码数据的起始部分。

例如，若要发送存储在数据包RAM中以地址0x0A0开始的RS(38, 28)数据包：

```
1924A0 // Set TX_BASE_ADR to point at Packet
       // RAM Address 0xA0.
192726 // Set PACKET_LENGTH_MAX to 38 bytes.
```

使用RS编码数据包时，不要对数据包采用循环冗余校验(CRC)。由于CMD_RS_DECODE具有较强的错误检测能力，使用CRC将会是多余的。将CRC_EN位设为逻辑0(地址0x126[5])可禁用发送时附加CRC以及接收时检查CRC。

如需确定何时接收数据包，可置位INTERRUPT_MASK_0寄存器(地址0x100[2])中的INTERRUPT_ CRC_CORRECT。若CRC_EN设为逻辑0，则接收到完整的数据包时，置位ADF7023中断引脚。使用CMD_RS_ DECODE确定数据包是否有效。如需清零中断，可向INTERRUPT_SOURCE_0(地址0x336[2])中的位2写入逻辑1。

注意，若要发送具有5字节以上纠错能力的大数据包，可在一个数据包内发送多个RS码，如图2所示。然而，每个码字都必须单独编码与解码。

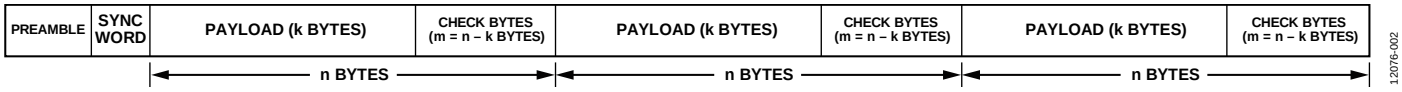

图2. 发送多个RS码字

RS编码和解码时间

典型RS编码时间列于表5。如图3所示，编码时间随编码长度(n)的增加而线性增加。

表5. 典型RS编码时间(m = 10)

编码长度 = n	时间(ms)
32	0.34
38	0.43
64	0.8
128	1.7
198	2.8

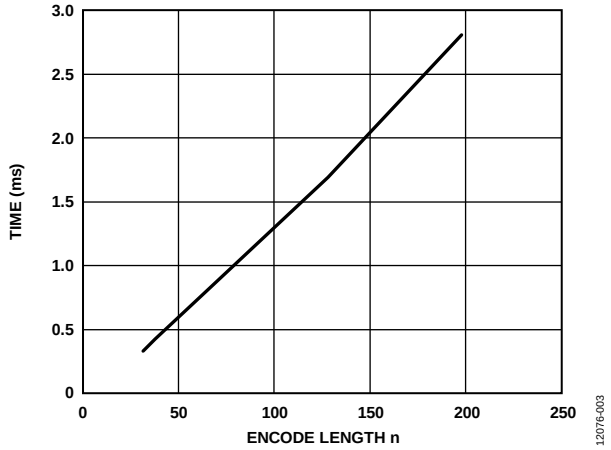

图3. 典型编码时间(m = 10)与编码长度n的关系

解码RS数据包的时间取决于编码长度和该编码长度中的错误字节数(见图4)。典型RS解码时间列于表6。

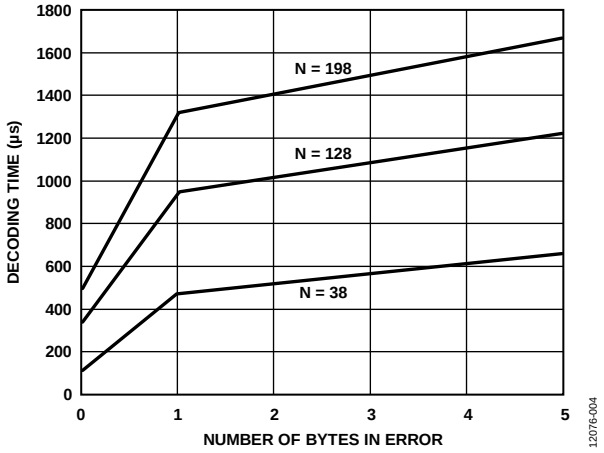

图4. 典型解码时间(m = 10)与纠错字节数的关系

表6. 典型RS解码时间(m = 10)

编码长度 = n	0个错误 时间(ms)	1个错误 时间(ms)	2个错误 时间(ms)	3个错误 时间(ms)	4个错误 时间(ms)	5个错误 时间(ms)
38	0.141	0.476	0.524	0.571	0.614	0.659
128	0.350	0.954	1.023	1.090	1.156	1.221
198	0.513	1.327	1.407	1.488	1.574	1.652

编码增益

对发送的数据包开销进行RS编码，并放宽接收器的PREAMBLE_MATCH(地址0x11B[3:0])和SYNC_ERROR_TOL(地址0x120[7:6])容差，便可提升数dB编码增益。不建议将错误字节的同步错误容差增加1位以上，也不建议将错误字节的前同步码容差增加2位对以上，因为这会让接收器检测到更多的错误数据包数量。各种数据速率下，相比非RS编码数据包，采用里德-所罗门(38, 28)编码数据包时可以达到的数据包错误率(PER)的改善情况参见图5、图6和图7。在图5、图6和图7中，PE是允许的前同步码错误数量，SE是允许的同步字错误数量，PL是数据包长度。

不同的同步字和前同步码容差下，采用纠错能力为5字节的里德-所罗门(38, 28)码时可获得的编码增益总结见表7。

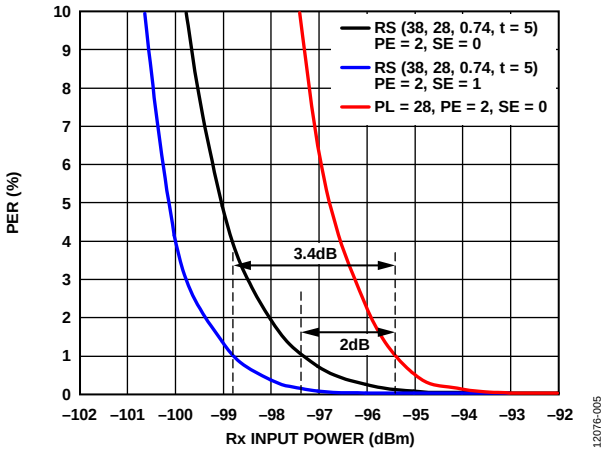

图5. 编码增益(300 kbps，24位同步字)

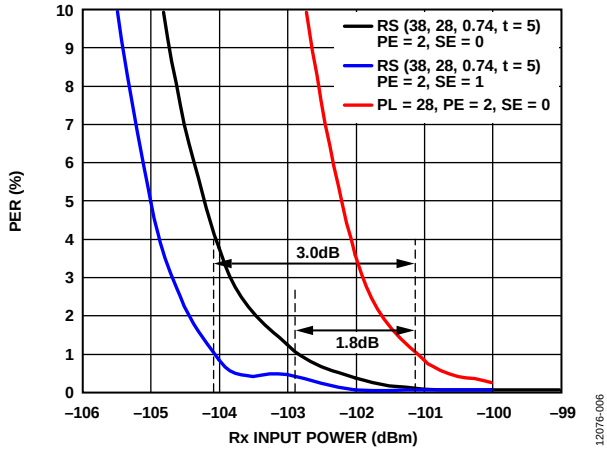

图6. 编码增益(100 kbps，24位同步字)

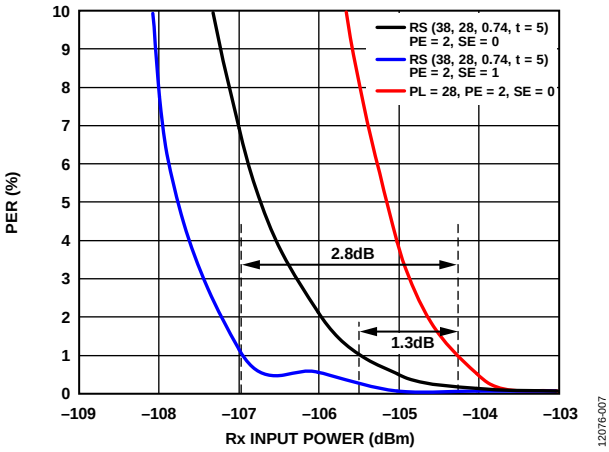

图7. 编码增益(38.4 kbps，24位同步字)

表7. RS编码增益总结

频移键控(FSK) 数据速率(kbps)	RS码	码率	前同步码错误(PE)容差	同步错误(SE)容差	编码增益(G)
38.4	(38, 28, t = 5)	0.74	2	0	1.3 dB
			2	1	2.8 dB
100	(38, 28, t = 5)	0.74	2	0	1.8 dB
			2	1	3 dB
300	(38, 28, t = 5)	0.74	2	0	2 dB
			2	1	3.4 dB

注释