Computer Systems, Inc.
MERCURY

Reed-Solomon Block and Hop Encoder Components

FM3TR Waveform Reference Implementation

SDR Forum Contract

March 23, 2007

Revision 1.0

Table of Contents

1 Component Name

RsBlockEncoder, RsHopEncoder

2 Component Processing Summary

This document combines the RsHopEncoder and RsBlockEncoder descriptions. The main differences between the hop and block encoders are the symbol size (number of bits per symbol) and the length of the input sequence.
Reed-Solomon (RS) forward error correction coding (FEC) is a linear block code which introduces redundant information to a block of data such that the receiver can correct for errors introduced by receiver noise. Generally, the more redundancy added to the packet allows the receiver to correct more symbol errors, but at the expense of reduced data rates. Unlike convolutional decoders, RS FEC algorithms operate on blocks of data with bits grouped into symbols.

3 Where used

The Reed Solomon Encoders were used in all data waveforms (excluding voice).

4 Data Input and Output Ports

The Reed Solomon Encoders each have one uses and one provides data port. Both the input (Rs<Hop/Block>EncoderIn) and output (Rs<Hop/Block>EncoderOut) accept a sequence of signed octets.

5 Control Interfaces

The RsHopEncoder and RsBlockEncoder components inherit the control interfaces from CF::Resource.

6 Component SCA Properties

Aside from DLL_ENTRY_POINT and DLL_NAME, the RS Encoders contain no additional properties.

7 Component Attributes/Key Variables

Below is a list of several key variables to the RS encoding algorithm with a brief description of their purpose. Their values identify properties unique to each RS component. This table is the same as the one in the RS deocder documentation.

Variable	*Description*	*RsBlockEncoder*	*RsHopEncoder*
n	Number of output symbols	105	16
k	Number of input symbols	72	14

N	Maximum value for each symbol (?)	127	31
m	Number of bits for each symbol	7	5
t	Maximum number of symbols that can be corrected	16	1

The following variables are used in both the RsBlockEncoder and RsHopEncoder, and have a common purpose for both components.

<block/hop>_prim_poly	Integer array of length m+1 (see table above) with binary primitive polynomial coefficients
<block/hop>_gen_poly	Integer array of length n-k+1 (see table above) with m-bit symbols representing the generating polynomial.
genpoly	Galois Field polynomial produced from the generating polynomial
blockParities	Galois Field polynomial produced by calculating the remainder of dividing genpoly into the incoming data packet
encMsg	Encoded message, a concatenation of the input packet and blockParities

8 Processing Details

The Reed-Solomon FEC coding algorithm operates on blocks of symbols. The error-correcting capability of the component is dependent mainly upon the number of additional rundundant bits added to the data block [1]. The RsBlockEncoder and RsHopEncoder components differ in both the number of bits in each symbol as well as the length of the input and output blocks, however the underlying algorithm to add this redundancy is the same.

The RS FEC coding algorithm relies on Galois Fields which is a branch of abstract algebra dealing with operations on polynomials. The SCA components use an open-source C++ Galois Filed arithmatic library [2] for most of the signal processing necessary for the RS algorithms.

Processing for the encoders goes as such:

1. An input packet of a certain length (72 symbols for the RsBlockEncoder, 14 symbols for the RsHopEncoder) is converted to a GaloisFieldPolynomial, encMsg.

2. The `blockParities` polynomial is calculated from the remainder of dividing the generator polynomial into `encMsg`.
3. `blockParities` is added to `encMsg`, converted to a sequence of octets, and passed to the port.

9 References

[1] J. G. Proakis, *Digital Communications, Fourth Edition*, McGraw-Hill, Boston, 2001.

[2] [Online]. Available: *http://www.partow.net/projects/galois/*.