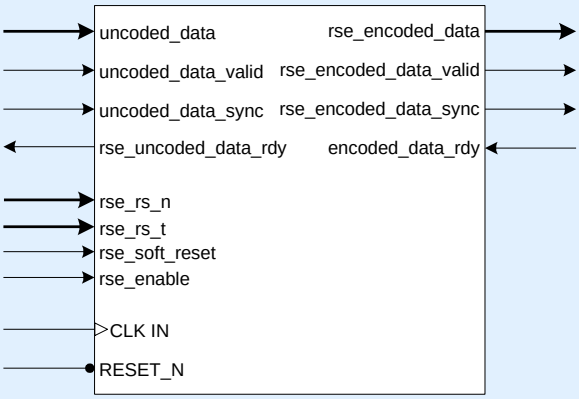
Reed Solomon Codec

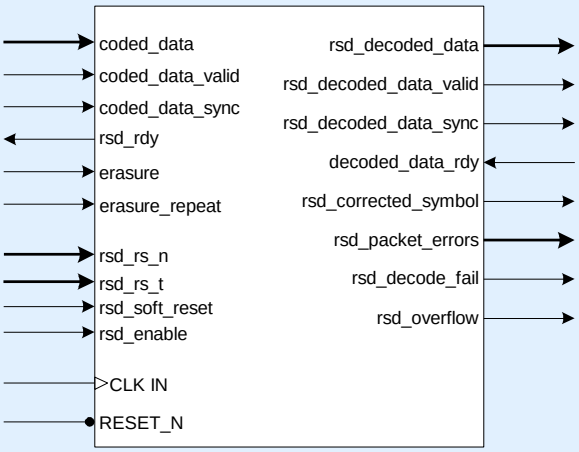
CMS0013

- Full errors and erasures Reed-Solomon CODEC using the Berlekamp Massey algorithm for efficient implementation.
- Can be synthesised for any Galois Field and Reed-Solomon code.
- Supports shortened codes.
- Supports Erasure processing as used in DVB-MPE and Variable-T (punctured) modes as used in IEEE 802.16.
- Variable-T mode allows register programming or synchronous "on-the-fly" loading for different values of N and T in a Reed-Solomon (N, N-2T) code.
- Full RDY-VALID handshaking on all data interfaces for simple interfacing with maximum throughput.
- Excellent performance in both FPGA and ASIC implementations.

Reed-Solomon Encoder



Reed-Solomon Decoder

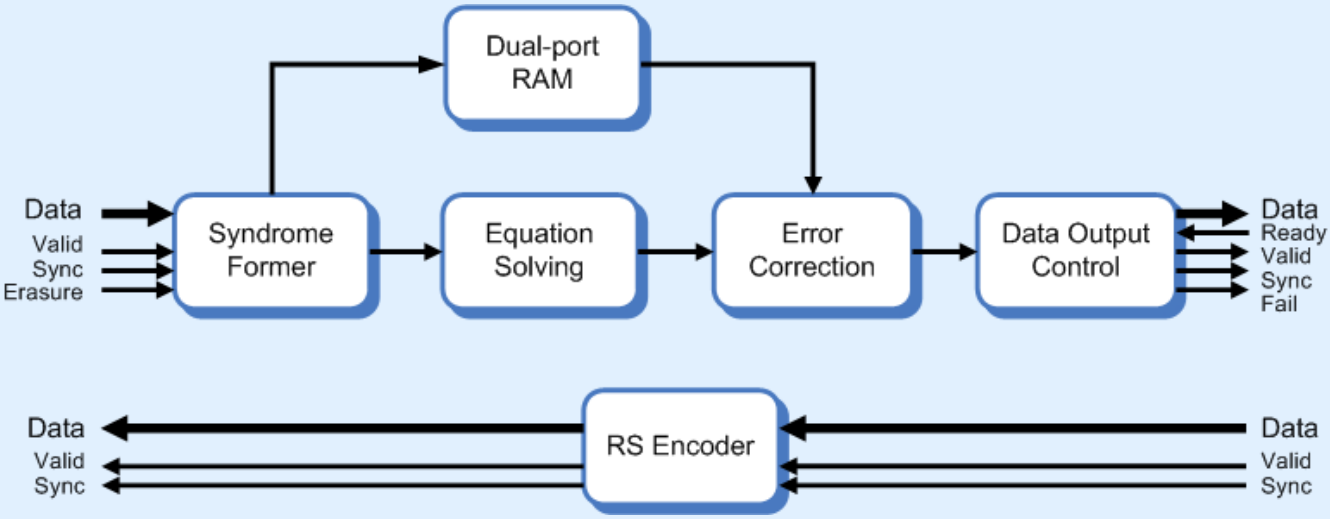


Contact information

Commsonic Ltd.
St. Johns Innovation Centre
Cowley Road
Cambridge
CB4 0WS
England

www.commsonic.com
sales@commsonic.com
tel. +44 1223 421845
fax +44 1223 421845

Block Diagram



Detailed Description

The Commsonic CMS0013 Reed Solomon Codec provides ultimate flexibility in its operation and build. The design uses the Berlekamp Massey algorithm in order to maximise speed and efficiency.

Synthesis Parameters

Setting up the Reed-Solomon codec is very simple. Firstly, it is necessary to set up the Galois field over which the code will operate using three parameters:

gal_bps. Bits per symbol (typically 7 or 8 for communications applications).

gal_alpha. Primitive root of the field (usually 2).

gal_field_gen_poly. Field Generator Polynomial (e.g. $x^8+x^4+x^3+x^2+1$ is usually used for GF{2^8}.

Once the field is defined, the Reed-Solomon code generating polynomial is configured with the parameters:

MAX_RS_T. The maximum number of symbol errors that can be corrected per codeword.

J0. The first power of alpha used in the code generator polynomial.

The code polynomial G(x) is then determined by:

$$G(x) = (1-\alpha^{J0})\ (1-\alpha^{J0+1})\ldots\ (1-\alpha^{J0+2*MAX\_RS\_T-1})$$

These parameters are set once in a VHDL package and are applicable to the full design. The field and code parameters are set by constants and fixed for a particular implementation of the block.

The remaining synthesis configuration options control the handling of erasures. Soft Erasure mode allows up to 2*T erasures to be corrected at any location. Variable-T mode provides erasure processing based on a pre-defined erasure pattern.

Each codeword contains 2T redundant symbols, allowing 2T items of information to be deduced. Error decoding requires two items of information per error: the location and the value. This allows a codeword with 2T check symbols to correct T errors.

Because the erasure locations are known, erasure decoding only has to deduce one item per erasure (the value). This allows the decoder to correct up to 2T erasures.

Soft Erasure Mode

BUILD_SOFT_ERASURES enables synthesis of full-erasure capability. Additional resources include three small dual-port RAMs used as erasure cache. This mode is appropriate for DVB-MPE decoding.

In this mode, the ERASE input is monitored and erased symbols are marked in the erasure cache.

Detailed Description (Cont'd)

Variable T Mode

BUILD_VARIABLE_T enables synthesis of 802.16 variable T mode, making it possible to adapt the correcting power for a particular application by adjusting the values of N and T using the registers, RS_N and RS_T.

RS codewords are encoded with 2*MAX_RS_T check symbols. When the real-time RS_T input is less than MAX_RS_T, 2*(MAX_RS_T – RS_T) check symbols are deleted (punctured) from the least-significant end of the polynomial. The punctured symbols are processed in the decoder as erasures.

For example, if the code instantiation selected is an RS(255, 239) code then the decoder is capable of correcting eight errors in every 255 symbols - since N=255 and T=8 are the maximum values permitted by this instantiation. Setting RS_N less than 255 will allow the same number of errors to be fixed in a shorter block size thereby increasing the correcting power of the code.

If bandwidth is an important consideration then setting RS_T to less than 8 can puncture the code. This reduces the correcting power of the code due to the redundancy reduction and therefore the bandwidth. Both N and T can be adjusted independently to give a wide range of coding powers and block sizes to suit any application.

An additional feature is the ability to accept new values on N and T on a packet-by-packet basis. The values of N and T are loaded with sync and valid to allow *on-the-fly* changes.

Principle I/O Description

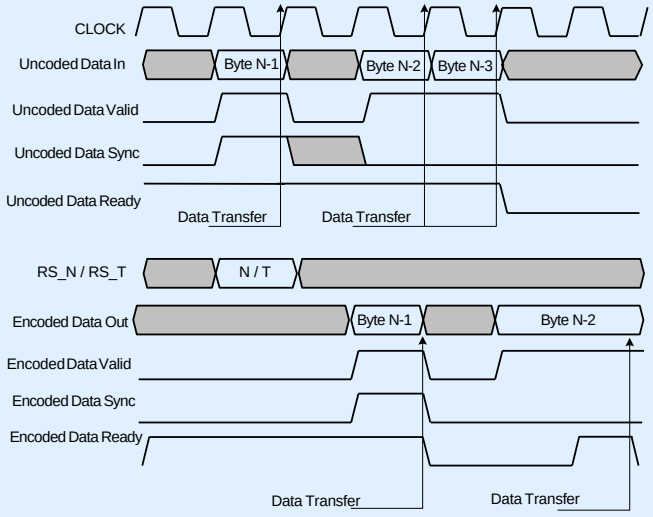
Decoder I/O	
coded_data	Reed-Solomon encoded symbols
coded_data_sync	RS block start flag
coded_data_valid	1 => coded_data, sync and erase flags are valid
erase	1 => incoming coded_data is to be erased
rsd_rdy	1 => decoder can accept coded_data. Transfer occurs when rdy = valid = 1
rsd_decoded_data	The decoded and corrected data output symbols
rsd_decoded_data_sync	Block start indicator for output blocks
rsd_decoded_data_valid	Strobe signal for output data and sync
decoded_data_rdy	This input indicates that the next block is ready to receive decoded data. Data transfer occurs when rdy = valid = 1
rsd_decode_fail	Indicates that the block output decode failed due to too many errors.
rsd_packet_errors	Number of errors fixed in the packet. Does not include corrected erasures
rsd_corrected_symbol	Indicates current output symbol was corrected by the RS decoder. Set = 1 for all corrected symbols including both errors and erasures
Decoder Registers	
rsd_rs_n	Number of symbols in the current codeword. Sampled at start of packet.
rsd_rs_t	Value of T for the current codeword. Sampled at start of packet.
rsd_enb	1 => Enable decoder. 0 => data is passed through without parity bytes.
rsd_soft_reset	1 => clear all internal storage
Encoder I/O	
uncoded_data	Uncoded data input symbols
uncoded_data_valid	1 => uncoded data and sync inputs are valid
uncoded_data_sync	1 => current input symbol is first in codeword
rse_uncoded_data_rdy	Encoder ready for new data byte. Data transfer occurs when rdy = valid = 1
rse_coded_data	The encoded data output symbols
rse_coded_data_sync	1 => current output symbol is first in codeword
rse_coded_data_valid	1 => encoded data and sync outputs are valid
rse_coded_data_rdy	Interface ready for new data byte. Data transfers when ready/valid are asserted
Encoder Registers	
rse_rs_n	Number of symbols in the current encoded block. Sampled at start of packet.
rse_rs_t	Value of T for the current codeword. Sampled at start of packet.
rse_enb	1 => Enable encoder. 0 => data is passed through without parity bytes.
rse_soft_reset	1 => clear all internal storage
Others	
clock	Clock input.
reset_n	Asynchronous active-low reset input.

Timing Diagrams

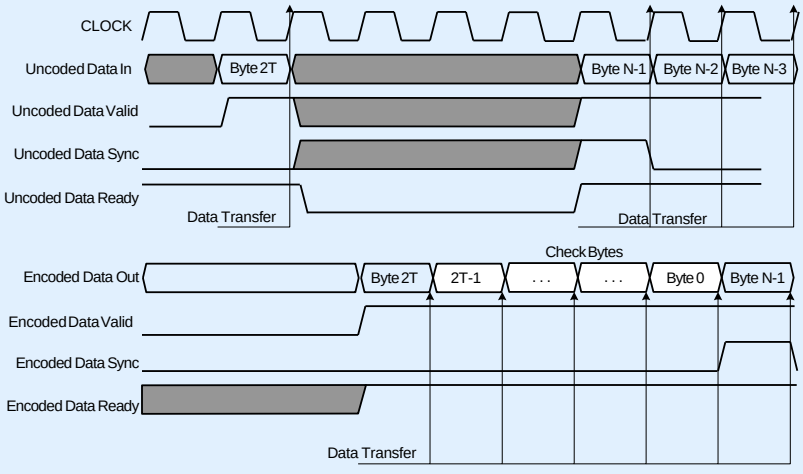
All data interfaces are implemented with RDY-VALID handshaking. This supports a simple interface in which each block may moderate the data flow to match its own data processing requirements. The data source asserts its VALID flag when it presents data. The data destination indicates availability using its RDY flag. Data is transferred on the clock cycle when both source and destination are available: RDY = VALID = 1.

By convention, RS codewords are transmitted MS byte first. The payload bytes K-1 down to 0 are transmitted first (K = N – 2T), followed by the check bytes numbered 2T-1 down to 0. The encoder asserts RDY = 0 while it inserts the 2T check bytes, as illustrated by the following diagrams:

RS Encoder packet start:

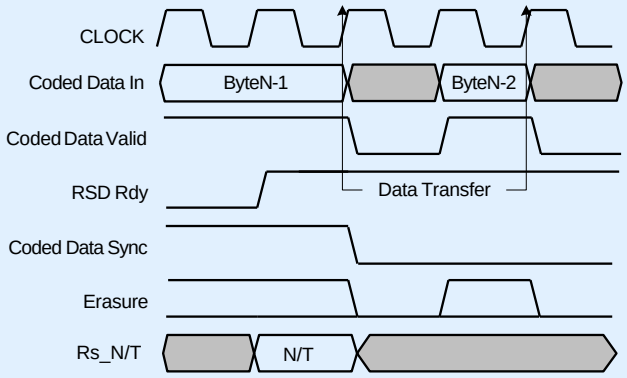


RS Encoder packet end:



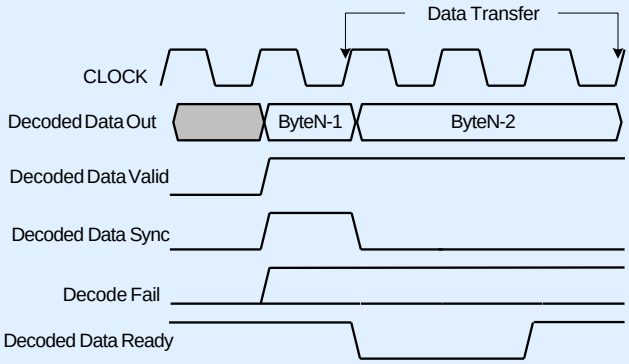
Timing Diagrams (Cont'd)

RS Decoder Data Input:



Note that the decoder will always take rsd_rdy low for (at least) one cycle at the end of each codeword. However, if the rest of the decoder is ready this will not cause an overflow.

RS Decoder Data Output:



EXAMPLE APPLICATIONS

DVB-MPE

The DVB-MPE specification defines an array of data with time-ordered bytes arranged in columns and RS codewords arranged as rows.

The array always has 255 columns and my have 256, 512, 768 or 1024 rows. The RS (255,191) code has T=32 and can correct up to 64 erases per codeword, yielding a nominal code rate of 3/4.

Data sections are protected by CRC. Received sections with CRC failures are erased. Unused data columns may be set to zero and not transmitted, increasing the effective correction power of the code. Likewise, some of the redundant check symbol columns may be punctured, reducing the correcting power of the code, along with its bandwidth overheads.

The rsd_corrected_symbol output indicates which symbols have been corrected by the decoder and need to be written back into the MPE frame buffer. Uncorrected output bytes may be ignored

The decoder also supports erasure retention. Because most codewords in an MPE frame will have the same erasure pattern, this mode allows the erasure polynomial to be retained from one codeword to the next, allowing considerable savings in execution time.

IEEE 802.16 Modem:

This requires substantial flexibility in the block size in order to support the variable modulation standards. The field used is always GF{2^8} generated with the polynomial $x^8+x^4+x^3+x^2+1$.

This gives the maximum value of N=255 by implication. The standard calls for a maximum T=8 and sets J0=0. These form the synthesis parameters for the block. In use, the codec should support variable block sizes and correcting powers as indicated.

Modulation	Code Rate	RS_N	RS_T
QPSK	1/2	32	4
QPSK	3/4	40	2
16 QAM	1/2	64	8
16 QAM	3/4	80	4
64 QAM	2/3	108	6
64 QAM	3/4	120	6

About Commsonic:

Commsonic is an IP and design services company that specialises in the development of ASIC, FPGA, DSP and board-level sub-systems for applications in wireless and wireline communications.

Our expertise is primarily in the gate- and power-efficient implementation of physical-layer (PHY) functions such as modulation, demodulation and channel coding, but we have extensive experience with all of the major elements of a modern baseband 'core' including medium access control (MAC), voiceband DSP, mixed-signal interfaces and embedded CPU and software.

Our services are available on a turn-key basis but they are usually provided as part of a support package attached to members of our expanding family of licensable IP cores.

Commsonic's IP spans the major Standards for cable, satellite and terrestrial digital TV transmission and includes high-performance, adaptable, single-carrier (QAM) and multi-carrier (COFDM) modulator and demodulator solutions for DVB-S/S2/DSNG, DVB-C/J.83/A/B/C, DVB-T/H, DVB-T2, ATSC and ISDB-T.

Commsonic's customers are typically semiconductor vendors and manufacturers of broadband transceiver equipment that demand leading-edge Standards-based or proprietary PHY solutions but don't have the internal resources necessary to get their products to market soon enough.

Commsonic Ltd.
St. Johns Innovation Centre
Cowley Road
Cambridge
CB4 0WS
England

www.commsonic.com
sales@commsonic.com
tel. +44 1223 421845
fax +44 1223 421845