Implementation of Berlekamp Algorithm for Error Detection and Correction of Multiple Random Errors Using Reed-Solomon Codes

P. Chiranjeevi[1] (M.Tech), D. Ramadevi[2], Asst.Prof. K. Jeevan Reddy[3], HOD

[1, 2, 3,] Department of ECE, Teegala Krishna Reddy Engineering College/JNTU, India

Abstract:

In the communication systems, RS codes have a widespread use to provide error protection. For burst errors and random errors, RS code has become a popular choice to provide data integrity due to its good error correction capability. This feature has been one of the important factors in adopting RS codes in many practical applications such as wireless communication system, cable modem, computer memory and ADSL systems. Reed Solomon codes are an important sub class of non-binary BCH codes. These are cyclic codes and are very effectively used for the detection and correction of burst errors. Galois field arithmetic is used for encoding and decoding of reed Solomon codes. The design experience will be formulated to form the complete design methodology of the FEC modules at the register-transfer level (RTL). Then we incorporate the knowledge into our RS code generator design flow.

Keywords: *RS codes, random errors, BCH codes, Galois Field, ADSL*

I. Introduction

Digital communication system is used to transport information bearing signal from the source to a user destination via a communication channel. The information signal is processed in a digital communication system to form discrete messages which makes the information more reliable for transmission. Channel coding is an important signal processing operation for the efficient transmission of digital information over the channel. It was introduced by Claude E. Shannon in 1948 by using the channel capacity as an important parameter for error free transmission. In channel coding the number of symbols in the source encoded message is increased in a controlled manner in order to facilitate two basic objectives at the receiver one is Error detection and other is Error correction. Error detection and Error correction to achieve good communication is also employed in devices. It is used to reduce the level of noise and interferences in electronic medium.

The aim of the project is to correct multiple random errors and burst errors that are occur during the transmission of the information by using Reed Solomon codes. The proposed code is designed using verilog coding and the results demonstrate that the reed Solomon codes are very efficient for the detection and correction of burst errors.

II. Proposed Product Codes

The Reed Solomon code is an algebraic code belonging to the class of BCH (Bose-Chaudhry-Hocquehen) multiple burst correcting cyclic codes. The Reed Solomon code operates on bytes of fixed length. Given m parity bytes, a Reed Solomon code can correct up to m byte errors in known positions (erasures), or detect and correct up to m/2 byte errors in unknown positions. This is an implementation of a Reed Solomon code with 8 bit bytes, and a configurable number of parity bytes. The maximum sequence length (code word) that can be generated is 255 bytes, including parity bytes. In practice, shorter sequences are used.

A typical system is shown in the figure 4.1 below:

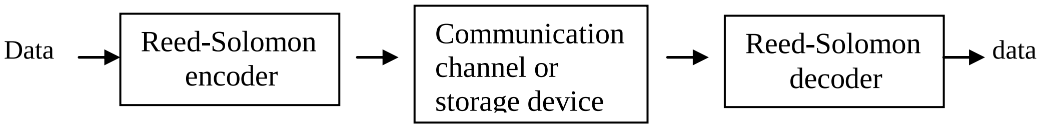

Fig 1 Typical RS encoder-decoder system

The Reed-Solomon encoder takes a block of digital data and adds extra "redundant" bits. Errors occur during transmission or storage for a number of reasons. The Reed-Solomon decoder processes each block and attempts to correct errors and recover the original data. The number and type of errors that can be corrected depends on the characteristics of the Reed-Solomon code. A Reed-Solomon code is specified as RS (n, k) with s-bit symbols.

This means that the encoder takes k data symbols of s bits each and adds parity symbols to make an n symbol codeword. There are n-k parity symbols of s bits each. A Reed-Solomon decoder can correct up to t symbols that contain errors in a codeword, where 2t = n-k.

The following figure 4.2 shows a typical Reed-Solomon codeword (this is known as a Systematic code because the data is left unchanged and the parity symbols are appended):

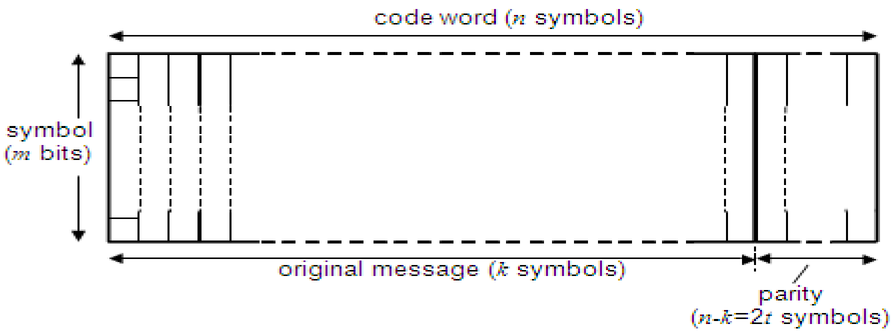

Fig 2 RS codeword format

1. Reed Solomon Encoder

Consider a Reed-Solomon code RS (255,247) with 8-bit symbols. Each codeword contains 255 code word bytes, of which 247 bytes are data and 8 bytes are parity. For this code: n = 255, k = 247, s = 8, 2t=8, t=4. The decoder can correct any 4 symbol errors in the code word: i.e. errors in up to 4 bytes anywhere in the codeword can be automatically corrected.

The k information symbols that form the message to be encoded as one block can be represented by a polynomial M(x) of order k-1, so that:

M(x) = Mk-1xk-1 + ... + M1x + M0

Where each of the coefficients Mk-1, ..., M1, M0 is an m-bit message symbol, that is, an element of GF(2m). Mk-1 is the first symbol of the message.

To encode the message, the message polynomial is first multiplied by xn-k and the result divided by the generator polynomial, g(x). Division by g(x) produces a quotient q(x) and a remainder r(x), where r(x) is of degree up to n-k-1. Thus:

$$\frac{M(x)\times x^{n-k}}{g(x)} = q(x) + \frac{r(x)}{g(x)}$$

The following diagram shows an architecture for a systematic RS(255,247) encoder:

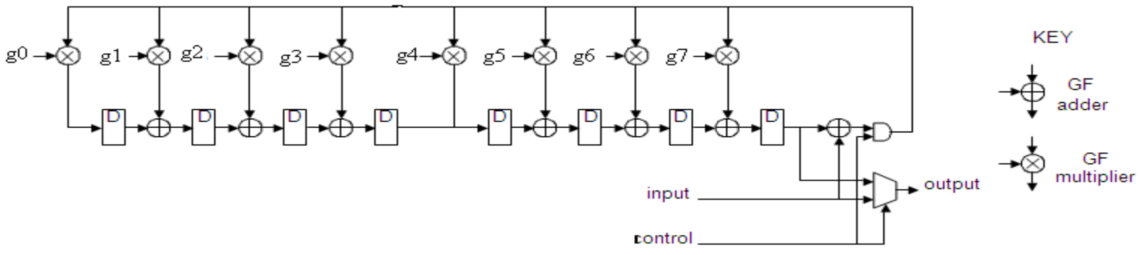

Fig 3 RS (255,247) Encoder

Reed-Solomon codes may be shortened by making a number of data symbols zero at the encoder, not transmitting them, and then re-inserting them at the decoder.

Reed Solomon Decoder

The Reed Solomon decoder tries to correct errors and/or erasures by calculating the syndromes for each codeword. Based upon the syndromes the decoder is able to determine the number of errors in the received block. If there are errors present, the decoder tries to find the locations of the errors using the Berlekamp Massey algorithm by creating an error locator polynomial.

The roots of this polynomial are found using the Chien search algorithm. Using Forney's algorithm, the symbol error values are found and corrected. For an RS (n, k) code where n - k = 2t, the decoder can correct up to t symbol errors in the code word. Given that errors may only be corrected in units of single symbols (typically 8 data bits).

The functional flow chart of the Reed Solomon Decoder is shown in the figure 5.1 below:

The blocks of the Reed Solomon Decoder are:

1) Syndrome calculation block
2) Error Location Determination block
3) Error value calculation block
4) Error correction block

The purpose of the decoder is to process the received code word to compute an estimate of the original message symbols.

The RS decoder block is shown in the figure below:

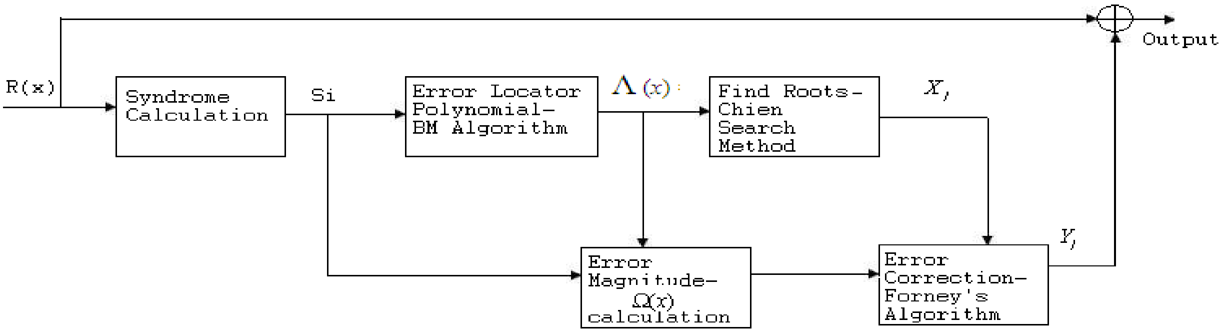
Output
R(x)
Syndrome Calculation
Si
Error Locator Polynomial– BM Algorithm
$\Lambda(x)$
Find Roots- Chien Search Method
X_i
Error Magnitude– $\Omega(x)$ calculation
Error Correction– Forney's Algorithm
Y_i

Fig 4. RS Decoder block

There are three main blocks to the decoder first is syndrome generator, then Barlekamp Massey algorithm and the Chien/Forney block. The output of the Chien/Forney block is an estimate of the error vector. This error vector is then added to the received codeword to form the final codeword estimate. Note that the error value vector Y comes out of the Chien/Forney block in reverse order, and it must pass through a LIFO/FIFO block before it is added to the received codeword R(x).

III. Performance

The proposed code can correct up to 16 symbol errors by column wise decoding. In the proposed code since two shortened RS codes are used it can correct up to 16 symbol errors.

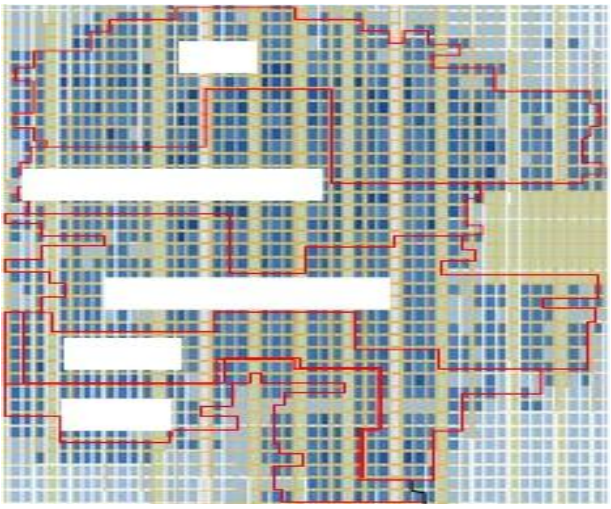

Fig 5 Layout of FPGA chip for the proposed RS code

IV. Results

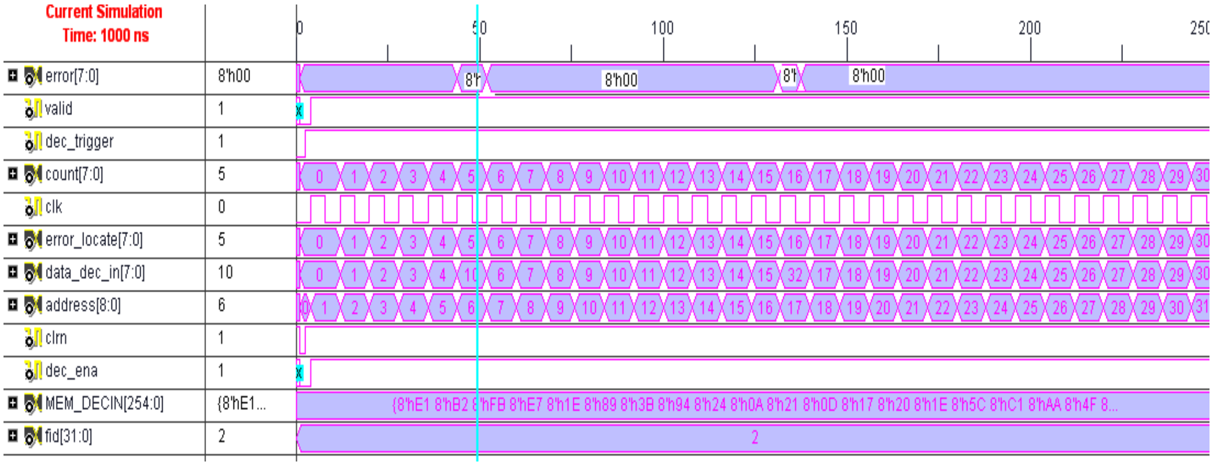

Fig 6 Result of RS(255,231) code

V. Conclusion

In this thesis, error detection and correction techniques have been used which are essential for reliable communication over a noisy channel. The effect of errors occurring during transmission is reduced by adding redundancy to the data prior to transmission. The redundancy is used to enable a decoder in the receiver to detect and correct errors. Cyclic Linear block codes are used efficiently for error detection and correction. The encoder splits the incoming data stream into blocks and processes each block individually by adding redundancy in accordance with a prescribed algorithm. Likewise, the decoder processes each block individually and it corrects errors by exploiting the redundancy present in the received data.

In this work, architectures were modeled using HDL and the functional simulation was carried out using Xilinx ISE 12.4 simulator. The compilation, synthesis and place and route, timing are done with RTL and SOC Encounter of Cadence Tool. Synthesis of the architectures was carried out on SPATAN 3 board with XC3S4000-4PQ208 target devices.

References

[1] M. Mehnert, D. F. von Droste, and D. Schiel, “VHDL Implementation of a (255, 191) Reed Solomon Coder for DVB-H,” *IEEE 10th International Symposium on Consumer Electronics (ISCE), pp. 1-5*, 2006.

[2] S. B. Wicker and V. K. Bhargava, eds., Reed-Solomon Codes and their Applications. *New York: IEEE Press*, 1994.

[3] S. Lin and D. J. Costello, *Error Control coding (2nd edition)*. Upper Saddle River, NJ: Prentice-Hall, 2004.

[4] R. Koetter and A. Vardy, \Algebraic Soft-Decision Decoding of Reed-Solomon Codes," *IEEE Transactions on Information Theory, vol. 49, no. 11*, pp. 2809{2825, 2003.

[5] V. Guruswami and M. Sudan, \Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes," *IEEE Transactions on Information Theory, vol. 45, no. 6*, pp. 1757{1767, 1999.

[6] R. J. McEliece, \The Guruswami-Sudan Decoding Algorithm for Reed-Solomon Codes," *Tech. Rep. 42-153, JPL Interplanetary Network Progress Report*, 2003.