ANALOG DEVICES

AN-1292
APPLICATION NOTE

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

Reed-Solomon Forward Error Correction and the ADF7023

by Stephen Hinchy and Kalim Khan

INTRODUCTION

This application note describes the Reed-Solomon (RS) firmware module available for the ADF7023 transceiver. The firmware module, which contains both RS forward error correction and Advanced Encryption Standard (AES) encryption, is named **rom_ram_7023_2_2_RS_AES.dat** and can be found at ftp://ftp.analog.com/pub/RFL/FirmwareModules/ADF7023.

In Reed-Solomon encoding, check symbols are appended to the transmitted data. When received, these symbols are used both to detect the presence of errors and to correct for errors in the received data. A Reed-Solomon code is specified as RS(n, k), where n and k are as shown in Figure 1. The number of bytes in the error that can be corrected by RS decode is t = (n − k)/2.

The ADF7023 firmware module implementation of RS encoding and decoding is flexible, and the user can select values of n and k that enable up to five bytes of error within a packet to be corrected. RS encoded packets are resilient to both burst and random errors, and exhibit coding gain to achieve improved link margin. The RS code rate for various payload lengths with t = 5 is shown in Table 1.

Table 1. RS Code Rate

No. of Encoded Bytes (n)	No. of Data Bytes (k)	Overhead (m = n − k)	No. of Correctable Bytes (t = m/2)	Code Rate (R = k/n)
38	28	10	5	0.74
128	118	10	5	0.92
198	188	10	5	0.95

COMMANDS AND PACKET RANDOM ACCESS MEMORY REGISTER LOCATIONS

Packet random access memory (RAM) registers from Address 0x010 to Address 0x0D5 are available to store Reed-Solomon encoded data. Thus, the maximum encoded length (n) possible is 198 bytes. All other packet RAM registers are reserved for use either by the ADF7023 or by the Reed-Solomon firmware.

Table 2 shows the registers that the user must initialize prior to RS encoding or decoding. These register definitions are specific to the firmware module and are not applicable to normal operation of the ADF7023.

Table 2. Register Locations to Initialize Prior to RS Encoding or Decoding

Register Address[1]	Register Name	Description
0x0D7	VAR_RSPCKTADR	Pointer to the packet RAM address of the start of a sequence of k data bytes for encoding, or n data bytes and check symbols for decoding
0x0D8	VAR_RS_K	The value of k, that is, the number of data bytes
0x0D9	VAR_RS_N	The value of n, that is, the number of data bytes plus check symbols

[1] These register definitions are specific to the firmware module and are not applicable to normal operation of the ADF7023.

The commands shown in Table 3 are necessary to initialize an RS encode, perform an RS encode, or perform an RS decode. See the RS Procedure section and RS Decode Procedure section of this application note for additional information regarding RS encode and RS decode procedures.

Table 3. Reed-Solomon Commands

Command	Code	Description
CMD_RS_ENCODE_INIT	0xD1	Initializes the internal variables required for the Reed-Solomon encoding
CMD_RS_ENCODE	0xD0	Calculates and appends the Reed-Solomon check bytes to the transmit payload data stored in packet RAM
CMD_RS_DECODE	0xD2	Performs a Reed-Solomon error correction on the received payload data stored in packet RAM

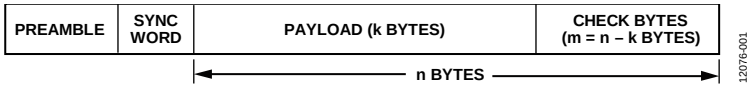

Figure 1. Reed-Solomon Encoded Packet

TABLE OF CONTENTS

REVISION HISTORY

2/14—Revision 0: Initial Version

RS PROCEDURES

WRITING THE RS FIRMWARE MODULE TO THE ADF7023

Prior to using the RS firmware module, the user must write it to the program RAM of the ADF7023. The following steps detail how to write the RS firmware module to program RAM:

1. Ensure that the ADF7023 is in the PHY_OFF state.
2. Issue the CMD_RAM_LOAD_INIT command (Address 0xBF).
3. Write the module to program RAM using an SPI memory block write (0x1E00[firmware module]; see the Block Write section of the ADF7023 data sheet).
4. Issue the CMD_RAM_LOAD_DONE command (Address 0xC7).

The firmware module is now stored in program RAM.

DETERMINING WHEN REED-SOLOMON COMMANDS ARE COMPLETE

The CMD_FINISHED interrupt can be used to determine when the CMD_RS_ENCODE_INIT, CMD_RS_ENCODE, and CMD_RS_DECODE commands are complete. To enable this interrupt, set Bit 0 (CMD_FINISHED) of the INTERRUPT_MASK_1 register (Address 0x101[0]). When this mask bit is set, the interrupt pin of the ADF7023 is asserted upon completion of any command. The interrupt is cleared by writing Logic 1 to Bit 0 of INTERRUPT_SOURCE_1 (Address 0x337[0]). Further information on the use of interrupts is available in the Interrupt Generation section of the ADF7023 data sheet.

RS ENCODE PROCEDURE

In the following example, 28 data bytes are encoded so that up to five bytes can be corrected (RS(38, 28)). These 28 bytes of data are in the packet RAM starting at Address 0x0A0.

```
// 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 DECODE PROCEDURE

In the following example, an RS(38, 28) encoded packet is decoded and the number of bytes in error is read back.

```
//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.
```

Table 4. Meaning of RS Decode Readback Value

Value of 0x0D6	Meaning of Readback Value
0x00	The data and check symbols indicate no errors present.
0x01	The decoder corrected 1 byte.
0x02	The decoder corrected 2 bytes.
0x03	The decoder corrected 3 bytes.
0x04	The decoder corrected 4 bytes.
0x05	The decoder corrected 5 bytes.
0xFF	There are 6 or more bytes with errors present, and the bytes cannot be corrected.

TRANSMITTING AND RECEIVING RS ENCODED DATA

The ADF7023 supports both fixed length and variable length packets; however, fixed length packets are recommended when transmitting RS encoded data. PACKET_LENGTH_MAX (Address 0x127) must be set to match the length of the encoded data (n). Set TX_BASE_ADR (Address 0x124) to point at the start of the encoded data.

For example, to transmit an RS(38, 28) packet stored in packet RAM starting at Address 0x0A0:

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

When using RS encoded packets, do not append a cyclic redundancy check (CRC) to the packet. A CRC is superfluous due to the error detection capability CMD_RS_DECODE. Appending the CRC in transmit and checking the CRC in receive can be disabled by setting the CRC_EN bit to Logic 0 (Address 0x126[5]).

To determine when a packet is received, set INTERRUPT_CRC_CORRECT in the INTERRUPT_MASK_0 register (Address 0x100[2]). With CRC_EN set to Logic 0, when a complete packet is received, the interrupt pin of the ADF7023 is asserted. Use CMD_RS_ DECODE to determine if the packet is valid. To clear the interrupt, write Logic 1 to Bit 2 of INTERRUPT_SOURCE_0 (Address 0x336[2]).

Note that to transmit a large packet with more than five bytes of error correction capability, it is possible to transmit multiple RS code words within one packet, as shown in Figure 2. However, each code word must be encoded and decoded separately.

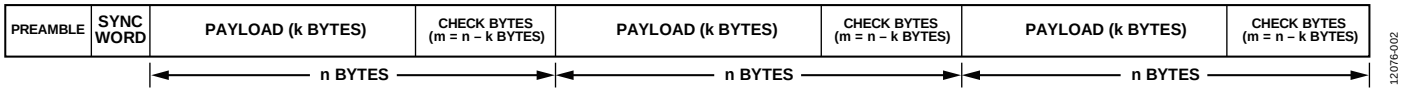

Figure 2. Transmitting Multiple RS Code Words

RS ENCODE AND DECODE TIMES

Typical RS encode times are listed in Table 5. As shown in Figure 3, the encode time increases linearly with the encoded length (n).

Table 5. Typical RS Encode Times for m = 10

Encoded Length = n	Time (ms)
32	0.34
38	0.43
64	0.8
128	1.7
198	2.8

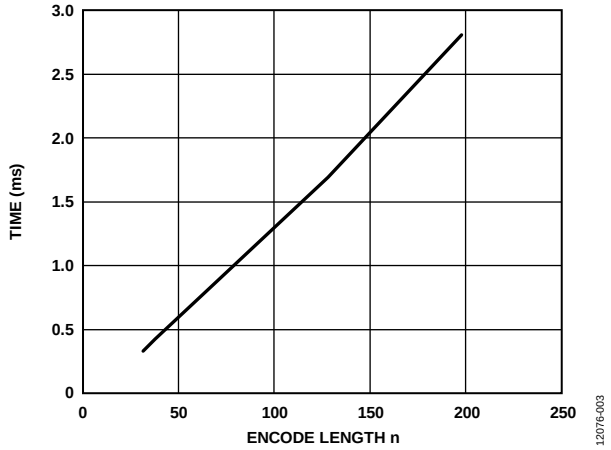

Figure 3. Typical Encoding Time for m = 10 vs. Encoded Length n

The time to decode an RS packet depends upon both the encoded length and the number of bytes in error within that encoded length (see Figure 4). Typical RS decode times are listed in Table 6.

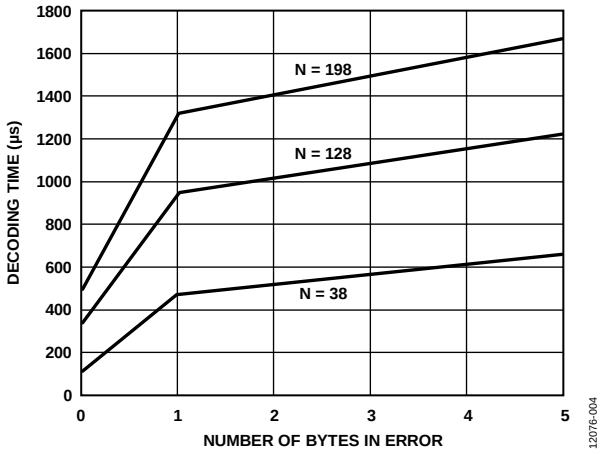

Figure 4. Typical Decode Time for m = 10 vs. Number of Bytes Corrected

Table 6. Typical RS Decode Times for m = 10

Encoded Length = n	Time for 0 Errors (ms)	Time for 1 Error (ms)	Time for 2 Errors (ms)	Time for 3 Errors (ms)	Time for 4 Errors (ms)	Time for 5 Errors (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

CODING GAIN

Several decibels of coding gain improvements can be achieved by RS encoding the payload of a transmitted packet and relaxing the tolerance of the PREAMBLE_MATCH (Address 0x11B[3:0]) and SYNC_ERROR_TOL (Address 0x120[7:6]) of the receiver. Increasing the sync error tolerance more than one bit in error and the preamble tolerance more than two bit pairs in error is not recommended, because it results in an increasing number of false packet detects at the receiver. The improvements in packet error rate (PER) achievable at various data rates with Reed-Solomon (38, 28) encoded packets over nonRS encoded packets are shown in Figure 5, Figure 6, and Figure 7. (In in Figure 5, Figure 6, and Figure 7, PE is the number of preamble errors allowed, SE is the number of sync word errors allowed, and PL is the packet length.)

A summary of the coding gains achieved with a Reed-Solomon (38, 28) code with an error correcting capability of 5 bytes, at various sync word and preamble tolerances, is shown in Table 7.

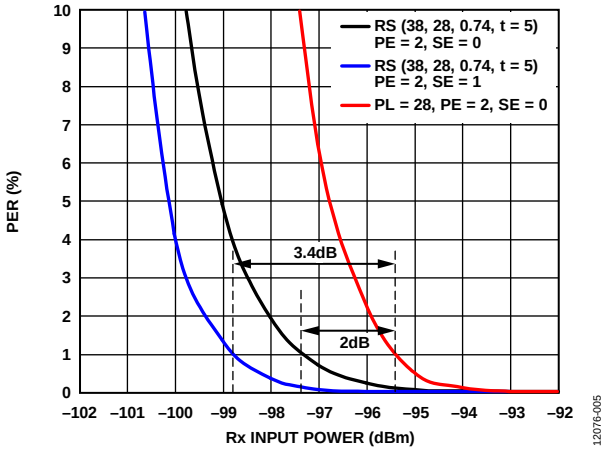

Figure 5. Coding Gain at 300 kbps with a 24-Bit Sync Word

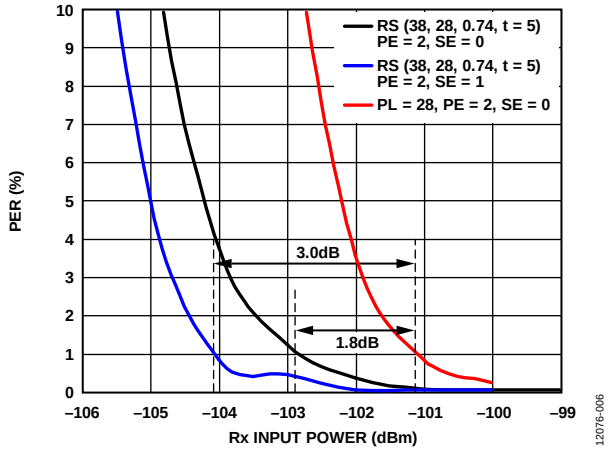

Figure 6. Coding Gain at 100 kbps with a 24-Bit Sync Word

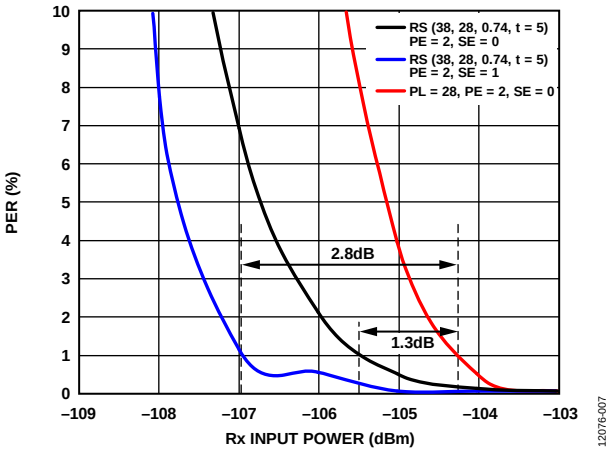

Figure 7. Coding Gain at 38.4 kbps with a 24-Bit Sync Word

Table 7. Summary of RS Coding Gain

Frequency Shift Keying (FSK) Data Rate (kbps)	RS Code	Code Rate	Preamble Error (PE) Tolerance	Sync Error (SE) Tolerance	Coding Gain (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

NOTES

AN12076-0-2/14(0)