Reed-Solomon Decoder

January 26, 1998 Product Specification

Integrated Silicon Systems, Ltd.

50 Malone Rd
Belfast BT9 5BS
Northern Ireland
Phone: +44 1232 664664
Fax: +44 1232 669664
E-Mail: info@iss-dsp.com
URL: www.iss-dsp.com

Features

- Programmable solution for high data rate Reed Solomon decoding
- Can be configured to support a range of standards involving Reed Solomon encoding e.g. European Telecommunication Standard 300 421 and 300 429
- Paramaterized architecture can be customized using the following parameters:
    - Symbol size
    - Number of errors corrected
    - Number of erasures corrected
    - Field polynomial
    - Generator polynomial
    - Statistics gathering
    - Interfaces
- Single decoder implementation supports any valid data block length
- Systematic code structure where each code word can be partitioned into original data block and appended parity symbols
- Decoder can process continuous or burst data
- Supports high speed (>300 Mbps) applications
- Symbol-wide inputs and outputs, clocked by a single symbol-rate clock
    - Design can be further optimized if higher rate clock (e.g. bit rate clock) is available
- Simple core interface allows easy integration into larger systems

AllianceCORE™ Facts

Core Specifics		
Device Family	Spartan	XC4000XL
CLBs Used	See Table 1	
IOBs Used		
System Clock fmax		
Device Features Used	RAM, 3-state buffers, Carry Logic, RPM	
Supported Devices/Resources Remaining		
See Table 1		
Provided with Core		
Documentation	Core Documentation	
Design File Formats	.ngd or XNF netlist VHDL Source RTL available extra	
Constraint Files	Timespec, .cst and.tnm files	
Verification Tool	Synopsys VSS	
Schematic Symbols	None	
Evaluation Model	None	
Reference designs & application notes	None	
Additional Items	Bit-accurate C Model is available	
Design Tool Requirements		
Xilinx Core Tools	Alliance 1.3	
Entry/Verification Tool	Synopsys Design Compiler Synopsys VSS	
Support		
Support provided by ISS.		

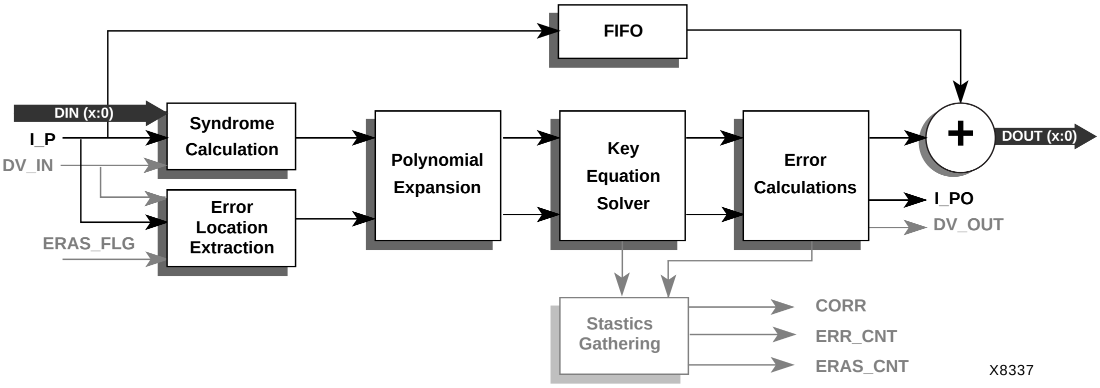

Figure 1: Reed-Solomon Decoder Block Diagram - clock and reset signals omitted for clarity

Table 1: Example Encoder Implementations

	Example #1	Example #2	Example #3
Correctable Errors	8	4	4
Bits/Symbol	8	8	8
Erasure Support?	No	No	No
Statistics Gathering?	No	Yes	Yes
Xilinx Device	XC4036XL-1	XCS40-3	XC4020E-2
CLBs Used	941	740	740
IOBs Used	19	24	24
CLKIOBs Used	1	1	1
System Clock fmax	40 MHz	20 MHz	20 MHz
Data Rate	320 Mbps	160 Mbps	160 Mbps
CLBs Remaining	355	44	44
I/O Remaining	268	200	200

Applications

- Digital video and audio broadcast.
- Digital satellite broadcast.
- Data storage and retrieval systems (e.g. hard disks, CD-ROM etc.).

General Description

The Reed Solomon decoder is a Xilinx FPGA-based core for systems where data error detection/correction is required. The core implements the full functionality of a general Reed Solomon decoder and all necessary interface circuitry using a single Xilinx FPGA.

The source code version of the core is extremely flexible due to its parameterized design, and can be rapidly configured for a wide variety of applications. The AllianceCORE facts table shows density and performance metrics for a specific implementation example. ISS will customize and provide a Xilinx-specific implementation of the core tailored to the needs of your application, (see Ordering Information).

Polynomials

A typical Code Generator Polynomial for the decoder is:

$g(x) = (x +\alpha^0)(x + (^1)(x +\alpha^2)$ …..$(x +\alpha^{15})$, where $(=\alpha 02_{HEX}$ and the number of parity symbols is 16.

A typical Field Generator Polynomial is:

$p(x) = x^8 + x^4 + x^3 + x^2 + 1$, where the Galois Field used is $GF(2^8)$.

Functional Description

The Reed-Solomon decoder core is partitioned into modules as shown in Figure 1, and described below.

Syndrome Calculation

The syndrome calculator accepts the received symbols. It considers the symbol values as polynomial coefficients and determines if the series of symbols contained in a data block form a valid code word for the particular Reed-Solomon code chosen. It evaluates the polynomial for 2t syndrome values (where t is the number of correctable random errors) and detects whether the evaluations are zero (the data block is a code word) or non-zero (the data block is not a code word). Any block that is not a code word has been affected by errors.

Erasure Location Extraction

Erasures are errors in known locations in the received data block, and are flagged by other system functions such as the receiver demodulator. Since the location of an erasure is known, the Reed-Solomon decoder only needs to determine the magnitude of the error. This is unlike a random error where both error location and magnitude are calculated. Therefore, for a given correction power, twice as many erasures as random errors can be corrected.

A combination of erasures and random errors can be corrected provided the combination does not exceed the correction power of the code, which is directly related to the number of parity symbols added to a data block.

Erasure symbols are flagged using the ERAS_FLG input, and the erasure location extraction block produces a polynomial representing erasure locations.

Polynomial Expansion

The syndrome and erasure polynomials are combined in this block to produce a single unified polynomial that can be processed by the key equation solver.

Key Equation Solver

The key equation relates the combined syndrome/erasure polynomial with two error polynomials. The first is the error locator polynomial, which gives information about which symbols in the code word are in error. The second is the error evaluator polynomial which, when further processed by the error calculation circuit, yields information regarding the magnitude of each error. The key equation solver is responsible for the derivation of the locator and evaluator polynomials.

Error Calculation

This component uses the error locator and evaluator polynomials to compute the error values for each symbol in the received data block. As each error value is computed, the appropriate received symbol is fetched from a code word FIFO, which buffers the received symbols during the decoding process. Each error value is simply added to the received symbol to produce the corrected symbol. (Note that addition and subtraction are identical in the Finite Field arithmetic used in Reed-Solomon decoding.)

Statistics Gathering (Optional)

This block receives information from the key equation solver and the error calculator. It then computes statistics regarding the degree of data corruption caused by the transmission channel and the success of the decoder in correcting this corruption. Typical statistical functions include:

- Random error and erasure counts
- Flags to indicate uncorrectable data blocks
- Computation of bit error rates

This block along with the signals DV_IN, DV_OUT, CORR, ERR_CNT, and ERAS_CNT are optional. ISS can customize and provide this for specific applications depending upon your need.

Clock

All flipflops in the R-S decoder core operate on the rising edge of the input symbol rate clock CLK. Data inputs are latched and outputs generated on the rising edge of the clock. CLB count of the core can be reduced if a higher clock rate, such as the bit rate clock, is available. In this case some of the hardware functions within the decoder may be clocked at the higher rate and reused multiple times per symbol period, thus reducing the overall hardware requirements.

Latency

The latency of the Reed-Solomon decoder is heavily dependent on:

- Data block length
- Number of correctable errors
- Whether uncorrectable data blocks are to be flagged
- Degree by which the clock rate exceeds the symbol rate

The syndrome calculation and determination of block correctability usually takes one data block length each since all symbols in a block must be examined. Therefore, an uncorrectable signal is valid once the entire block has passed through the statistics gathering circuit.

To ensure that the uncorrectable signal emerges from the decoder simultaneous to the first data block symbol, the data must be delayed by one block length. Hence, the Syndrome Calculator and Statistics Gathering blocks account for two data block lengths of latency.

The remaining components contribute variable latency based on the clock rate and the number of correctable errors. It should be noted that these are only general guidelines for decoder latency and, if low latency is a priority, it can be achieved by trading off other design characteristics e.g. area. The latency of the Reed Solomon decoder example designs are shown in Table 1.

Pinout

The pinout of the Reed-Solomon decoder has not been fixed to specific FPGA I/O, allowing flexibility with a users application. Signal names are given in Table 2.

Input Interface

The input interface to the decoder consists of a bit parallel symbol data input, DIN, information/parity flag, I_P, input data valid flag, DV_IN, erasure indicator flag, ERAS_FLG, the symbol rate clock, CLK and an asynchronous reset, RES. All inputs are sampled on the CLK rising edge and the control inputs refer to the current input symbol.

I_P together with DV_IN identify valid input information and parity symbols. At the start of a block, both I_P and DV_IN are asserted to indicate valid information symbols at the data input. After the information symbols have been received the I_P input is deasserted, while DV_IN is held asserted, to indicate valid parity symbols on the input.

During periods when no valid input symbols are being input, DV_IN is deasserted. The erasure indicator input flag, ERAS_FLG, is asserted to indicate erasures in the input data stream that have been identified by, for example, a Viterbi decoder function in the system.

Output Interface

The output interface of the decoder consists of a bit parallel symbol data output, DOUT, information/parity flag, I_PO, output data valid, DV_OUT, correctable indicator, CORR, errors corrected count, ERR_CNT and erasures corrected count, ERAS_CNT. All outputs are clocked out on the rising edge of the symbol clock, CLK.

I_PO and DV_OUT are asserted when the first symbol of a block is output to indicate valid information symbols. I_PO is deasserted when parity symbols are output, while DV_OUT remains asserted. DV_OUT is deasserted at the end of the block if the start of the next block is not available, otherwise it is remains asserted.

If a block can be corrected, then CORR is also asserted at the first symbol of the block and remains in this state for the duration of the block output, and is deasserted with DV_OUT. If CORR is not asserted then the decoder was unable to correct the block.

ERR_CNT indicates the number of errors that were corrected for a block, while ERAS_CNT indicates the number of erasures that were corrected for a block. These signals go active at the first output symbol and are maintained for the duration of the block output from the decoder.

Table 2: Core Signal Pinout

Signal	Signal Direction	Description
Input Interface Signals		
DIN[x:0]	Input	Input data symbols
I_P	Input	Information / parity input indicator, which identifies the current input symbol as information symbol or parity symbol. 1 = information symbol 0 = parity symbol
ERAS_FLG*	Input	Optional input flag that marks the position of erasures in the input data stream.
CLK	Input	Symbol clock; uses 1 FPGA CLKIOB pin
RES	Input	Asynchronous reset, active high
Output Interface Signals		
DOUT[x:0]	Output	Data output symbols
I_PO	Output	Output data information /parity flag 1= information 0 = parity
Statistics Gathering Signals (Optional)		
DV_IN	Input	Data valid input. When 1 indicates valid input data.
DV_OUT	Output	Output data valid flag, active high
CORR	Output	Correctable output flag, which is output concurrent with the first symbol of a block and indicates that the block was correctable. 1 = correctable 0 = non-correctable
ERR_CNT	Output	Number of errors corrected in block
ERAS_CNT	Output	Number of erasures corrected in blocks

* Only relevant if erasure support is required.

Verification Methods

The core has been fully tested across a wide range of parameter settings, using input sequences compatible with many international digital video, audio and satellite standards.

Recommended Design Experience

Familiarity with system standards relevant to the particular application is assumed. A basic understanding of Reed Solomon decoding is useful, but not essential.

Available Support Products

A bit accurate C model of the Reed-Solomon decoder is available to assist in functional verification and system integration.

ISS also supplies a complete line of peripheral cores that can be integrated with the Reed Solomon Encoder and Decoder cores for a complete forward error correction system using Xilinx FPGAs. For more information, contact ISS directly regarding:

- Block and convolution interleavers and deinterleavers
- Scramblers and descramblers
- Sync detection and insertion
- Convolution encoders
- Viterbi decoders

Ordering Information

The Reed Solomon Decoder is available for purchase directly from ISS. The implementation will vary depending upon your application. To determine what is required for your system, fill out and fax the attached Implementation Request Form to ISS at +44 1232 669664.

Related Information

European Telecommunications Standards Institute

For information on European digital broadcasting systems standards contact:

European Telecommunications Standards Institute
6921 Sophia Antipolis Cedex
France
Phone: +33 92 94 42 00
Fax: +33 93 65 47 16

Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95214
Phone: 408-559-7778
Fax: 408-559-7114
URL: www.xilinx.com

For general Xilinx literature, contact:

Phone: 800-231-3386 (inside the US)
408-879-5017 (outside the US)
E-mail: literature@xilinx.com

For AllianceCORE specific information, contact:

Phone: 408-879-5381
E-mail: alliancecore@xilinx.com
URL: www.xilinx.com/products/logicore/alliance/tblpart.htm

Reed-Solomon Implementation Request Form

To: Integrated Silicon Systems, Inc.
FAX: +44 1232 669664
E-mail: info@iss-dsp.com

From: ____

Company: ____________________________________

Name: _______________________________________

Address: _____________________________________

Country: _____________________________________

Phone: _______________________________________

Fax: ____

E-mail: ______________________________________

ISS configures and ships Xilinx netlist versions of the Reed Solomon cores customized to your specification. Please fill out and fax this form so ISS can respond with an appropriate quotation that includes performance and density metrics for the target Xilinx FPGA.

Implementation Issues

1. Indicate your exact requirement:
   ______ Reed-Solomon encoder
   ______ Reed-Solomon decoder
   ______ Interleaver
   ______ Deinterleaver

2. Maximum symbol rate: _______

3. Number of bits per symbol: _______

4. Indicate clock availability:
   ______ Bit clock
   ______ Symbol Clock
   ______ Both

5. Code generator polynomial (e.g. $(x+a^1)(x+a^2)$...)

6. Field generator polynomial (e.g. $x^8+x^4+x^3+x^2+1$)

7. Number of errors: ______

8. Codeword length/format (e.g. 204, 188):

9. Erasure support required:
   Yes ______ No ______

10. Indicate error statistics reporting requirement(s):
    ______ Signal to show position of corrected errors
    ______ Signal to show position of uncorrected blocks
    ______ Other (please specify)

11. Indicate required control signals:
    ______ low to high transition at start of codeword, high to low transition at end of codeword
    ______ low to high transitional start of codeword, high to low transition at start of parity symbols
    ______ delayed control signals available at output
    ______ other (please specify)

12. Please specify any additional requirements:

13. If interleaver and/or deinterleaver functions are required, specify maximum depth of interleaving:

14. If the interleaver and/or deinterleaver are required is the depth of interleaving to be programmable and if so, to what degree?

Business Issues

15. Indicate timescales of requirement:
    a. date for decision ______
    b. date for placing order ______
    c. required delivery date ______

16. Indicate your area of responsibility:
    ______ decision maker
    ______ budget holder
    ______ recommender

17. Has a budget been allocated for the purchase?
    Yes ______ No ______

18. What volume do you expect to ship of the product that will use this core? ______

19. What major factors will influence your decision?
    ______ cost
    ______ customization
    ______ testing
    ______ implementation size

20. Are you considering any other solutions?