Research Paper Engineering

Performance Comparison of RS Decoder on Family of MAX CPLD using Verilog HDL

*Hitesh G. Kamani

* Department of Electronics & Communication, Gujarat Technological University, Ahmadabad, India.

ABSTRACT

In this paper design of Reed Solomon (255,239) Decoder and perform this decoder on family of MAX CPLD and compare the performance based on area occupied by the design and the speed at which the design can run and power dissipation. I applied forward error correction system to improve the overall performance of the system .The implementation is Written in Verilog HDL based on Barlekamp Massy, Forney and Chien Search Algorithm.

Keywords : Verilog HDL, FEC (Forward error Correction), CPLD (complex Programmable Logic Device)

1. INTRODUCTION

Forward error correction (FEC) in *transceiver* (transmitter/receiver pair) is used to deliver information from a source (transmitter) to a destination (receiver) through a noisy communication channel with a minimum of errors. FEC allows a receiver in the system to perform Error Detection and Correction (EDAC) without requesting a retransmission of the corrupted data. FEC offers a number of benefits:

FEC enables a system to achieve high data reliability.

FEC results in greater effective throughput of user data, because valuable bandwidth is not being used to retransmit corrupted data.

FEC yields performance gains and low error rates for systems in which other options, such as increasing the transmitted power or installing noise-limiting components, are too expensive.

System costs can be reduced by eliminating an expensive or sensitive component and compensating for the lost performance by a suitable FEC scheme.

2. OVERVIEW OF RS CODE

In coding theory, Reed–Solomon (RS) codes are non-binary [10] cyclic error-correcting codes invented by Irving S. Reed and Gustave Solomon. They described a systematic way of building codes that could detect and correct multiple random symbol errors. By adding t check symbols to the data, an RS code can detect any combination of up to t erroneous symbols, or correct up to $\lfloor t/2 \rfloor$ symbols. As an erasure code, it can correct up to t known erasures, or it can detect and correct combinations of errors and erasures. Furthermore, RS codes are suitable as multiple-burst bit-error correcting codes, since a sequence of b + 1 consecutive bit errors can affect at most two symbols of size b. The choice of t is up to the designer of the code, and may be selected within wide limits.

In Reed–Solomon coding, source symbols are viewed as coefficients of a polynomial p(x) over a finite field. The original idea was to create n code symbols from k sourcesymbols by oversampling $p(x)$ at $n > k$ distinct points, transmit the sampled points, and use interpolation techniques at the receiver to recover the original message. RS codes are block based error correcting codes and applied in wide range of system including storage device such as, compact disk DVD ,wireless and mobile communication ,digital television and digital video broadcasting, high speed modem.[3] RS codes are defined as The general form of the polynomial is given as [1]

G (x) = (x-α1) (x-αi-1)……. (x- α 1+2t)

Codeword is constructed using [2]

C(x) = G(x). I (x)

I(x) = information block, G(x) = generator polynomial(x) = code polynomial

3. HOW FEC WORKS

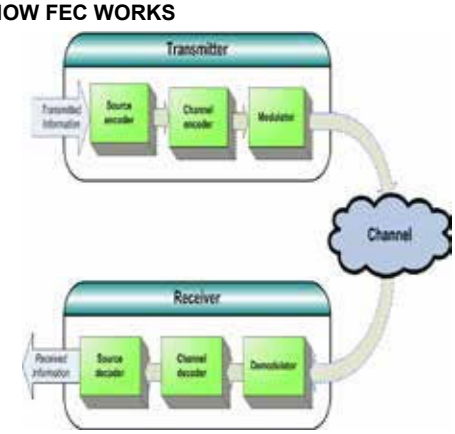

Fig 1: How FEC work [15]

4. REED SOLOMON DECODER

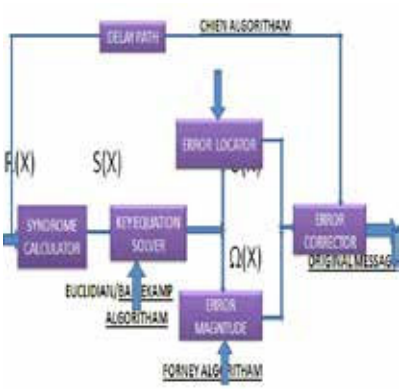

Fig 2: Reed Solomon Decoder Block Diagram

Fig: 2 show the general block diagram of the reed Solomon decoder.

4.1 SYNDROME CALCULATOR

The syndrome accumulate is the first step in the RS decoding process. This is done to detect if there are any errors in the received code word.

After encoding a given message, the code word polynomial

$c(X) = c_0 + c_1 X + ... + c_{n-1} X$
$r(X) = r_0 + r_1 X + ... + r_{n-1} X$
$r(X) = c(X) + e(X)$
$e(X) = r(X) - c(X) = e_0 + e_1 X + ... + e_{n-1} X$

Where $e_i = r_i - c_i$ is a symbol from GF(2^m).

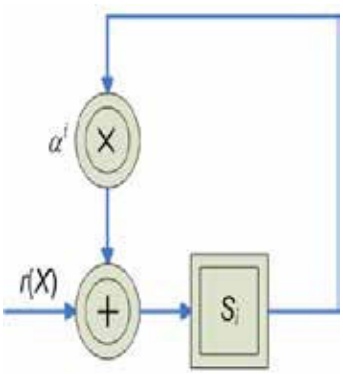

Fig 3: Syndrome Calculator [15]

4.2 KEY EQUATION SOLVER

Output of the syndrome calculator fed to the next block key eqation solver. It process the S(x) and to generate error locator polynomial σ(X) and error magnitude polynomial Ω(X).

That is, it solves the following equation that is referred to as the key equation.

σ(X) [1 + S(x)] = Ω(X) mod x2t+1

The algorithm used in RS decoding is based on Barlekamp Massy algorithm for finding the greatest common devisor (GCD) of two polynomials. Barlekamp massy algorithm is an iterative polynomial division algorithm.

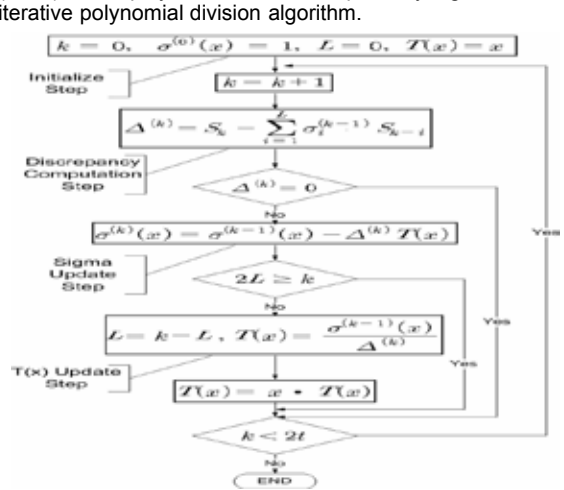

Fig 4: Flowchart of Berlekamp Massey algorithm [8]

4.3 ERROR LOCATOR

Once the error locator polynomial σ(x) has been computed, it needs to be evaluated to find its roots. The chien search (CS) algorithm is used to find these roots (fig) The CS is a brute force algorithm that evaluates the polynomial for all possible input values, and then checks to see which outputs are equal to zero. If an error occurs in position i, then the following equation equals zero. Where i = 0... (n - 1).

The CS evaluates the above equation for all the values of i and j and counts the number of times that the equation is equal to zero. The locations of the zeros are the error locations, and the number of zeros is the number of symbols in error. There are (t + 1) stages of the CS that are implemented in hardware. Each of these stages (where a stage consists of a multiplier, mux and register) represents a different value for j in the above CS equation. The search is run for n clock cycles (each clock cycle represents a different value of i in the above equation) and the output of the adder is examined to see if it is equal to zero. If it is equal to zero, the zero detect block will output a 1, otherwise, it will output a zero.

The output of the chien search block is thus a string of n bits that have values of either 0 or 1. Each 1 represents the location of a symbol in error. For the first clock cycle, the mux will route the error locator polynomial coefficient into the register. For the remaining (n -1) clock cycles, the output of the multiplier will be routed via the mux into the register. The exponents of the multipliers have negative values. However, these values can be precomputed using the modulo operator. The exponent of i is equal to (- i modulo n) = (-I modulo 255). For example, α^{-1} equals α^{254},α^{-2} equals α^{253} and so on.

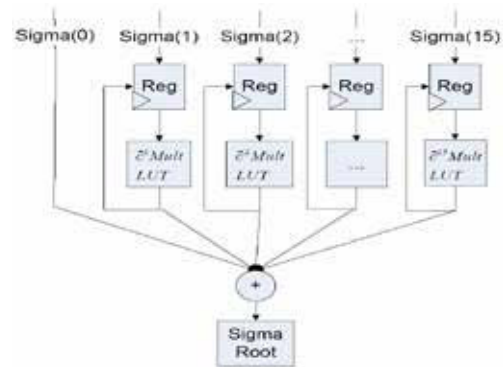

Fig 5: Chine Search Block[8]

4.4 ERROR MAGNITUDE

The Forney algorithm is used to compute the error values Y_i. To compute these values, Forney algorithm needs the error locator polynomial Λ(x) and error magnitude polynomial Ω(x). The equation for the error values is for $x=\alpha^{-1}$ where α^{-1} is a root of Λ(x). The computation of the formal derivative Λ'(x) is actually quite simple.

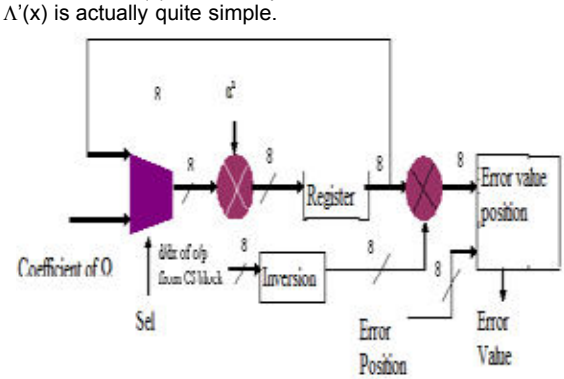

Fig 6: Error Magnitude Block [8]

4.5 ERROR CORRECTION

The output of the chien/Forney block is the error vector. This vector is the same size as the codeword. The vector contains non zero values in locations that correspond to errors. Because the error vector is generated in the reverse order of the received codeword, a FIFO must be applied to either the received codeword or the error vector to match the order of the bytes in both vectors. The output of the adder is the decoder's estimate of the original codeword.

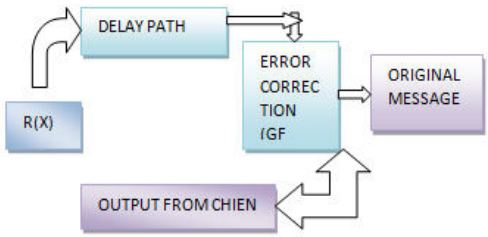

Fig 7: Error Correction Block

4.6 FIFO (FIRST IN FIRST OUT)

FIFO is an acronym for **First In, First Out**. This expression describes the principle of a queue or first-come, first-served (FCFS) behaviour, what comes in first is handled first, what comes in next waits until the first is finished, etc. Thus it is analogous to the behaviour of persons queuing, where the persons leave the queue in the order they arrive. In hardware form, a FIFO primarily consists of a set of read and write pointers, storage and control logic. Storage may be SRAM, flip-flops, latches or any other suitable form of storage. For FIFOs of non-trivial size a dual-port SRAM is usually used where one port is used for writing and the other is used for reading. A synchronous FIFO is a FIFO where the same clock is used for both reading and writing. An asynchronous FIFO uses different clocks for reading and writing.

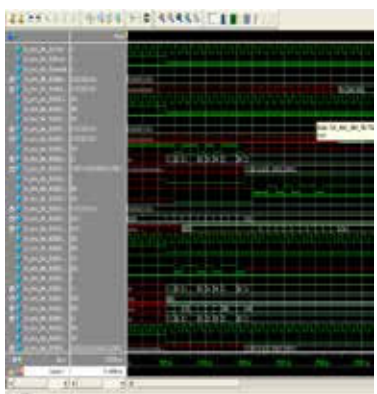

Fig 8: Wave form of RS Decoder

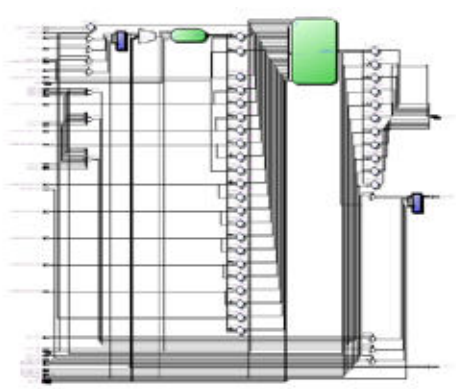

Fig 9: RTL View of Decoder

5. RESULT

Compile: Quartus II 10.1

Analysis & Synthesis Setting : Smart Compile
Compile Process Setting : Balanced

DEVICE UTILIZATION SUMMARY						
	MAX II	MAX V	MAX3000A	MAX 7000AE	MAX 7000B	MAX 70005
TOTAL LE	63/240 26%	63/80 79%	35/128 27%	35/128 27%	35/64 55%	35/64 55%
TOTAL PIN	25/80 31%	25/52 48%	29/80 36%	29/84 35%	29/36 81%	29/36 81%
TIMING SUMMARY						
SPEED GRADE	-5	-5	-5	-5	-5	-5
MINIMUM PERIOD	6.95 nS	30.4 nS	12.8 nS	12.8 nS	11.9 nS	16 nS
Fmax	143.93 MHz	32.9 MHz	78.13 MHz	78.13 MHz	84.03 MHz	62.50 MHz
POWER SUMMARY						
TOTAL THERMAL POWER DISSIPATION	46.22 mW	0.12 mW	107.83 mW	107.83 mW	106.5 Mw	NOT AVAILABLE

Table 1: the Comparison of Hardware and Timing and power Performance of RS Decoder

6. CONCLUSION

In this paper, FEC Decoder has been presented to remove error in wireless communication.FEC decoder design is done by RS (Reed Solomon) codes. The analysis and simulation is done using QUARTUS II 10.1. We implemented a RS coding system based on (255, 239) RS code via a Verilog hardware description language (VerilogHDL) and synthesized for the CPLD chip MAX family. The result shows that the decoder could operate at a max frequency of 143.93 MHz for MAX II.By comparing the RS Decoder on different MAX CPLD, We conclude that MAX II CPLD higher maximum frequency and occupies less area of CPLD compared to other MAX family chip, But total thermal power dissipation in MAX V low compared to other MAX family chip.

7. FUTURE SCOPE

Coding field of communication is highly growing area in present era of research. Still, it needs a lot of improvement in the field of error correction and coding techniques. By improving concatenation technique i.e. by concatenation of LDPC (Low Density Parity Check) as inner code and RS (Reed Solomon) as outer coding, we can enhance the performance of FEC system. The goal of our present work is to reduce the occupied area of CPLD and increase the efficiency. The development of highly effective decoding algorithms for the implementation is another area where the significant amount of research work can be done.

REFERENCES

[1] Lin, S. and D. J. Costello. 1983. Error Control Coding: Fundamental and Application. Pearson Prentice Hall. | [2] I. S. Reed and G. Solomon, “Polynomial Codes over Certain Finite Fields”, Journal of the Society of Industrial and Applied Mathematics. Pp 8: 300–304, 1960 | [3] Wicker, S. B. and E. Bhargava. 1994. Reed-Solomon Codes and their Applications. New York: IEEE Press. | [4] DVB, “Framing Structure, channel coding and modulation for digital terrestrial television”, ETSI EN 300 744, Vol 4.1, January 2001. | [5] B. Sklar, Digital Communication, 2nd edition Upper Saddle River, NJ, Prentice Hall, pp 436~460, 2001. | [6] M.A. Khan, S. Afzal, R. Manzoor, Hardware implementation of Shortened (48,38) Reed Solomon Forward Error Correcting Code, Proceedings IEEE INMIC 2003. | [7] M. H. Lee, S. B. Choi, and J. S. Chang, HIGH SPEED REED-SOLOMON DECODER, IEEE Transactions on Consumer Electronics, Vol. 41, No. 4, NOVEMBER 1995. | [8] An FPGA Implementation of IEEE802.16aSHORTENED&PUNCTURED Reed Solomon Code. | [9] FPGA Implementation of RS Codec for Digital Video Broadcasting. | [10]Cipra, Barry A. (1993), "The Ubiquitous Reed–Solomon Codes", SIAM News 26 (1) | [11]Berlekamp, Elwyn R. (1967), Nonbinary BCH decoding, International Symposium on Information Theory, San Remo, Italy | [12]Berlekamp, Elwyn R. (1984) [1968], Algebraic Coding Theory, Laguna Hills, CA: Aegean Park Press, ISBN 0-89412-063-8 | [13]Forney, Jr., G. (October 1965), "On Decoding BCH Codes", IEEE Transactions on Information Theory 11 (4): 549–557, doi:10.1109/TIT.1965.1053825 | [14]Gill, John (unknown), EE387 Notes #7, Handout #28, Stanford University, retrieved April 21, 2010 | [15]Analyzing and implementing a Reed-Solomon Decoder for Forward Error Correction in ADSL