International Journal of Engineering Studies.
ISSN 0975-6469 Volume 9, Number 1 (2017), pp. 1-9

http://www.ripublication.com

ECC design strategy targeting area optimization in reconfigurable device (FPGA) using VLSI Technique

Nitin S. Sonar[1]
[1] *Research Student, Department of Electronics, Shivaji University, Kolhapur, India*

R.R. Mudholkar[2]
[2] *Professor, Department of Electronics, Shivaji University, Kolhapur, India.*

Abstract

Forward error correction (FEC) plays an important role in the field of telecommunication and information theory as it improves the capacity of a channel. It has been observed that Reed Solomon Error Corrector is a powerful method for error detection and correction. Error Correction codes (ECC) are a mean of including redundancy in a stream of information bits to allow the detection and correction of symbol errors during transmission. This research paper describes new design strategy targeting area optimization in reconfigurable device (FPGA). Area optimization is a major issue as smaller components allow for better system adaption, which is an expected feature of reconfigurable system for space applications. The approach is applied in the design stage of a component for the communications module of an on-board computer system. The chosen component is a Reed-Solomon encoder, which has been implemented using a Hardware Description Language (VHDL) targeting an FPGA platform. The research work investigates traditional alternatives for the encoder implementation, introduces the new architecture of the error corrector behind the proposed approach, describes the design process and discusses the area figures reached by the new design.

Keywords: Forward error correction (FEC), Error Correcting Code (ECC), Reed-Solomon Codes, FPGA.

INTRODUCTION

In order for the transmitted data to be corrected in the event that it acquires errors, it has to be encoded. The receiver uses the appended encoded bits to determine and correct the errors upon reception of the transmitted signal. The number and type of errors that are correctable depend on the specific Reed-Solomon coding scheme used. [1-4]

Error control coding techniques are based on the addition of redundancy to the information message according to a prescribed rule thereby providing data at higher bit rate. This redundancy is exploited by decoder at the receiver end to decide which message bit actually transmitted. Reed-Solomon codes are an important sub–class of non-binary Bose-Chaudhuri-Hocquenghem (BCH) codes.

Reed Solomon codes are efficient and non-binary error correcting codes. In computing, a linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value. The mathematics of a cyclic redundancy check, used to provide a quick check against transmission errors, is closely related to those of an LFSR [5-10].

Generally the encoder is implemented using general Linear Feedback Shift Register (LFSR), but in present design of an encoder Galois LFSR is used to make the system more efficient. In the Galois configuration, when the system is clocked, bits that are not taps are shifted one position to the right unchanged. The taps, on the other hand, are XOR'd with the output bit before they are stored in the next position. The new output bit is the next input bit. Note that Galois LFSRs do not concatenate every tap to produce the new input (the XORing is done within the LFSR and no XOR gates are run in serial, therefore the propagation times are reduced to that of one XOR rather than a whole chain), thus it is possible for each tap to be computed in parallel, increasing the speed of execution [11-14].

Thus because of the use of Galois LFSR the propagation times are reduced. This increases the speed of execution. Galois field arithmetic is used for encoding and decoding of Reed-Solomon codes. Galois field multipliers are used for encoding the information block. The encoder attaches parity symbols to the data using a predetermined algorithm before transmission. At the decoder, the syndrome of the received code word is calculated. VHDL implementation creates a flexible, fast method and high degree of parallelism for implementing the Reed Solomon codes.

The new design of Reed-Solomon code presented here has following important features,

- Area optimization in reconfigurable device FPGA.
- Fully synchronous design using a single clock.
- Detects condition when the number of errors is too high to be corrected.
- Supports different Reed-Solomon coding standards.

- System offers status and performance monitoring of signals.
- Low cost VLSI architecture with less complexity.
- Supports continuous input data with no gaps between blocks.
- Continuous, high-speed, time-domain Reed-Solomon decoding architecture.

DECODER

A Reed-Solomon code word has 2t syndromes that depend on the errors corrected not on the transmitted code word. The syndromes can be calculated by substituting the 2t roots of the generated polynomial g(x) into r(x), the received code word. The error-correcting codes become more efficient as the block size increases because the effect of noise becomes less than that for small block size.

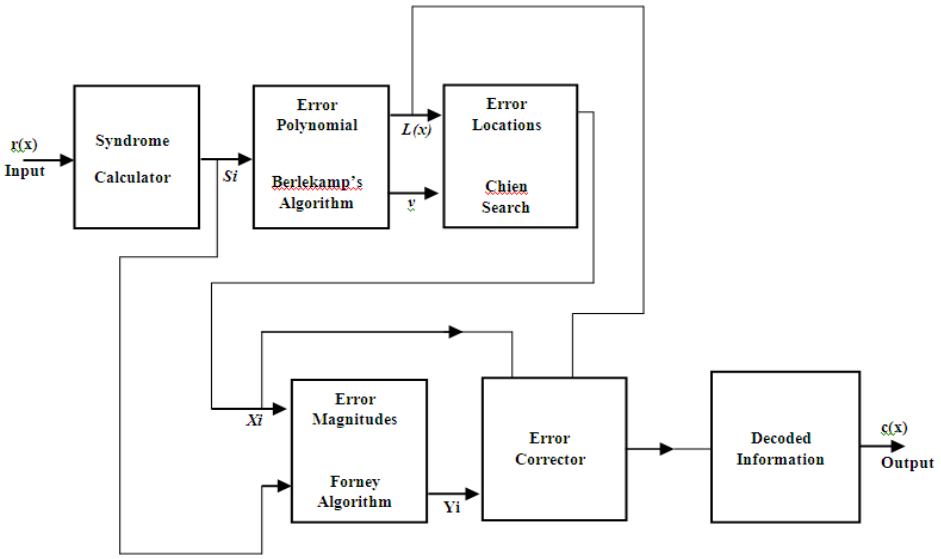

Figure 1: Block diagram of the Reed Solomon Decoder

The symbols represented in Figure 1 are:

r(x) = Received code word

Si = Syndromes

L(x) = Error Locator Polynomial

Xi = Error Locations

Yi = Error Magnitudes

C(x) = Recovered Code word

v = Number or errors

The Reed Solomon decoder can be implemented by using FPGA for improving the performance as well as quality of signal.

RESULTS

Simulation Results

The simulation result of (31, 21) Reed Solomon error corrector is shown in Figure 2. The figure illustrates the data word (111111100000000000000) which entered to the FPGA encoder to be encoded. After adding the parity bits to the data, the code word are performed to be (11101000000000000000000000000000) as shown in the Figure 2. We are getting the corrected code word at dout as (111111100000000000000).

The second simulation result is shown in the Figure 3. The figure illustrates the data word (00000000000011111111) which entered to the FPGA encoder to be encoded. After adding the parity bits to the data, the code word are performed to be (00110011000000000000000000000000) as shown in the Figure 3. We are getting the corrected code word at dout as (00000000000011111111). Thus this RS code detects as well as corrects the errors in the code word.

Thus RS codes have been shown to be excellent error correcting codes among codes of short lengths. The simulation test result shows that the architecture can operate at a clock frequency of 200MHz to achieve a throughput of 4.2 Gb/s. Thus RS codes are simple to encode and relatively simple to decode. Due to these qualities, there is much interest in the exact capabilities of these codes.

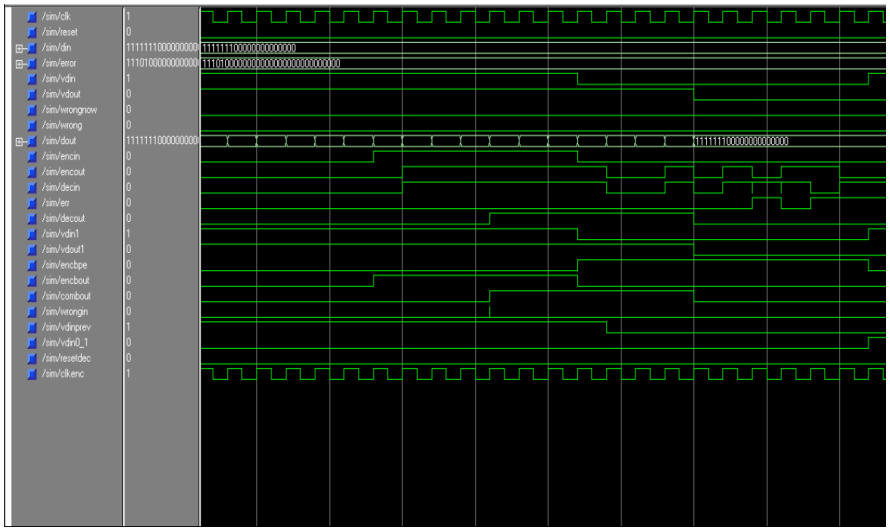

Figure 2: Simulation Result-1 of a (31, 21) Reed-Solomon Corrector

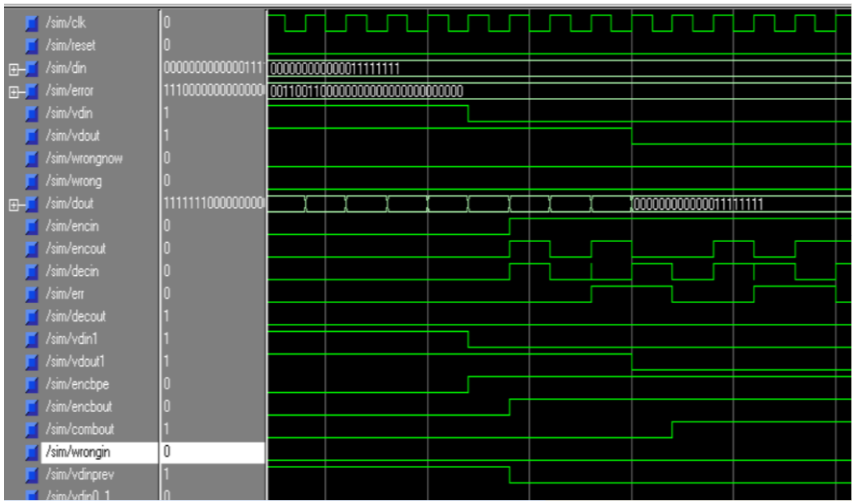

Figure 3: Simulation Result-2 of a (31, 21) Reed-Solomon Corrector

FPGA Spartan kit is shown in the Figure 4. The FPGA is connected with a computer in order to download the software of each system into an FPGA chip as shown in Figure 5.

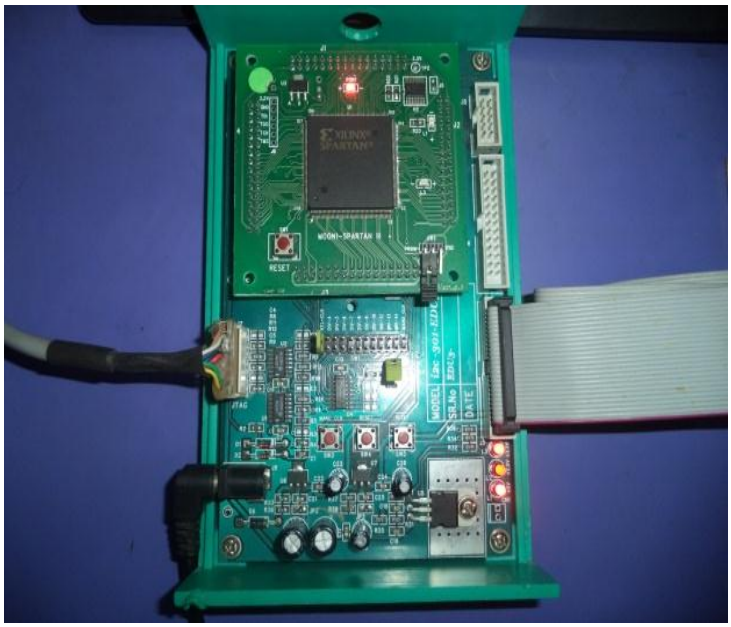

Figure 4: Spartan Kit

Figure 5: Experimental Setup

Table 1: Device Utilization Summary using Model-Sim

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	126	9,312	1%
Number of 4 input LUTs	145	9,312	1%
Number of occupied Slices	105	4,656	2%
Number of Slices containing only related logic	105	105	100%
Number of Slices containing unrelated logic	0	105	0%
Total Number of 4 input LUTs	145	9,312	1%
Number used as logic	139		
Number used as Shift registers	6		
Number of bonded IOBs	79	158	50%
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	2.46		

From Table 1 we can see that the Number of slice Flip Flops utilization is only 1% i.e. only 126 Number of slice Flip Flops are used. Also we can observe that number of input LUTs and number of occupied slices utilization is only 1% and 2% respectively. This shows that this design requires less area. Thus this new architecture of Reed Solomon Error corrector reduces the occupied area of FPGA and increases the overall efficiency of the system.

RTL Schematic of RS Corrector is shown in Figure 6.

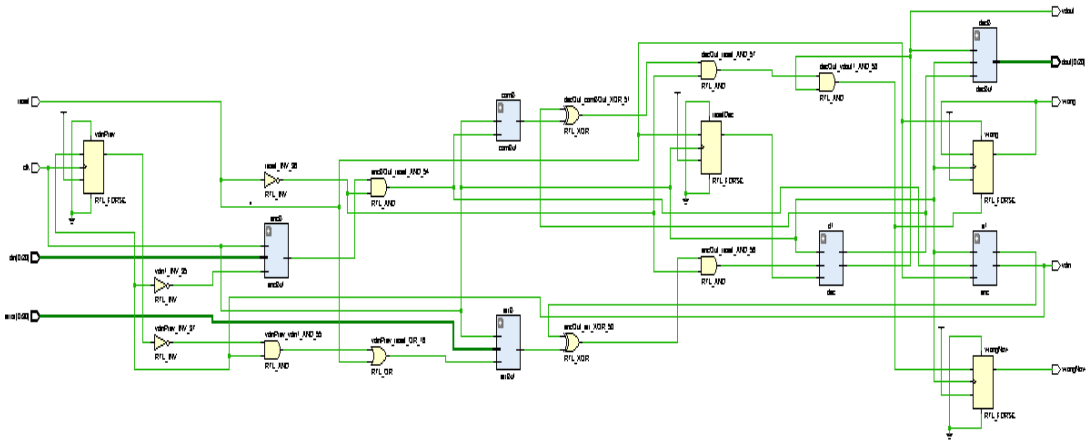

Figure 6: RTL Schematic of RS Corrector

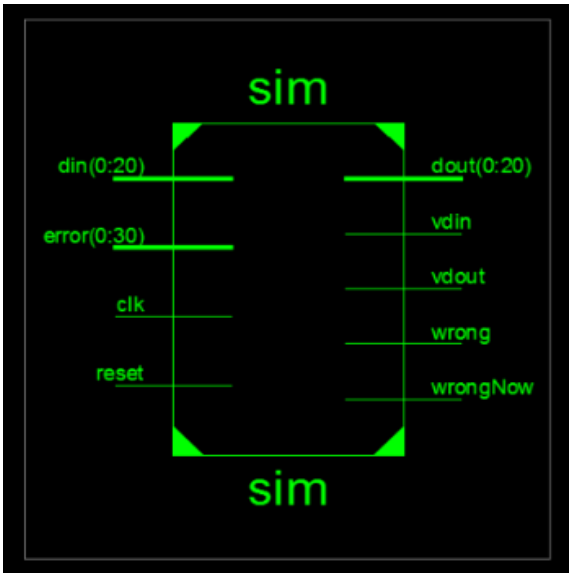

Figure 7: Technology Schematic of RS Corrector

Figure 7 shows the technology schematic of RS Corrector. This schematic is generated after the optimization and technology targeting phase of the synthesis process. It shows a representation of the design in terms of logic elements optimized to the target Xilinx device or "technology"; for example, in terms of of LUTs, carry logic, I/O buffers, and other technology-specific components. Viewing this schematic allows you to see a technology-level representation of your HDL optimized for a specific Xilinx architecture, which might help you discover design issues early in the design process.

CONCLUSION

This research paper describes new design strategy targeting area optimization in reconfigurable device (FPGA). Area optimization is a major issue as smaller components allow for better system adaption, which is an expected feature of reconfigurable system for space applications. The approach is applied in the design stage of a component for the communications module of an on-board computer system. The chosen component is a Reed-Solomon encoder, which has been implemented using a Hardware Description Language (VHDL) targeting an FPGA platform. The research work investigates traditional alternatives for the encoder implementation, introduces the new architecture of the error corrector behind the proposed approach, describes the design process and discusses the area figures reached by the new design. Since the new design described in this research work, occupies very small area on FPGA, it presents an ultra-low cost VLSI architecture of RS corrector with reduced hardware complexity. Even though this new design requires very less area on FPGA it features a multiple-error correction capability. Thus it enables programmable solution to a large variety of applications employing error control coding techniques in the communication and consumer electronics field.

ACKNOWLEDGMENTS

This work is carried out in the Digital/VLSI Lab of the Ibra College of Technology, Oman and in the VLSI Lab of the Shivaji University, India. Authors would like to thank all the authorities and staff of the Ibra College of Technology, Oman and Department of Electronics, Shivaji University, India, for providing maximum facilities of the Laboratories.

REFERENCES

[1] Wicker, S. B., & Bhargava, V. K. (1999). Reed-Solomon codes and their applications. John Wiley & Sons.

[2] Koetter, R., & Vardy, A. (2003). Algebraic soft-decision decoding of Reed-Solomon codes. Information Theory, IEEE Transactions on, 49(11), 2809-2825.

[3] Guruswami, V., & Sudan, M. (1998, November). Improved decoding of Reed-Solomon and algebraic-geometric codes. In Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on (pp. 28-37). IEEE.

[4] McEliece, R. J. The Guruswami-Sudan Decoding Algorithm for Reed-Solomon Codes, 2003. IPN Progress Report, 42-153.

[5] Könemann, B. (1991, April). LFSR-coded test patterns for scan designs. InProc. of European Test Conference (pp. 237-242).

[6] Krawczyk, H. (1994, January). LFSR-based hashing and authentication. InAdvances in Cryptology—CRYPTO'94 (pp. 129-139). Springer Berlin Heidelberg.

[7] Krishna, C. V., Jas, A., & Touba, N. (2001). Test vector encoding using partial LFSR reseeding. In Test Conference, 2001. Proceedings. International (pp. 885-893). IEEE.

[8] Dufaza, C., & Cambon, G. (1991, April). LFSR based deterministic and pseudo-random test pattern generator structures. In Proc. European Test Conference(Vol. 199, No. 1).

[9] Goresky, M., & Klapper, A. M. (2002). Fibonacci and Galois representations of feedback-with-carry shift registers. Information Theory, IEEE Transactions on,48(11), 2826-2836.

[10] Alaus, L., Noguet, D., & Palicot, J. (2008, May). A reconfigurable linear feedback shift register operator for software defined radio terminal. In Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on (pp. 319-323). IEEE.

[11] Joux, A., & Delaunay, P. (2006). Galois LFSR, embedded devices and side channel weaknesses. In Progress in Cryptology-INDOCRYPT 2006 (pp. 436-451). Springer Berlin Heidelberg.

[12] Dubrova, E. (2009). A transformation from the Fibonacci to the Galois NLFSRs.Information Theory, IEEE Transactions on, 55(11), 5263-5271.

[13] Mukherjee, N., Rajski, J., Mrugalski, G., Pogiel, A., & Tyszer, J. (2011). Ring generator: an ultimate linear feedback shift register. Computer, 44(6), 64-71.

[14] Joo, Y., Niu, D., Dong, X., Sun, G., Chang, N., & Xie, Y. (2010, March). Energy-and endurance-aware design of phase change memory caches. In Proceedings of the Conference on Design, Automation and Test in Europe (pp. 136-141). European Design and Automation Association.