Mercury Computer Systems, Inc.

Reed-Solomon Block and Hop Decoder Components

FM3TR Waveform Reference Implementation

SDR Forum Contract

March 23, 2007

Revision 1.0

Table of Contents

1 Component Name

RsBlockDecoder, RsHopDecoder

2 Component Processing Summary

This document combines the RsHopDecoder and RsBlockDecoder descriptions. The main differences between the hop and block decoders 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 decoders were used in all data waveforms (not including voice).

4 Data Input and Output Ports

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

5 Control Interfaces

The RsBlockDecoder and RsHopDecoder components inherit the control interfaces from CF::Resource. The RsBlockDecoder also contains two identical interfaces for error control (MAC_ErrCtrl_Out, MAC_ErrCtrl_Out2).

6 Component SCA Properties

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

7 Component Attributes/Key Variables

Below is a list of several key variables to the RS decoding 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 enocoder documentation.

Variable	*Description*	*RsBlockDecoder*	*RsHopDecoder*
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 RsBlockDecoder and RsHopDecoder, 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
`syndromeZeroFlag`	Boolean value set true if all values of syndrome vector are zero
`decMsg`	Decoded message

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 RsBlockDecoder and RsHopDecoder 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 decoders goes as such:

1. An input packet of a certain length (105 symbols for the RsBlockDecoder, 16 symbols for the RsHopDecoder) is converted to a GaloisFieldPolynomial, `decMsg`.
2. The syndrome polynomial is calculated. From [1, p. 448], “the [syndrome vector] has components tha are zero for all parity check equations that are satisfied and nonzero for all parity check equations that are not satisfie.” This implies that if the syndrome vector is comprised solely of zeros, no errors were detected. If this is the case, the input packet may be passed without further processing. Otherwise, the errors must be located and corrected.
3. If the syndrome vector contains non-zero elements, errors exist in the input packet. Calculate the error locations, solve for the error magnitudes and correct the errors.

In addition to these functions, the RsBlockDecoder also contains a pair of error control ports to tell both the FileOutput_MAC_LLC and MAC_Receive components if too many errors exist on the packet. This allows erroneous packets to be retransmitted.