

Hardware Implementations of RS Decoding Algorithm for Multi-Gb/s Communication Systems

Dr. Sanjeev Kumar¹, Dr. Seema Verma², Sandeep Kulhari³

¹ Principal, Sine International Institute of Technology, Jaipur, Rajasthan, India

² Professor Dept. of Electronics, Banasthali Vidyapith, Banasthali, Rajasthan, India

³ Ph.D Scholar, Shri Jagdishprasad Jhabarmal Tibrewala University (JJT), Jhunjhunu, Rajasthan, India

Abstract: - In this paper, we have designed the VLSI hardware for a novel RS decoding algorithm suitable for Multi-Gb/s Communication Systems. Through this paper we show that the performance benefit of the algorithm is truly witnessed when implemented in hardware thus avoiding the extra processing time of Fetch-Decode-Execute cycle of traditional microprocessor based computing systems. The new algorithm with less time complexity combined with its application specific hardware implementation makes it suitable for high speed real-time systems with hard timing constraints. The design is implemented as a digital hardware using VHDL.

Keywords: RS Decoder, VLSI, MATLAB, Scientific Computing & Simulation, Data Communication, FPGA Design

I. INTRODUCTION

The Reed-Solomon codes are found in various applications ranging from digital communications to high speed recording systems [1]. The basic reason for such widespread use of RS codes is its ability to correct burst errors thereby yielding highly error-free systems. Since every particular application has its own performance and time requirements, a large number of techniques exist for high speed decoding of RS codes, each of them being suitable for certain types of systems [7].

Nevertheless, as the technology progresses and concepts like high definition (HD) broadcast and Internet of Things becomes a reality, we would require highly stable error free systems that can operate at multi-GHz speeds [2]. For such systems we would require very high speed RS decoding capability which is basically the motivation behind growing research in this field.

The problem of high speed RS decoding can be approached via two broad ways. One is the exploration of new algorithms with low running time complexity. The other is development of new hardware architectures with high throughput [8]. A lot of research has been done in both these areas and consequently a lot of solution exists too [9].

The approach presented herein lies in the interface of the two aforementioned areas. It is the intent of this work to develop a new algorithm which is suitable for implementation as VLSI hardware using parallel computing. The chain of thought that became the motivation behind this research was that a new

algorithm with low running time complexity may be efficient for computing but it may not be suitable for direct hardware implementation due to various system level requirements. Although they can be implemented in software, a software based solution is not feasible for high speed real-time systems as it is slower than hardware. If on the other hand research is done on only efficient hardware architectures for Reed – Solomon decoding, it may not be suitable for every algorithm. And optimization of hardware on a case-by-case basis for implementation of every algorithm although will work in principle but is not desired given stringent time to market constraints in today's time. Therefore, a new general framework of parallel algorithms is being proposed that can be easily implemented in hardware using parallel architectures.

II. REVIEW OF REED SOLOMON CODES

The Reed – Solomon codes are cyclic codes with symbols made up of m -bit sequences, where m is any positive integer having a value greater than 2 [3]. RS (n,k) codes on m -bit symbols exist for all n and k for which the following condition is satisfied:

$$0 < k < n < 2^m + 2 \quad \dots (1)$$

Where k is the number of data symbols being encoded, and n is the total number of code symbols in the encoded block. For most conventional RS (n,k) codes,

$$(n, k) = (2^m - 1, 2^m - 1 - 2t) \quad \dots (2)$$

Where t is the symbol-error correcting capability of the code and $n - k = 2t$ is the number of parity symbols. RS codes achieve the largest possible code minimum distance for any linear code with the same encoder input and output block lengths. The code minimum distance is given by:

$$d = n - k + 1 \quad \dots (3)$$

The RS code generator polynomial can be written as:

$$g(x) = \prod_{i=b}^{b+d-2} (x - \alpha^i) \quad \dots (4)$$

With the code being defined over $GF(q)$, q being a prime power and the code length is $n = q - 1$, b is any natural number and α is a primitive element of $GF(q)$. The received vector is represented as a polynomial:

$$R(x) = \sum_{i=0}^{n-1} r_i x^i = C(x) + E(x) \quad \dots (5)$$

Where $C(x)$ is the codeword and $E(x)$ is the error vector. The i -th error in the error vector $E(x)$ has a locator given as :

$$Z_i \in \{\alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{n-1}\} \quad \dots (6)$$

And the error value

$$Y_i \in GF(q) \setminus 0 \quad \dots (7)$$

The error locator polynomial is given as:

$$W(x) = \prod_{i=1}^t (x - Z_i) \quad \dots (8)$$

Where $t \leq (d - 1) / 2$. The message polynomial of the RS code can be shown to be:

$$M(x) = \sum_{i=0}^{k-1} m_i x^i \quad \dots (9)$$

The component c_i of the codeword $C(x)$ is computed as :

$$c_i = M(\alpha^i), i \in [0, n-1] \quad \dots (10)$$

For our work we have chosen Gao's algorithm as the starting point and we describe it here for the case of $b = 1$ [11].

III. REED SOLOMON DECODING

Step 1: Construct an interpolating polynomial $T(x)$ such that

$$T(\alpha^i) = r_i, i \in [0, n-1] \quad \dots (11)$$

Where degree of $T(x) < n$.

Step 2: Compute the partial GCD by solving the congruence such that

$$W(x)T(x) \equiv P(x)[\text{mod}(x^n - 1)] \quad \dots (12)$$

Where degree of $P(x) < (n+k)/2$

The above step can be done by applying the extended Euclidean algorithm to $x - 1$ and $T(x)$ and we can obtain unique pair of polynomials $P(x)$ and $W(x)$.

Step 3: Compute the message polynomial as

$$M(x) = \frac{P(x)}{W(x)} \quad \dots (13)$$

The choice of Gao's algorithm as the starting point was dictated by the fact that it has very good asymptotic complexity of $O(n(\log n))$.

IV. PARALLEL APPROACH TO RS DECODING

The above algorithm is very efficient but once implemented in hardware will suffer from various implementation issues. In particular, the first step of Gao's algorithm is nothing but taking the discrete Fourier transform over $GF(q)$ [4]. The number of multipliers in the circuit is code length dependent and even if we develop efficient DFT hardware suitable for step 1, step 2 and 3 will still take time to compute. The salient features of our new algorithm are as follows:

Step 1: Construct an interpolating polynomial $T(x)$ such that

$$T(\alpha^i) = r_i, i \in [0, n-1] \quad \dots (14)$$

Where degree of $T(x) < n$.

Step 2: Define a set of linear functionals over $T(x)$ that maps elements of $T(x)$ to its algebraic dual $T^*(x)$. That is :

Reed – Solomon codes can be decoded in both time and frequency domain [5]. Consequently, a lot of algorithms already exist for efficient decoding in both these domains. However,

$$f : T(x) \rightarrow T^*(x) \quad \dots (15)$$

Step 3: Check for roots of $T(x)$ in the extended field of $T(x)$ and by applying the extended Euclidean Algorithm to $x - 1$ and $T(x)$ compute the partial GCD by solving the congruence such that

$$W(x)T(x) \equiv P(x)[\text{mod}(x^n - 1)] \quad \dots (16)$$

Where degree of $P(x) < (n+k)/2$

Step 3 can be implemented in parallel and using suitable choice of codeword length, can converge to solution faster.

Step 4: Compute the message polynomial as

$$M(x) = \frac{P(x)}{W(x)} \quad \dots (17)$$

This division can also be carried out by a new division algorithm that is derived from Finite Field Arithmetic. Basically this division can be computed in a much shorter time using modular arithmetic as these polynomials are defined over finite fields whose basis can be recovered from Step 2. Hence, we see that by suitable derivation of a division algorithm modulo $x - 1$ and by computing the GCD using a parallel algorithm can sufficiently reduce the computation time of reed-Solomon decoding. Furthermore, our algorithm is rather a framework that can be applied to general classes of frequency domain decoding of Reed-Solomon Codes.

V. HARDWARE IMPLEMENTATION

The proposed decoding framework has been implemented on Xilinx Virtex 6 FPGA using Xilinx ISE Design Suite 14.1. The CLB architecture of Virtex 6 FPGA can be seen in figure 1.

As shown, each CLB has two slices which do not have direct connections to each other and each slice is organized as column. Each slice in a column has independent carry chain. The extensive logic resources provided by this FPGA are enough to contain most complex designs with very large gate count. This is one of the reasons for choosing this particular FPGA.

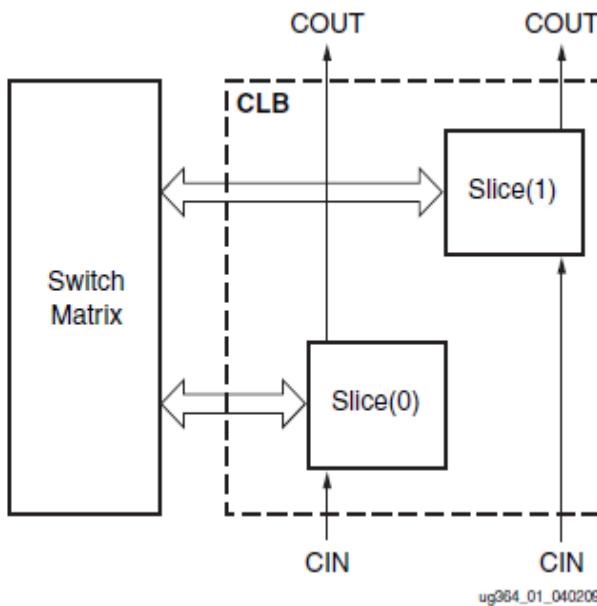


Figure 1: Arrangement of slices within the CLB (Courtesy Xilinx Inc.)

Each slice contains four look-up tables, eight storage elements, wide-function multiplexers and general carry logic. These slices are used to provide logic, arithmetic and ROM functions. In addition to this, some slices support two additional functions: ***storing data using distributed RAM and shifting data with 32-bit registers***.

The hardware design created in this research work is simulated for its accuracy and its performance is duly verified. A testbench is created to test the design with certain assumed inputs corresponding to codewords in Galois Field represented in binary format and consequently the outputs of the design are checked. Both behavioural and post-synthesis simulation of the design has been carried out. Correct behavioural simulation of the design also led to its ASIC implementation which has been described in the next section.

The simulation of the hardware as seen in figure 2 shows that the hardware is efficient in decoding the RS codes. The simulation shows that each of the independent codewords fed to the design are parallelly computed by the hardware and the corresponding decoded words are displayed. It can be seen, as the errors in codewords are increased the hardware also takes more time to decode them. However, the increase in time does not follow a linear trend and shows more computational efficiency with each successive clock cycle. The decoded words can be seen as values displayed in the timing diagram within the timing waveform. The state changes that have been shown preceding the value shows the computation time of the hardware

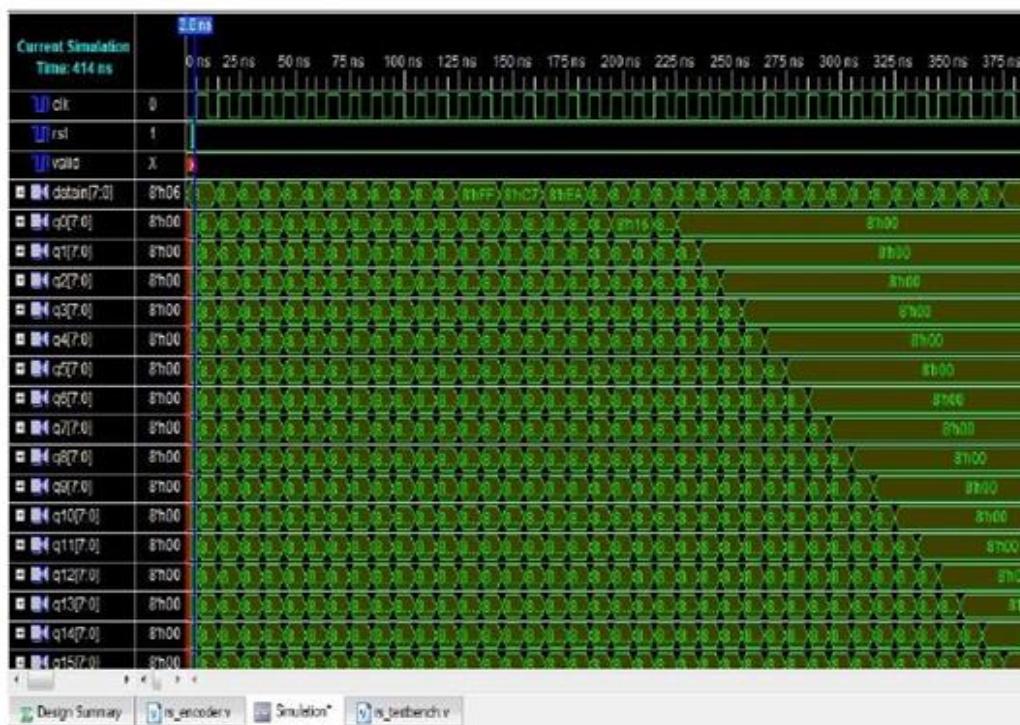


Figure 2: Simulation of the Hardware for the Proposed Decoding Framework

VI. CONCLUSION

In this paper, we showed the hardware implementation of a novel algorithm for RS decoding and proved its performance benefit through hardware and system level simulation. The curves and requisite data obtained were quite close to the theoretical limit thus validating the design. Our implementation show reasonable performance and has proven that the algorithm is capable of decoding arbitrary length RS codes within the constraints of real-time communication systems.

REFERENCES

- [1]. S. Reed and G. Solomon, “Polynomial Codes Over Certain Finite Fields”, Journal of the Society for Industrial and Applied Mathematics, Vol. 8, pp. 300 – 304 , 1960.
- [2]. H. Liu, H. Ma, M. E. Zarky and S. Gupta, “Error Control Schemes for Networks: An Overview”, Mobile Networks and Applications, Volume 2, pp. 167-182, 1997
- [3]. R. E. Blahut, “A Universal Reed Solomon Decoder”, IBM Journal of Research and Development, Vol. 28, No. 2, 1984
- [4]. G. Fettweis and M. Hassner, “A combined Reed Solomon encoder and syndrome generator with small hardware complexity”, Proceedings of IEEE International Symposium on Circuits and Systems, 1992.
- [5]. Ahmed, R. Koetter and N. R. Shanbhag, “VLSI Architectures for soft decision decoding of Reed Solomon Codes”, IEEE Conf. Communications, vol. 5, pp. 2584-2590, 2004
- [6]. H.-C. Chang and C. B. Shung, “New serial architectures for the Berlekamp–Massey algorithm,” IEEE Trans. Commun., vol. 47, pp. 481–483, Apr. 1999.
- [7]. Ahmed, N. R. Shanbhag and R. Koetter, “Systolic interpolation architectures for soft-decoding Reed-Solomon codes”, IEEE Workshop on Signal Processing Systems, 2003.
- [8]. W. J. Gross, F. R. Kschischang, R. Koetter and P. G. Gulak, “Architecture and Implementation of Interpolation Processor for Soft-decision Reed Solomon Decoding” IEEE Transactions on VLSI Systems, vol. 15, no. 3, pp. 309-318 2007.
- [9]. S. Lee and H. Lee, “A High-Speed Pipelined DegreeComputationless Modified Euclidean Algorithm Architecture for Reed-Solomon Decoders,” IEICE Trans. on Fund. of Electronics, Communications, and Computer Sciences, Vol.E91-A, No.3, pp.830-835, March, 2008.