Design and comparison of Reed Solomon Decoder on Two Different FPGA using Verilog HDL

Hitesh G.Kamani

Department of Electronics & Communication
Gujarat Technological University, Ahmadabad, India.
Mob: +919974158939
Email: hitesh.patel00113@gmail.com

ABSTRACT: In this paper design of Reed Solomon (255,239) Decoder and perform this decoder on two FPGA of quartus Stretix III (EP35L50F484C4) and Cyclone IV GX (EP4CGX15BF14C6) and compare the performance based on area occupied by the design and the speed at which the design can run and total thermal 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).

1. INTRODUCTION

There are two techniques for error correction one is retransmission and another is forward error correction system. Error Detection in a block of data then request a retransmission, known as automatic repeat request (ARQ) for sensitive data. It is a two way transmission It is appropriate for low delay channel and channel with a return path. Forward error correction coding is designed so that error can be corrected at the receiver. This is appropriate for delay sensitive and it is a one way transmission.

2. OVERVIEW OF RS CODE

Reed Solomon codes are popular FEC scheme discovered and described in a paper by Reed Solomon in 1960[2].RS codes are very efficient in correcting random symbol error and random burst error. 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 RS(N,K).RS codes operate on special area of mathematics known as Galois field. Galois field also known as finite field. A finite field has the property that arithmetic operation on field element always has a result in that field.

The general form of the polynomial is given as [1]

$G(x) = (x-\alpha 1)(x-\alpha i-1)\ldots\ldots\ldots (x-\alpha 1+2t)$

Code word 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

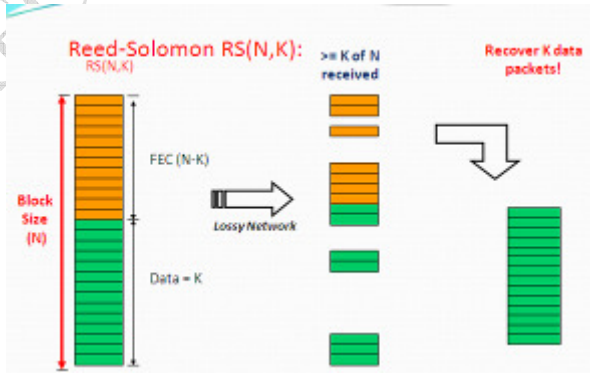

Fig 1: How FEC work

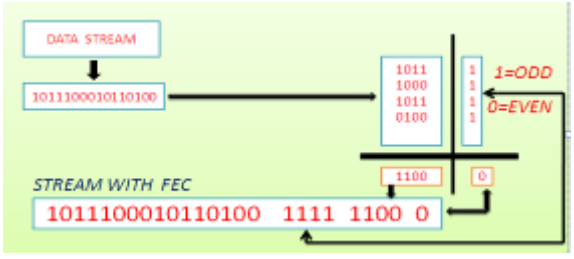

Fig 2: Data Stream of FEC

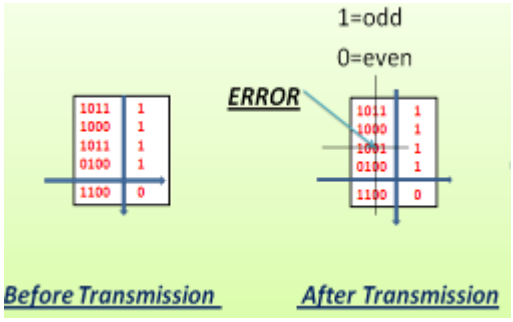

Fig 3: How Error Correction with FEC

4. REED SOLOMON DECDER

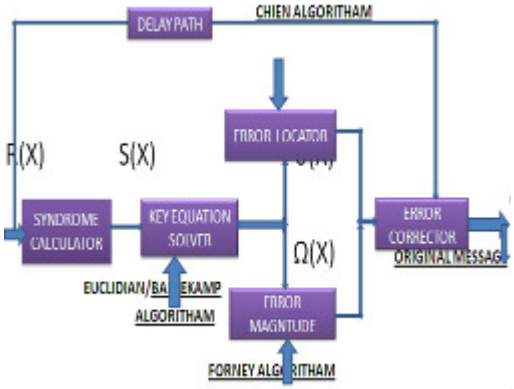

Fig 4: Reed Solomon Decoder Block Diagram

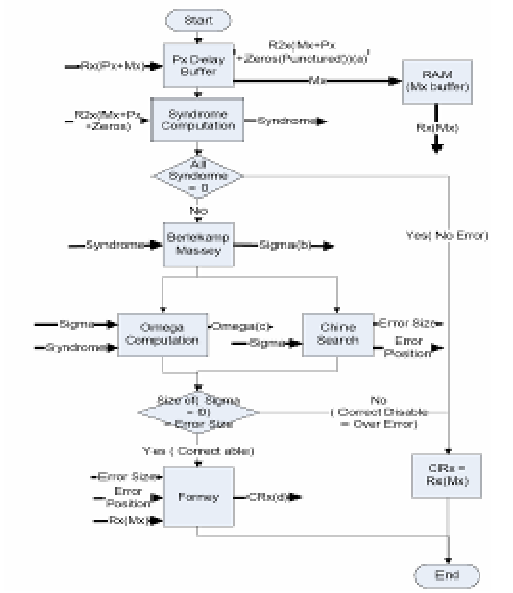

Fig 5: Reed Solomon decoder flow chart [8]

Fig: 4 show the general block diagram of the reed Solomon decoder.

4.1 SYNDROME CALCULATOR

The first block is the syndrome calculator(X) is the received polynomial.Recieved polynomial is divided by the generator polynomial if reminder is zero then there is no error in the received data and if it is not zero then we received data with error, the output of the syndrome calculator is S(X).

$R(x) / (x + a^i) = Q_i (x) + S(X)/ (x + a^i)$ for $0 <= I <= 2t - 1$

Where Qi = Quotient

$(x + a^i)$ = Factor of the generate polynomial

α^i = Root of the primitive polynomial $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

These remainder S_i resulting from these divisions are known as syndromes.

$S(X) = Qi(x) * (x+ \alpha^{i)} + R(x)$

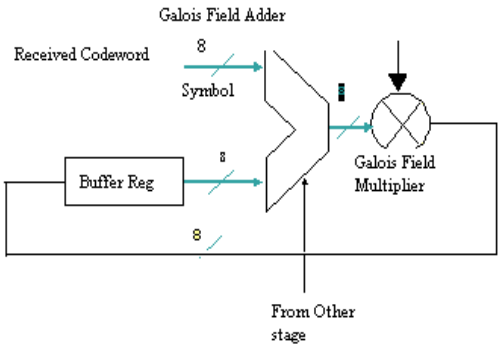

Fig 6: Syndrome Calculator [8]

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.

$$\sigma(X)\ [1 + S(x)] = \Omega(X) \bmod x2t+1$$

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.

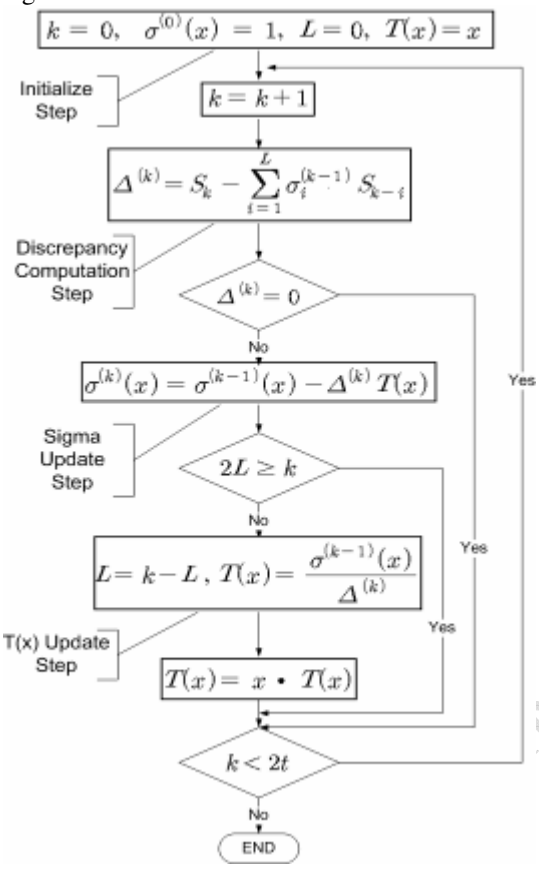

Fig 7: Flowchart of Berlekamp Massey algorithm [8]

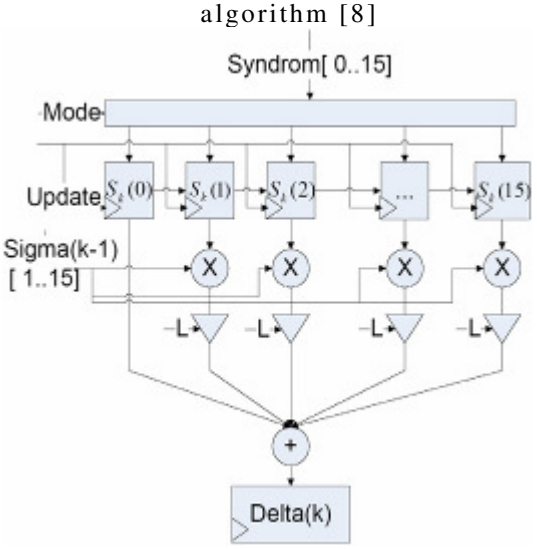

Fig 8: Discrepancy Computation Block [8]

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.

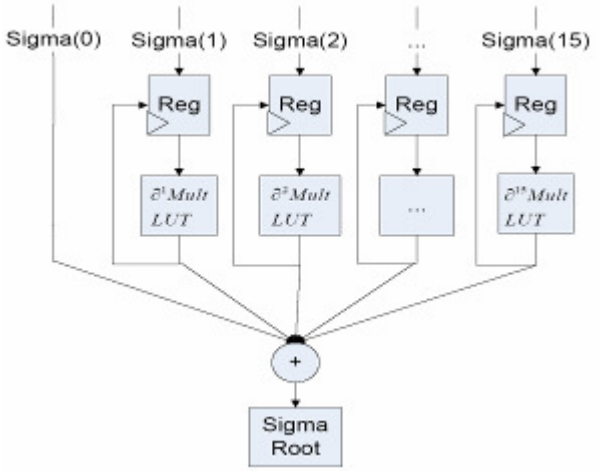

Fig 9: Chine Search Block[8]

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 $\Lambda(x)$ and error magnitude polynomial $\Omega(x)$. The equation for the error values is for $x=\alpha^{-1}$ where α^{-1} is a root of $\Lambda(x)$. The computation of the formal derivative $\Lambda'(x)$ is actually quite simple.

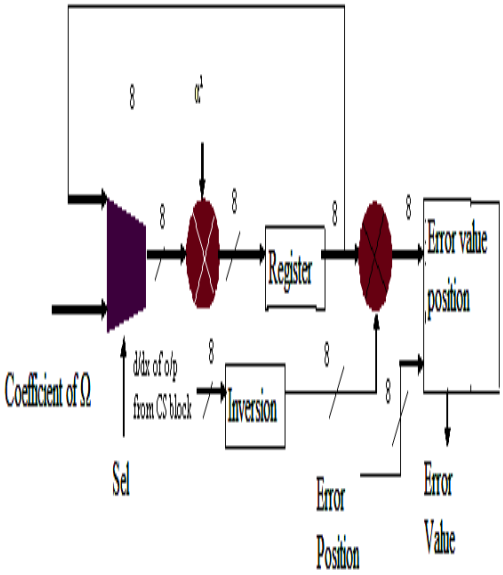

Fig 10: Error Magnitude Block [8]

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.

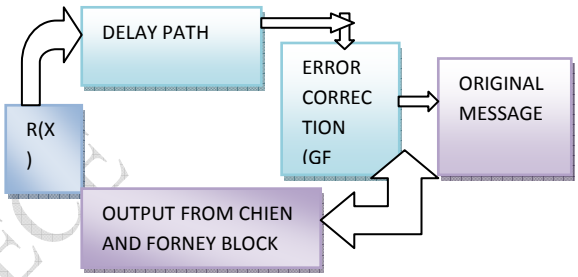

Fig 11: Error Correction Block

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.

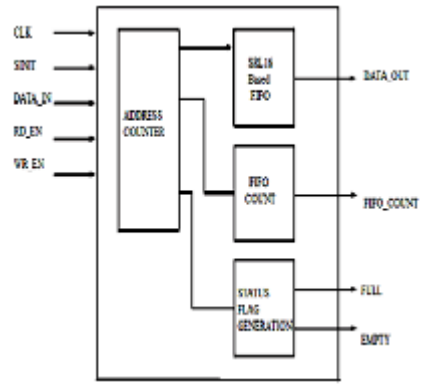

Fig 12: Block Diagram of FIFO[8]

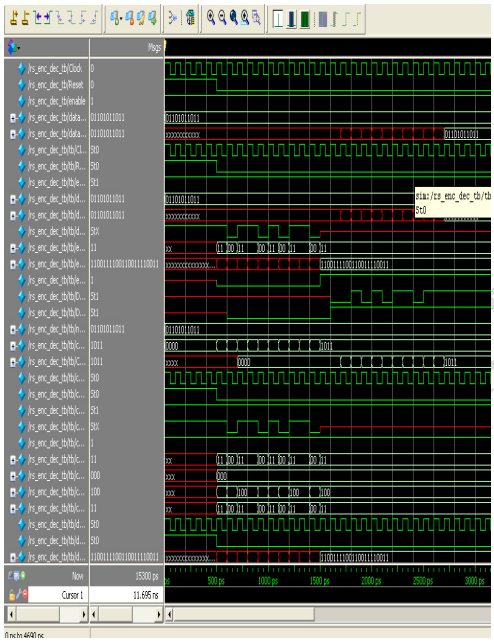

Fig 13: Wave form of RS Decoder

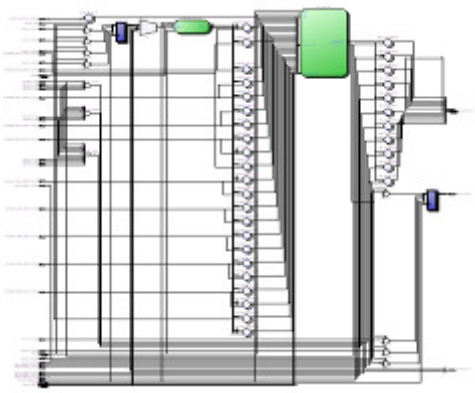

Fig 14: RTL View of Decoder

5. RESULT

	STRETIX III			CYCLONE IV GX		
DEVICE UTILIZATION SUMMARY						
	USED	AVAILABLE	UTILIZATION	USED	AVAILABLE	UTILIZATION
Total Combinational Function	52	38000	<1%	63	14,400	<1%
Number Of Register	16	38000	<1%	16	14,400	<1%
Total Pin	25	296	8%	25	81	31%
TIMING SUMMARY						
Speed Grade	-4			-6		
Maximum Frequency	412.71 MHz			307.98 MHz		
Minimum Period	2.423 ns			3.247 ns		
POWER SUMMARY						
Total Thermal Power Dissipation	398.67 mW			88.65 mW		

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 FPGA chip Stretix III (EP35L50F484C4) and Cyclone IV GX (EP4CGX15BF14C6).The result shows that the decoder could operate at a max frequency of 412.71 MHz for Stretix III and 307.98 MHz for Cyclone IV GX.

By comparing the RS Decoder on two different FPGA ,We conclude that Stretix III higher maximum frequency and occupies less area of FPGA compared to Cyclone IV GX, But total thermal power dissipation in Stretix III higher compared to Cyclone IV GX.

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 FPGA 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.

8. 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&PUNCTU RED Reed Solomon Code.

[9] FPGA Implementation of RS Codec for Digital Video Broadcasting.