

AHA Application Note

Primer: Reed-Solomon Error Correction Codes (ECC)

ANRS01_0404



Comtech EF Data Corporation

■ 1126 Alturas Drive ■ Moscow ID 83843 ■ tel: 208.892.5600 ■ fax: 208.892.5601 ■ www.aha.com ■

Table of Contents

1.0 Introduction	1
2.0 Basics	1
2.1 How Error Correcting Codes Work	2
2.2 The Advantages of Error Detection and Correction	3
2.3 Channel Capacity and Coding Limits	4
2.4 Channel Noise and Error Types	4
3.0 Reed-Solomon Block Codes	5
3.1 Characteristics	5
3.2 Parameters	6
3.3 Code Rate (R)	6
3.4 Interleaving	6
3.5 Correction Power of RS Codes	7
3.6 Summary	7
4.0 Code Performance	7
4.1 Probability of Error	7
4.2 Probability of a Mis-Correct	9
4.3 Code Performance Curves	11
5.0 Choosing a Code	13
5.1 Issues	13
5.2 Matching the Code to the Channel Noise	13
5.3 Cost vs. Performance	14
6.0 Conclusions	14
6.1 Reed-Solomon Coprocessors by AHA	14
7.0 About AHA	15
8.0 Additional Reading	16

List of Figures

Figure 1: Random Symbol Block Error Performance for the RS(255,k) Code for k=235, t=10,
Through k=253, t=1. 11

Figure 2: Performance Curves for RS Codes of Rate .92 11

Figure 3: Performance Curves in Terms of CBER for RS Codes of Rate .92 12

Figure 4: Performance Curves for RS Codes of Different Block Lengths, for t=10 and t=5 13

1.0 INTRODUCTION

This document presents the basics of error detection and correction (EDAC) using Reed-Solomon codes. It addresses the following questions:

- What is forward error correction?
- What are block codes and what specifically are Reed-Solomon (RS) codes?
- What parameters specify an RS code?
- What kinds of channel noise can an RS code handle?
- What system performance improvements are possible by using EDAC, and specifically RS codes?

The integrity of received data is a critical consideration in the design of digital communications and storage systems. Many applications require the absolute validity of the received message, allowing no room for errors encountered during transmission and/or storage.

Reliability considerations frequently require that Forward Error Correction (FEC) techniques be used when Error Detection And Correction (EDAC) strategies are required. The power of FEC is that the system can, without retransmissions, find and correct limited errors caused by a transport or storage system. While there are several approaches to FEC, this note will concentrate on the Reed-Solomon codes. These codes provide powerful correction, have high channel efficiency, and are very versatile. With the advent of VLSI implementations, such as the AHA PerFEC 4000 series, RS codes can be easily and economically applied to both high and low data rate systems. In some new circuits, the FEC function is integrated with data formatters and buffer managers.

2.0 BASICS

Error detection and correction codes use redundancy to perform their function. This means that extra code symbols are added to the transmitted message to provide the necessary detection and correction information.

The simplest form of redundancy for detecting errors in digital binary messages is the parity-check scheme. In the even parity scheme, an extra bit is attached to each message block so the total number of 1's in the data block is even. A transmission error is detected when the number of 1's in the received message is odd. This simple scheme will detect only an odd number of errors in the transmitted message and cannot correct erroneous messages.

Redundancy is used by all Forward Error Correction (FEC) codes to perform EDAC. FEC codes allow a receiver in the system to perform EDAC without requesting a retransmission.

In 1948, C.E. Shannon published a classic technical paper on using redundancy to perform EDAC. In it, he proved that impressive performance gains can be realized with the proper use of redundancy, but the paper gave no indication as to which codes might be used to achieve these gains.

In the following years rapid advancements were made in both EDAC theory and EDAC practice. In 1960, I. Reed and G. Solomon developed the "block code" coding technique called Reed-Solomon (RS) coding. Today, RS codes remain popular due to standards compliance and economic implementations.

2.1 HOW ERROR CORRECTING CODES WORK

Correcting errors, once they have been detected, requires the addition of several redundancy symbols. The number of redundant symbols is determined by the amount of error correction required. These additional check symbols must contain enough information to locate the position and determine the value of the erroneous information symbols.

A simple example of error correction using Hamming codes, will help to explain how Reed-Solomon codes work.

The table below shows all possible code words for a (7,4) Hamming code. This means there are a total of 7 bits where 4 are information bits (and therefore 3 are check bits). Bits A, B, C, and D represent the data (information bits). Bits E, F, and G represent the check bits which are calculated by performing an exclusive-or operation (XOR, denoted by $\oplus$) on the specified data bits as follows:

$$\text{Equation 1 - Bit E} = A \oplus B \oplus C$$

$$\text{Equation 2 - Bit F} = A \oplus B \oplus D$$

$$\text{Equation 3 - Bit G} = A \oplus C \oplus D$$

	DATA BITS				CHECK BITS		
	A	B	C	D	E	F	G
1	0	0	0	0	0	0	0
2	0	0	0	1	0	1	1
3	0	0	1	0	1	0	1
4	0	0	1	1	1	1	0
5	0	1	0	0	1	1	0
6	0	1	0	1	1	0	1
7	0	1	1	0	0	1	1
8	0	1	1	1	0	0	0
9	1	0	0	0	1	1	1
10	1	0	0	1	1	0	0
11	1	0	1	0	0	1	0
12	1	0	1	1	0	0	1
13	1	1	0	0	0	0	1
14	1	1	0	1	0	1	0
15	1	1	1	0	1	0	0
16	1	1	1	1	1	1	1

The entire range of possible code-words (or “dictionary”), therefore, consists of only 16, 7-bit words (out of a possible 128 combinations). If a code word is received that does not match one of these 16, then it is, by definition, in error.

To calculate which word is in error, the XOR equations are performed on the incoming data. If no error has occurred, then the three equations will all return correct (Logical True). If any one of the 7 bits is in error, then a certain subset of the three XOR equations will fail, because each of the 7 bits occurs in a different subset of the three equations. The table below shows which equations will be false for each bit in error. Once the incorrect bit is located, it is corrected by inverting it.

<i>BIT IN ERROR</i>	<i>EQUATION 1</i>	<i>EQUATION 2</i>	<i>EQUATION 3</i>
NONE	TRUE	TRUE	TRUE
A	FALSE	FALSE	FALSE
B	FALSE	FALSE	TRUE
C	FALSE	TRUE	FALSE
D	TRUE	FALSE	FALSE
E	FALSE	TRUE	TRUE
F	TRUE	FALSE	TRUE
G	TRUE	TRUE	FALSE

For example, assume that we received the following code word:

1 0 0 0 0 0 1

We perform the XOR equations as follows:

$$\begin{aligned}\text{Equation 1} = E &= A \oplus B \oplus C \\ 0 &\neq 1 \oplus 0 \oplus 0 \quad (\text{False})\end{aligned}$$

$$\begin{aligned}\text{Equation 2} = F &= A \oplus B \oplus D \\ 0 &\neq 1 \oplus 0 \oplus 0 \quad (\text{False})\end{aligned}$$

$$\begin{aligned}\text{Equation 3} = G &= A \oplus C \oplus D \\ 1 &= 1 \oplus 0 \oplus 0 \quad (\text{True})\end{aligned}$$

Thus, by definition, if Equations 1 and 2 are false, and Equation 3 is true, then bit B is in error. When we invert it (correct it) we get:

1 1 0 0 0 0 1

which is a correct code word.

Thus, we see that if any one of the 7 bits is in error, we can find and correct the error by interpreting the pattern of failed XOR equations.

Reed-Solomon codes work essentially the same as Hamming codes, except RS codes deal with multi-bit symbols rather than individual bits. For example a (255,235) Reed-Solomon code specifies a total block length of 255 bytes (or symbols); 235 bytes used for information and 20 check bytes. The check bytes are calculated in a similar manner to the 3 check bits in the Hamming code example above and are appended to the end of the data block. Reed-Solomon codes are much more complex however, and require a significant amount of arithmetic and logical processing.

2.2 THE ADVANTAGES OF ERROR DETECTION AND CORRECTION

EDAC has a number of advantages for the design of high reliability digital systems:

- 1) Forward Error Correction (FEC) enables a system to achieve a high degree of data reliability, even with the presence of noise in the communications channel. Data integrity is an important issue in most digital communications systems and in all mass storage systems.

- 2) In systems where improvement using any other means (such as increased transmit power or components that generate less noise) is very costly or impractical, FEC can offer significant error control and performance gains.
- 3) In systems with satisfactory data integrity, designers may be able to implement FEC to reduce the costs of the system without affecting the existing performance. This is accomplished by degrading the performance of the most costly or sensitive element in the system, and then regaining the lost performance with the application of FEC.

In general, for digital communication and storage systems where data integrity is a design criteria, FEC needs to be an important element in the trade-off study for the system design. The introduction of the PerFEC line of FEC encoders and decoders makes powerful FEC implementation a realistic goal for most digital communication and storage systems. More than ever before, FEC is available for a wide range of applications.

2.3 CHANNEL CAPACITY AND CODING LIMITS

System capacity, C , in bits per second gives an upper limit to the number of bits per second that can be reliably transmitted across a given communications channel. In a paper published in 1948, Shannon showed that the system capacity for channels perturbed by additive white Gaussian noise is a function of three system parameters:

- W - channel bandwidth in Hz
- S - received signal power
- N - additive noise power

The capacity relationship among these parameters, known as the Shannon-Hartley Theorem, can be stated as:

$$C = W \log_2[1 + S/N]$$

or

$$C = W \log_2[1 + (E_b/N_o)(C/W)]$$

where: E_b is the signal energy per bit and N_o is the noise power level in Watts/Hz.

Shannon proved that on an infinite bandwidth channel, with a sufficiently complicated coding scheme, it is possible to transmit information with an arbitrarily small error rate. This can be accomplished at a transmission rate of (R) bits/sec, where $R < C$. For a rate $R > C$, it is not possible to achieve an arbitrarily small error rate no matter what code is used and no matter how much redundancy is added.

It may be shown from the Shannon-Hartley Theorem that the required limit for E_b/N_o approaches the Shannon limit of -1.6 dB as W increases without bound. An excellent measure of a code's performance is how well it performs in relation to the Shannon bound. Shannon's initial work proved that good codes do exist, but he never showed how to generate the codes. Today using modern codes, including the Reed-Solomon codes, coding systems have been designed to operate within a few dB of the Shannon bound.

2.4 CHANNEL NOISE AND ERROR TYPES

A system's noise environment can cause errors in the received message. Properties of these errors depend upon the noise characteristics of the channel. Errors which are usually encountered fall into three broad categories:

- 1) Random errors - the bit error probabilities are independent or nearly independent of each other. Additive noise typically causes random errors.

- 2) Burst errors - the bit errors occur sequentially in time and as groups. Media defects in digital storage systems typically cause burst errors.
- 3) Impulse errors - large blocks of the data are full of errors. Lightning strikes and major system failures typically cause impulse errors.

Random errors occur in the channel when individual bits in the transmitted message are corrupted by noise. Random errors are generally caused by thermal noise in communications channels. We will show in Section 5.2, that block codes and specifically the Reed-Solomon codes can be a good code choice to correct random channel errors.

Burst errors happen in the channel when errors occur continuously in time. Burst errors can be caused by fading in a communications channel or by large media and mechanical defects in a storage system. For some codes, burst errors are difficult to correct, however, block codes (including Reed-Solomon codes) handle burst errors very efficiently.

Impulse errors can cause catastrophic failures in the communications system that are so severe they may be unrecognizable by FEC using present-day coding schemes. In general all coding systems fail to reconstruct the message in the presence of catastrophic errors. However, certain codes like the Reed-Solomon codes can detect the presence of a catastrophic error by examining the received message. This is very useful in system design because the unrecoverable message can at least be flagged at the decoder.

The following sections describe RS codes and focus on their performance in each of these noise environments.

3.0 REED-SOLOMON BLOCK CODES

3.1 CHARACTERISTICS

Block codes differ from other EDAC codes because they process data in batches or blocks rather than continuously. The data is partitioned into blocks, and each block is processed as a single unit by both the encoder and the decoder.

There are two classifications of block codes: systematic and non-systematic. Non-systematic codes add redundancy and transform the coded message such that no part of the original message is recognizable from the un-decoded message. Non-systematic codes must be decoded properly before any message information is available at the receiver.

With systematic codes the message data is not disturbed in any way in the encoder and the redundant symbols are added separately to each block. The AHA RS codecs implement a systematic block code. All of these actions appear to be taking place continuously in real time, regardless of the error patterns encountered because of the internal architecture of the PerFEC codecs.

For an RS code, each symbol may be represented as a binary m -tuple. RS codes may be considered to be a special case of Bose-Chaudhuri-Hocquenghem (BCH) codes.

3.2 PARAMETERS

The parameters of a Reed-Solomon code are:

m	=	the number of bits per symbol
n	=	the block length in symbols
k	=	the uncoded message length in symbols
$(n-k)$	=	the parity check symbols (check bytes)
t	=	the number of correctable symbol errors
$(n-k)$	=	$2t$ (for $n-k$ even)
$(n-k)-1$	=	$2t$ (for $n-k$ odd)

Therefore, an RS code may be described as an (n,k) code for any RS code where, $n \leq 2^m - 1$, and $n - k \geq 2t$.

RS codes operate on multi-bit symbols rather than on individual bits like binary codes. The AHA PerFEC codecs are typical of RS codes and use 8-bit symbols. This allows symbols to correspond to digital bytes.

Consider the RS(255,235) code. The encoder groups the message into 235 8-bit symbols and adds 20 8-bit symbols of redundancy to give a total block length of 255 8-bit symbols. In this case, 8% of the transmitted message is redundant data. In general, due to decoder constraints, the block length cannot be arbitrarily large.

The block length for the PerFEC codecs is bounded by the following equation:

$$1 + 2t \leq n \leq 255$$

The number of correctable symbol errors (t), and block length (n) is set by the user.

3.3 CODE RATE (R)

The code rate (efficiency) of a code is given by:

$$\text{code rate} = R = k/n$$

where k is the number of information (message) symbols per block, and n is total number of code symbols per block. This definition holds for all codes whether block codes or not.

Codes with high code rates are generally desirable because they efficiently use the available channel for information transmission. RS codes typically have rates greater than 80% and can be configured with rates greater than 99% depending on the error correction capability needed. The RS codes used in the AHA PerFEC codecs have rates which can be as high as 99.2%.

3.4 INTERLEAVING

Interleaving is another tool used by the code designer to match the error correcting capabilities of the code to the error characteristics of the channel. Interleaving in a digital communications systems enhances the random-error correcting capabilities of a code to the point that it can also become useful in a burst-noise environment.

The interleaver subsystem rearranges the encoded bits over a span of several block lengths. The amount of error protection, based on the length of bursts encountered on the channel, determines the span length of the interleaver. The receiver must be given the details of the bit arrangement so the bit stream can be de-interleaved before it is decoded. The overall effect of interleaving is to spread out the effects of long bursts so they appear to the decoder as independent random bit errors or shorter more manageable burst errors.

The AHA RS codecs require external circuitry to accomplish interleaving.

3.5 CORRECTION POWER OF RS CODES

In general, an RS decoder can detect and correct up to $(t = r/2)$ incorrect symbols if there are $(r = n - k)$ redundant symbols in the encoded message. If the code is being used only to detect errors and not to correct them, (r) errors can be detected. One redundant symbol is used in detecting and locating each error, and one more redundant symbol is used in identifying the precise value of that error.

This concept of using redundant symbols to either locate or correct errors is useful in the understanding of erasures. The term “erasures” is used for errors whose position is identified at the decoder by external circuitry. If an RS decoder has been instructed that a specific message symbol is in error, it only has to use one redundant symbol to correct that error and does not have to use an additional redundant symbol to determine the location of the error.

If the locations of all the errors are given to the RS codec by the control logic of the system, $2t$ erasures can be corrected. In general, if (E) symbols are known to be in error (eg. erasures) and if there are (e) errors with unknown locations, the block can be correctly decoded provided that $(2e + E) < r$.

3.6 SUMMARY

In summary, RS block codes have four basic properties which make them powerful codes for digital communications:

- 1) An RS decoder acts on multi-bit symbols rather than on single bits. Thus, up to eight bit-errors in a symbol can be treated as a single symbol error. Strings of errors, or bursts, are therefore handled efficiently.
- 2) The RS codes with very long block lengths tend to average out the random errors and make block codes suitable for use in random error correction.
- 3) RS codes are well-matched for the messages in a block format, especially when the message block length and the code block length are matched. Block length is variable on the fly with the PerFEC codecs and therefore the message block length and the code block length can always be matched.
- 4) The complexity of the decoder can be decreased as the code block length increases and the redundancy overhead decreases. Hence, RS codes are typically large block length, high code rate, codes.

4.0 CODE PERFORMANCE

4.1 PROBABILITY OF ERROR

The most common measure of performance for any error correction code is the estimated probability of decoder or transmission error. Since block codes act on symbols, we will first deal with symbol errors rather than bit errors.

The probability of an uncorrectable error is given by P_{UE} . An uncorrectable error occurs when more than (t) received symbols are in error in a given block. When this happens one of two actions result at the decoder. Either the message block is recognized as being uncorrectable and is flagged as such (this is called a recognized error) or the error pattern

is assumed by the decoder as correctable, and the decoder mistakenly corrects the entire message block to the wrong message (this is called a decoding error). When a decoding error occurs, the entire code block ($n-k$ symbols) is decoded incorrectly. The probability of a decoding error, P_{DE} , is dealt with separately in Section 4.2.

P_{UE} , the probability of an uncorrectable error, is the ratio of the number of uncorrectable code blocks to the total number of received code blocks, in the limiting case where the number of code blocks received becomes large.

An important parameter in determining P_{UE} is the channel symbol error rate P_{SE} . This is the ratio of the number of received symbol errors to the total number of received symbols, in the limit as the number of symbols becomes large.

P_{SE} is the probability that the channel will change a symbol during the transmission of the message. Without FEC the channel error probability, P_{SE} , would also be the received symbol error probability. However, with FEC, the decoded symbol error probability can be reduced by many orders of magnitude.

The following equation will show how to calculate error rate improvement for a channel that produces symbol errors. In the example, we will assume that the symbol errors are independent and that no erasure information is available.

An expression for the probability of an uncorrectable error is given by:

$$P_{UE} = 1 - \sum_{i=0}^t \binom{n}{i} (P_{SE})^i (1 - P_{SE})^{n-i}$$

where n is the number of symbols per code block. The symbol

$$\binom{n}{i}$$

is the binomial coefficient and is evaluated as:

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

where:

$$n! = \prod_{i=1}^n i = (1)(2)(3) \dots (n-1)(n)$$

An example, using typical parameters from the PerFEC codecs, illustrates the power of using FEC to improve system error performance. Substituting $n = 255$, $t = 5$ and $P_{SE} = 10^{-3}$ into the previous equation, P_{UE} is 3×10^{-7} . This shows over three orders of magnitude improvement in error performance using RS codes for FEC.

A P_{UE} of 3×10^{-7} is interpreted as: for every $1/(3 \times 10^{-7})$ or (3.3×10^6) message blocks, on the average, one of them will be uncorrectable.

The bit error rate (P_B) rather than the symbol error rate (P_{SE}), may be known for a given channel. Under the assumption of purely random bit errors, we can write:

$$P_{SE} = 1 - (1 - P_B)^m$$

where m is the number of bits per symbol.

For more complicated, less random error characteristics, the P_{SE} needs to be determined on a case-by-case basis. In general the RS codes perform better as the bit error pattern becomes less random. The formulas presented in this section generally predict larger error probabilities than will be encountered with correlated or burst-type error patterns.

The error probability calculation for general burst errors is complicated when interleaving is used. However, under some simplifying assumptions, the calculation is straightforward and will be described next.

P_{UEB} is the probability of an uncorrectable error when there are burst errors on the channel and interleaving is used. To calculate P_{UEB} , several limiting assumptions are made:

- 1) (I) symbols are changed by each burst error.
- 2) The interleaving depth is (I) symbols.
- 3) There is no erasure decoding.

In general, the interleaving depth is set such that expected bursts will impact only one interleave. Assumption 1 is then seen to be a worst case condition.

Let P_{BURST} be the probability of a burst error, given by the ratio of the number of received burst symbol errors to the total number of symbols sent. Then:

$$P_{UEB} \leq 1 - \sum_{i=0}^t \binom{n}{i} [(P_{BURST})(I)]^i [1 - (P_{BURST})(I)]^{(n-i)}$$

where: I is the interleaving depth.

Note that this is the same as the calculation for P_{UE} from the previous page and that $(P_{BURST})(I)$ has simply been substituted in that equation for P_{SE} . These equations are equivalent because the assumptions allow each burst to be treated as (I) independent symbol errors.

This formula shows that large performance improvements are available using FEC, even in the presence of channel burst errors. For example, if the $P_{BURST} = 10^{-4}$ and the burst length is 5 symbols, then the interleaving depth is also 5. With these values and using a code where $n = 255$ and $t = 5$, the probability of an uncorrectable error, P_{UEB} , calculates to be 5×10^{-9} . This shows a very significant performance improvement for this type of difficult burst error channel.

Note that P_{BURST} is defined in terms of symbol errors. A burst of length 4 symbols is a burst of length (4m) bits. Very long bursts in terms of bit errors, are efficiently handled using block codes with even modest interleaving.

When a block code fails to correct a block, or mis-corrects a block, the data reliability within the entire block is lost. Thus, the errors, when they occur, invalidate full blocks of data. The standard bit error rate performance figure (BER), which is used to analyze independent bit errors, can be misleading because of this. A more appropriate figure of merit for block codes, and one frequently used in the magnetic recording industry, is the Corrected Bit Error Rate ($CBER$). The $CBER$ is the reciprocal of the expected number of correct bits between errors.

The $CBER$ is given by:

$$CBER = \frac{P_{UE}}{[(n)(m)]}$$

where m is the number of information bits per symbol and n is the number of bits per block.

With the proper use of block code based FEC, storage systems can obtain a $CBER$ less than 10^{-10} . A $CBER$ of 10^{-10} means that on the average there will be 10^{10} consecutive correct bits between errors.

4.2 PROBABILITY OF A MIS-CORRECT

“Mis-correction” or “decoding error” is the name given to an erroneous EDAC operation. In mis-correction the received block contains a combination of errors such that the block is corrected to the wrong message at the decoder. This kind of mis-correction happens when the error pattern mimics another received message block within the code’s span of correctability. Thus, the codec misinterprets the situation, and performs a mistaken correction (a “mis-correct”) which yields a totally wrong message block at the receiver.

In binary coding theory, the minimum Hamming distance (d^*) of a code is the smallest number of bit positions that when changed (toggled), will turn one valid binary code word into another valid code word. This parameter was developed by Richard Hamming who also was instrumental in the development of coding theory.

The minimum Hamming distance for an RS block code is given by:

$$d^* = n - k + 1$$

Letting (e) be the actual number of errors (of unknown location) in the received message, and (E) be the number of erasures (of known location), if:

$$2e + E \leq d^* = (n - k + 1)$$

then any linear block code such as an RS code can flawlessly decode and reconstruct the received message. If there are so many errors that this condition is not met, one of two situations occurs:

- 1) the error is properly detected by the decoder and becomes a detected error, or
- 2) the erroneous message appears to the decoder as a correctable error and the error is corrected to the wrong code word and becomes a decoding error.

The conditional probability of a decoding error, conditioned on the occurrence of an uncorrectable error, $P_{DE|UE}$, is given for ($d^* - E - 1$) even by:

$$P_{DE|UE} \leq \frac{1}{\left(\frac{d^* - E - 1}{2}\right)!}$$

and for ($d^* - E - 1$) odd by:

$$P_{DE|UE} \leq \frac{1}{\left(\frac{d^* - E - 1}{2}\right)!(2^m - 1)}$$

If ($d^* - E - 1$)/2 is not an integer, then the largest integer that is not greater than ($d^* - E - 1$)/2 is used.

Assuming no erasures, for the RS(255,235) code $P_{DE|UE}$ is given by:

$$P_{DE|UE} \leq \frac{1}{10!} = 2.8 \times 10^{-7}$$

and with one erasure $P_{DE|UE}$ becomes:

$$P_{DE|UE} \leq \frac{1}{9!(255)} = 1.1 \times 10^{-8}$$

However, with one erasure only nine errors may be corrected rather than ten. Finally, the probability of interest to us is P_{DE} , the probability of a decoding error, and is written:

$$P_{DE} = (P_{DE|UE})(P_{UE})$$

This can also be expressed as:

$$P_{DE} \leq \frac{1}{l!} \times P_{UE}$$

For every extra check byte used, P_{DE} is increased by at least 1/256.

These results indicate that when this particular code is used, less than one out of a million uncorrectable errors will not be recognized as such. Thus, even if the decoder cannot correctly reconstruct the message, it will almost always determine there was in fact a problem, and will set the “unrecoverable” flag. This useful capability is a characteristic of linear block codes in general, and of the RS codes used by AHA’s PerFEC codecs in particular. An “uncorrectable” flag is a standard feature of the PerFEC codecs.

4.3 CODE PERFORMANCE CURVES

Reed-Solomon code performance is most easily illustrated with a set of curves. Figure 1 shows a typical performance curve with the probability of symbol error on the horizontal axis and the probability of an uncorrectable error on the vertical axis for a class of RS $(255, k)$ codes for $k=235, t=10$, through $k=253, t=1$. This curve is for decoding of random symbol errors and includes no erasures.

The curves of Figure 1 illustrate the performance of codes of different rate. For the RS(255,235) code, the rate is $(235/255) = .92$, while for the RS(255,253) code, the rate is $(253/255) = .99$.

Figure 1: Random Symbol Block Error Performance for the RS(255,k) Code for $k=235, t=10$, Through $k=253, t=1$.

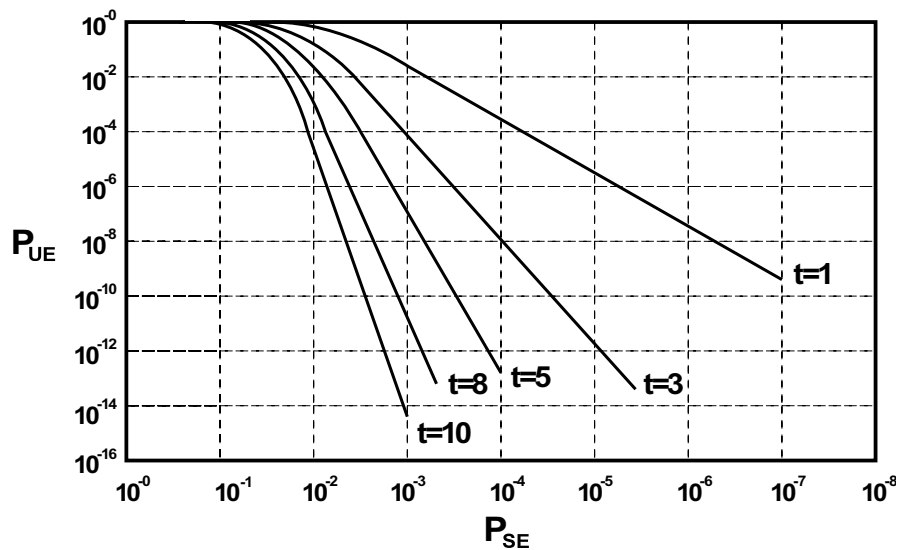


Figure 2: Performance Curves for RS Codes of Rate .92

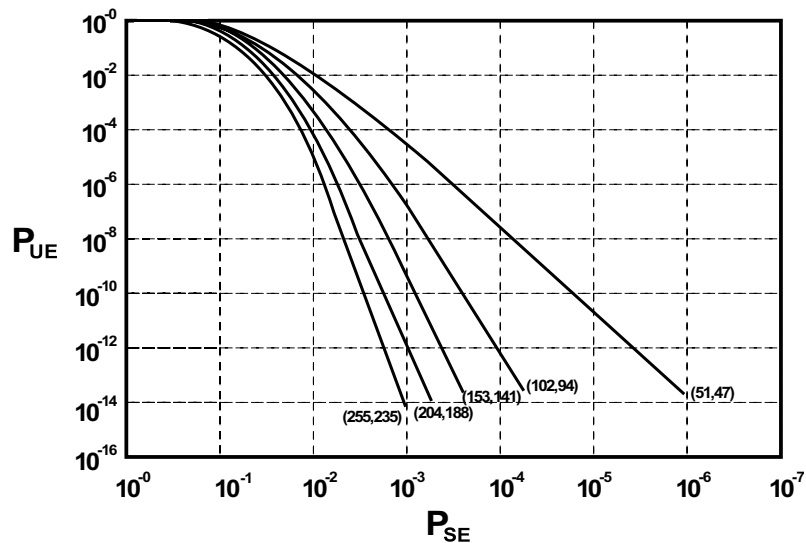


Figure 2 presents the performance of several RS codes of rate 92. The rate is kept constant by increasing the block length. For the RS(51,47) code with $m=8$, the block length is $8 \times 51 = 408$ bits. The RS(255,235) with $m=8$ has block length $8 \times 255 = 2040$ bits.

The important thing to notice here is for bit error rates below 10^{-2} , the performance curve for the RS(255,235) code has a very steep slope. The 10^{-2} value forms a threshold on the input P_{SE} for the satisfactory performance of the code. Long block length codes all exhibit this property. This steep slope is preferred for data communications, because large improvements in output P_{UE} are possible for small improvements in input P_{SE} .

Notice that for the RS(255,235) code, when the input P_{SE} is about 5×10^{-3} , the output P_{SE} is approximately 10^{-14} . This type of improvement is typical of the performance of the Reed-Solomon codecs offered in the PerFEC line. The PerFEC codecs allow an adjustable block length up to $n = 255$.

Figures 1 and 2 present the code performance in terms of P_{SE} and P_{UE} . P_{UE} is the probability of a symbol error and can be different than the bit error rate, BER . P_{UE} is the probability of an uncorrectable error in a data block. When an uncorrectable error occurs the integrity of the entire data block is lost. For this reason, the Corrected Bit Error Rate ($CBER$) is useful. The $CBER$ is the reciprocal of the expected number of correct bits between errors (See Section 4.0 of this primer.) The data in Figure 2 converted to $CBER$ is shown in Figure 3.

Figure 3 shows that $CBER$ is less than 10^{-17} when $P_{SE} = 10^{-3}$ for the RS(255,235) code. This says that if one in a thousand received symbols are in error, there will be on an average of 10^{17} bits between errors after decoding. Put another way, if the bit rate is 45 Mbps and the symbol error rate is 10^{-3} , with this code the time between errors will be greater than 70 years!

Figure 4 shows a parametric study of code performance (P_{UE} vs P_{SE}) for codes of different block lengths, for $t = 10$ and $t = 5$. Again, note the steep slope on the curves for P_{SE} greater than 10^{-2} . Also note the curve slope is steeper as the code rate decreases and as the redundancy, (t), increases. This is characteristic of all good codes.

Figure 3: Performance Curves in Terms of $CBER$ for RS Codes of Rate .92

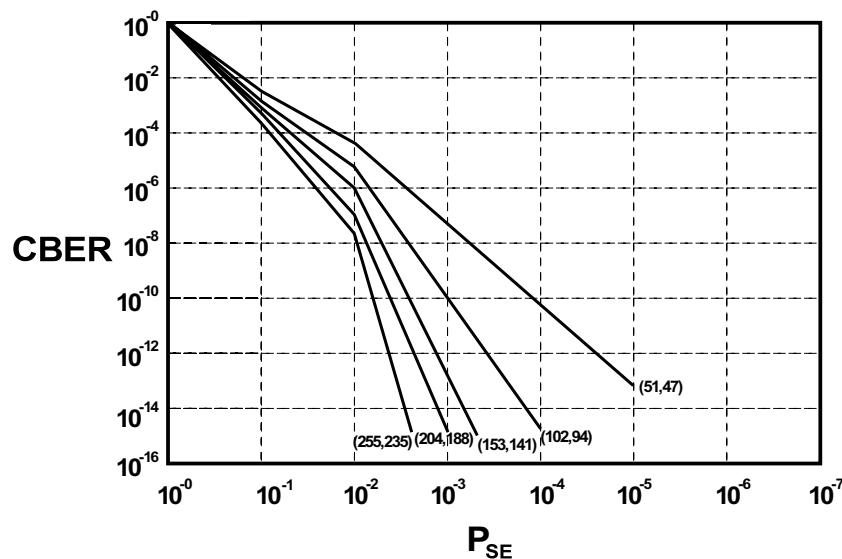
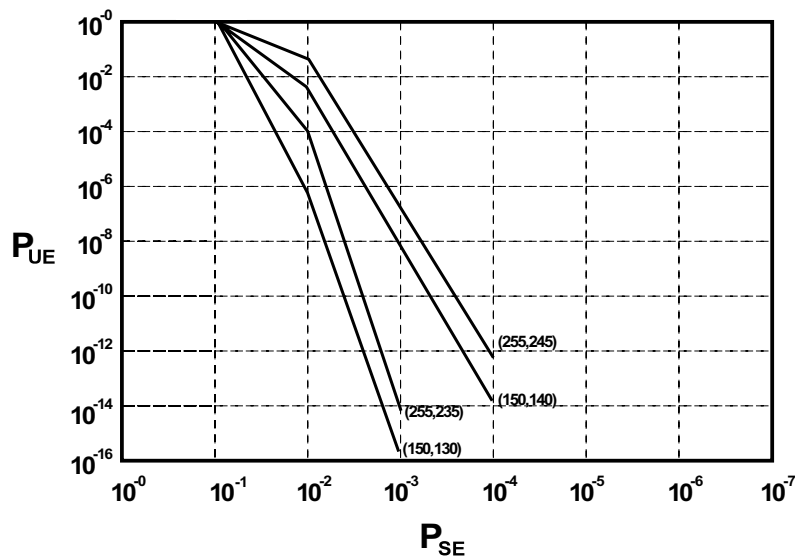


Figure 4: Performance Curves for RS Codes of Different Block Lengths, for $t=10$ and $t=5$



5.0 CHOOSING A CODE

5.1 ISSUES

In general a good code must:

- 1) have the ability to correct and detect the errors found in the channel
- 2) be suited to the noise environment of the channel
- 3) be efficient in the use of redundancy
- 4) have a coding and decoding algorithm which can be economically implemented using available technology

Two important issues arise when choosing a code for a given application:

- 1) What is the noise environment of the channel and what types of errors are expected in the system?
- 2) Of the available codes, which code or codes are well-suited to the noise environment and which ones can be implemented into the system with the best cost/performance tradeoffs?

5.2 MATCHING THE CODE TO THE CHANNEL NOISE

As stated in Section 2.4 there are three broad categories of errors in a communications channel:

- 1) Random errors - where the bit error probabilities are independent of each other, or nearly so
- 2) Burst errors - where the bit errors occur sequentially in time
- 3) Impulse errors - where large blocks of the data are full of errors

It is important for the code being used in a system design to be suited to the noise environment of the channel. Therefore, it is important to understand the applicability of the Reed-Solomon codes to each of these noise categories.

For random errors, two appropriate FEC choices are a convolutional code or a block code such as the Reed-Solomon code. Although the convolutional codes perform well with random errors, long distance block codes (block codes with large Hamming distance, like the PerFEC AHA4011, AHA4012 and AHA4013), often perform even better. A long block length averages out and randomizes the noise over that block. Long constraint length convolutional codes are not practical to implement so they cannot take advantage of this averaging property. Also, if there is uncertainty in the random noise model for a system channel, a large length block code is likely to be the best choice.

In a binary data channel, a burst error consists of consecutive errors in a string of received data. A burst error encompassing many bits is likely to show up as just a few symbols in error because RS codes deal with multi-bit “symbols” (8-bit bytes in the usual case). Consequently, RS codes are among the best codes for a burst-error environment.

Impulse errors can cause a catastrophic error in the communications system that is so severe as to be unrecognizable by FEC using any practical coding scheme. Thus, a retransmission of the message is usually requested if this is possible. A special feature of the AHA PerFEC line of codecs is the ability to recognize and flag an undecodable message. The system can be designed to depend on this flag to direct a retransmission of the message or cause some special handling of the erroneous message.

5.3 COST vs. PERFORMANCE

Block codes and specifically the Reed-Solomon codes used within the PerFEC line of codecs offered by AHA provide a competitive solution for practical FEC. The inherent power of Reed-Solomon block codes, and the simple single-chip solution provided by the VLSI implementation, combine to produce a superior cost/performance ratio.

6.0 CONCLUSIONS

Forward Error Correction (FEC) means that a digital system can detect and reconstruct an erroneous transmitted message at the receiver, without requesting a retransmission. The FEC system accomplishes this by analyzing the redundant data transmitted along with the message. Modern coding theory has devised block codes for FEC, notably the Reed-Solomon (RS) codes which are used within the AHA PerFEC family of VLSI integrated circuits. AHA has made the implementation of Reed-Solomon coding both economical and practical.

The ability of the RS codes to correct and detect both random errors and burst errors makes the RS codes among the most powerful EDAC codes in use today. The VLSI implementations of the RS codecs by AHA make sophisticated forward error correction available for any application.

6.1 REED-SOLOMON COPROCESSORS BY AHA

AHA provides a line of high performance programmable and non-programmable RS codecs, encoders, and decoders. The present PerFEC devices can support sustained data rates of up to 12.5 MBytes per second regardless of the number of errors in a block. These ICs incorporate patented custom designs including hundreds of thousands of CMOS transistors that maximize functionality and minimize power and chip size.

The AHA4011/12/13 are software programmable codes that can be used as sender end encoders or receiver end decoders or both on a time share basis. They can correct up to 20 erasures or 10 errors. These devices can be multiplexed for higher data rates.

7.0 ABOUT AHA

The AHA Products Group (AHA) of Comtech EF Data Corporation develops and markets superior integrated circuits, boards, and intellectual property cores for improving the efficiency of communications systems everywhere. AHA has been setting the standard in Forward Error Correction and Lossless Data Compression for many years and provides flexible and cost effective solutions for today's growing bandwidth and reliability challenges. Comtech EF Data is a wholly owned subsidiary of Comtech Telecommunications Corporation (NASDAQ" CMTL). For more information, visit: www.aha.com.

8.0 ADDITIONAL READING

Berlekamp, E., Peile, R., Pope, S., "The Application of Error Control to Communications," IEEE Communications Magazine, Vol. 25, No. 4, April 1987, pp 44-57. A very good introduction to coding for error correction. Much of the information for this application note is taken from this paper.

Bhargava, V., "Forward Error Correction Schemes for Digital Communications," IEEE Communications Magazine, Vol. 21, No. 1, January 1983, pp 11-19. A non-mathematical overview of coding theory.

Geisel, W.A., "Tutorial on Reed-Solomon Error Correction Coding", NASA Technical Memorandum No. 102162, August, 1990.

Reed, I.S. and Solomon, G., "Polynomial Codes Over Certain Finite Fields," J. Soc. Ind. Appl. Math., Vol. 8, pp. 300-304, and Math. Rev. Vol. 23B, P. 510, 1960.

Shannon, C.E., "A Mathematical Theory of Communication," Bell System Tech. Jour., vol 27, pp. 379-423 and 623-656, 1948.

Sklar, B., "A Structured Overview of Digital Communications- A Tutorial Review - Part I," IEEE Communications Magazine, Vol. 21, No. 5, August 1983, pp 5-17. An overview to the coding problem and the Shannon constraints.

Sklar, B., "A Structured Overview of Digital Communications- A Tutorial Review - Part II," IEEE Communications Magazine, Vol. 21, No. 7, October 1983, pp 6-21. A good introduction to both block and convolutional codes.