Research Paper

Volume : 2 | Issue : 5 | May 2013 • ISSN No 2277 - 8179

Design and Comparision of Rs Encoder and Rs Decoder on Family of Cyclone FPGA Using VHDL

Engineering

KEYWORDS : VHDL, FEC (Forward error Correction), FPGA (Field Programmable Gate Array)

Hitesh G.Kamani

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

ABSTRACT

In this paper design of Reed Solomon (255,239) Encoder and Decoder and perform this Codec on family of Cyclone FPGA 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 VHDL based on Barlekamp Massy, Forney and Chien Search Algorithm.

1. INTRODUCTION

In a communication transmission system, data is transferred from a transmitter to a receiver across a physical medium of transmission or wireless channel [16]. The channel wired or wireless is generally affected by noise or fading, which introduces errors in the data being transferred from sender to receiver [17]. The transmission of data over a specific channel has to be pre defined with specific bandwidth and data rate for evaluation of channel [18]. During transmission of data over a channel, errors are obvious having depth as per prevailing conditions instantaneously [19]. For correcting the errors, we have to use error control strategies. There are two main error control strategies adopted in communication systems:

1. Error correcting techniques
2. ARQ technique

2. OVERVIEW OF RS CODE

Error correcting codes have a wide range of applications in different fields like digital data communications, memory system design, and fault tolerant computer design among others. Reed-Solomon (RS) code is a well known non-binary linear block code. It is popularly used for error correction in many applications like storage devices (CD, DVD), wireless communications, high speed modems and satellite communications. For example RS (28, 24) and RS (32, 28) codes with interleaving are popularly used for storage in CD. A 16 bytes error correcting RS (255, 223) code has been used for digital μ-wave radio. The code RS (255, 239) having 8 bytes error correcting capability has been recommended as a outer code in WiMax. To preserve important header information, MB-OFDM UWB adopts RS (23, 17) codes. (90)

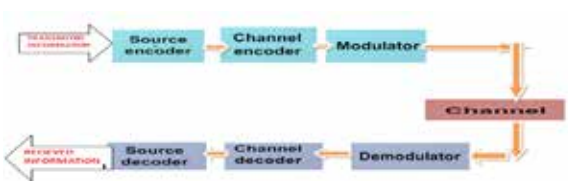

Fig 1: How FEC work [15]

An RS code with 8-bit symbols will use a Galois field GF(28). For RS (255,239) code, n = block length = 255, k = no. of un-coded massage symbols= 239, 2t = (n-k) = number of parity symbols =16, t = maximum number of errors can be corrected =8. The original massage can be recovered by employing the RS decoder provided number of errors in the received codeword is less than or equal to eight

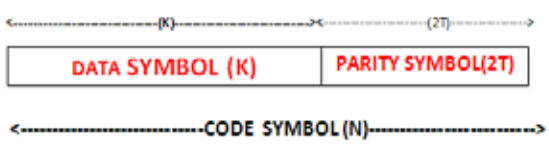

Fig 2: RS(n, k) Codeword[24]

3. REED SOLOMON ENCODER

The Reed Solomon Encoder reads in k data symbols computes the n – k symbols, append the parity symbols to the k data symbols for a total of n symbols. The encoder is essentially a 2t tap shift register where each register is m bits wide. The multiplier coefficients are the coefficients of the RS generator polynomial. The general idea is the construction of a polynomial, the coefficient produced will be symbols such that the generator polynomial will exactly divide the data/parity polynomial. [25] The transmitted codeword is systematically encoded and defined in as a function of the transmitted message m(x), the generator polynomial g(x) and the number of parity symbols 2t as given below.

$$c(x) = m(x)^* 2t + m(x) \bmod g(x) \quad (21)$$

Where, g(x) is the generator polynomial of degree 2t and given by,

$$g(x) = (x + \alpha i)(x + \alpha i+1).(x + \alpha i+2t-2).(x + \alpha i+2t-1) \quad (22)$$

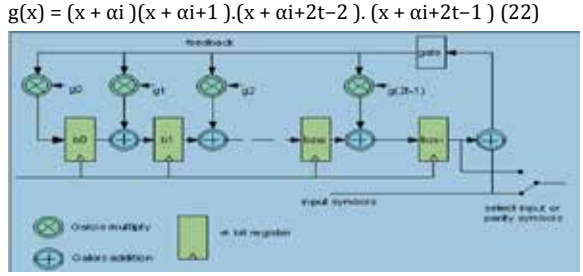

Fig 3: Reed Solomon Encoder Block Diagram[24]

The Encoder architecture shows that one input to each multiplier is a constant field element, which is a coefficient of the polynomial g(x). For a particular block, the information polynomial M(x) is given into the encoder symbol by symbol. These symbols appear at the output of the encoder after a desired latency, where control logic feeds it back through an adder to produce the related parity. This process continues until all of the k symbols of M(x) are input to the encoder. During this time, the control logic at the output enables only the input data path, while keeping the parity path disabled. With an output latency of about one clock cycle, the encoder outputs the last information symbol at (k+1)th clock pulse. Also, during the first k clock cycles, the feedback control logic feeds the adder output to the bus. After the last symbol has been input into the encoder (at the kth clock pulse), a wait period of at least n-k clock cycles occurs. During this waiting time, the feedback control logic disables the adder output from being fed back and supplies a constant zero symbol to the bus. Also, the output control logic disables the input data path and allows the encoder to output the parity symbols (k+2th to n+1th clock pulse). Hence, a new block can be started at the n+1th clock pulse [23].

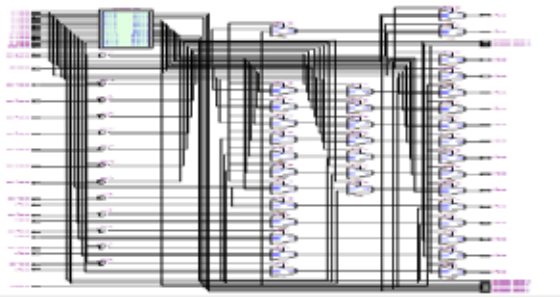

Fig 4. RTL View Of RS Encoder

4. REED SOLOMON DECODER

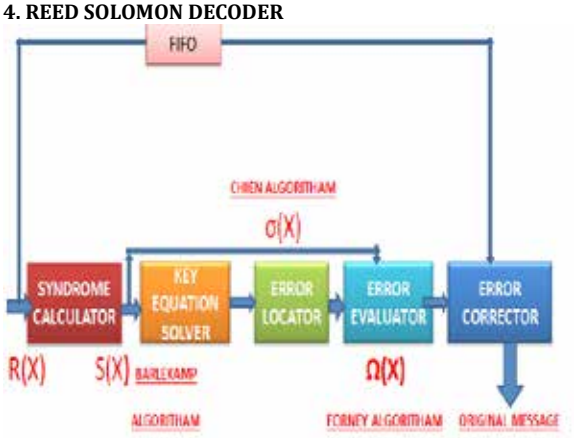

Fig 5: Reed Solomon Decoder Block Diagram

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

$c(X) = c_0 + c_1 X + \ldots + c_{n-1} X$

$r(X) = r_0 + r_1 X + \ldots + r_{n-1} X$

$r(X) = c(X) + e(X)$

$e(X) = r(X) - c(X) = e_0 + e_1 X + \ldots + e_{n-1} X$

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

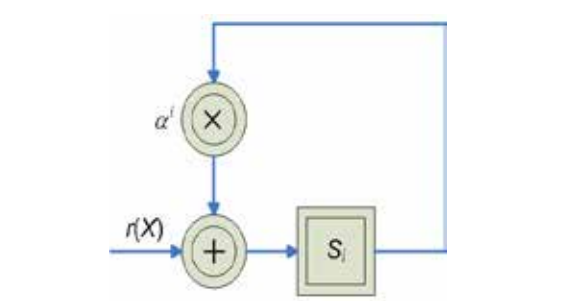

Fig 6: Syndrome Calculator [15]

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.

$$\sigma(X)\,[1 + S(x)] = \Omega(X) \bmod x^{2t+1}$$

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.

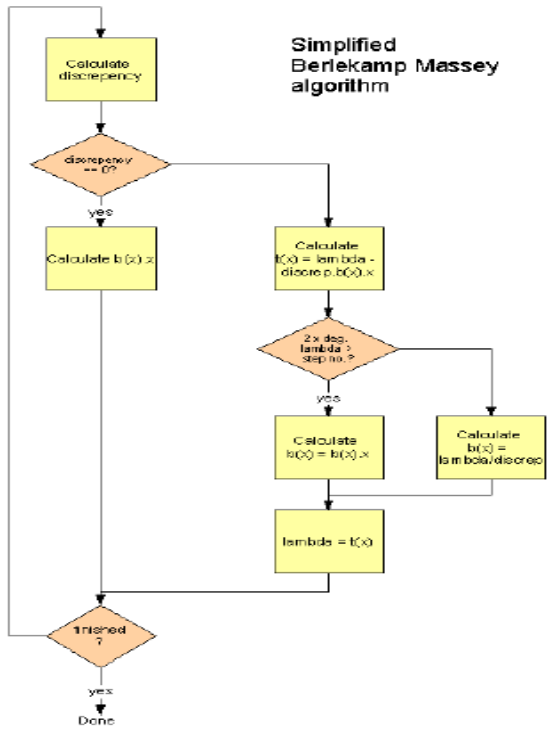

Fig 7: Flowchart of Berlekamp Massey algorithm [24]

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 j is equal to (- i modulo n) = (-I modulo 255). For example, α^{-1} equals α^{254}, α^{-2} equals α^{253} and so on.

Volume : 2 | Issue : 5 | May 2013 • ISSN No 2277 - 8179

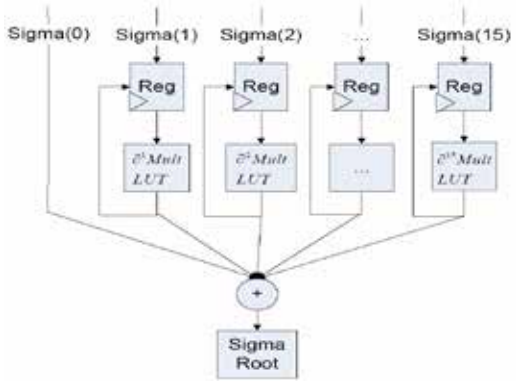

Fig 8: Chine Search Block[8]

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 σ (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 σ (x). The computation of the formal derivative σ '(x) is actually quite simple.

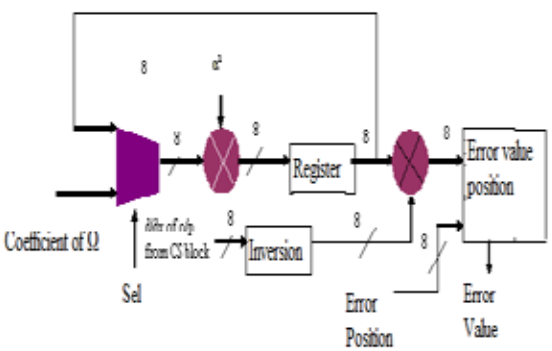

Fig 9: Error Magnitude Block [8]

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.

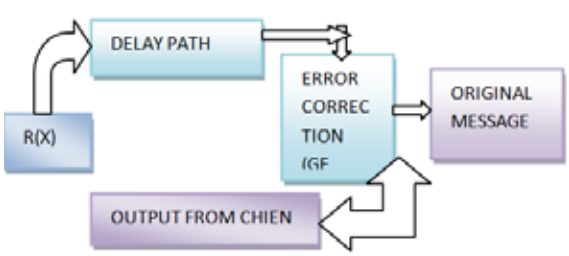

Fig 10: Error Correction Block

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.

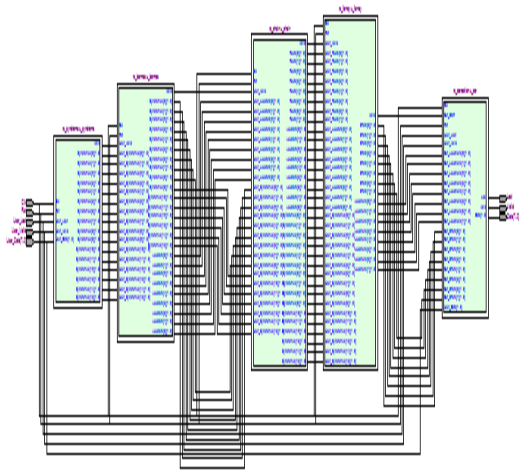

Fig 11: RTL View of Decoder

5. RESULT

Compile: Quartus II 10.1
Analysis & Synthesis Setting: Smart Compile
Compile Process Setting : Balanced

	CYCLONE		CYCLONE II		CYCLONE III	
	DEVICE UTILIZATION SUMMURY					
	USED	AVAILABLE	USED	AVAILABLE	USED	AVAILABLE
Total Logic Element	268	5980	272	4608	274	55856
Number Of Register	190	----------	190	---------	190	----------
Total Pin	23	185	23	158	23	328
	TIMING SUMMURY					
Speed Grade	-6		-6		-6	
Maximum Frequency	255.95MHz		306.04MHz		357.78MHz	
Minimum Period	3.9nS		3.3nS		2.8nS	
	POWER SUMMURY					
Total Thermal Power Dissipation	60.04mW		31.72mW		109.01mW	

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

	CYCLONE		CYCLONE II		CYCLONE III	
	DEVICE UTILIZATION SUMMURY					
	USED	AVAILABLE	USED	AVAILABLE	USED	AVAILABLE
Total Logic Element	110	2910	108	68416	108	55856
Number Of Register	70	----------	70	---------	70	----------
Total Pin	22	65	23	622	22	328
	TIMING SUMMURY					
Speed Grade	6		6		6	
Maximum Frequency	193.39MHz		275.56MHz		299.13MHz	
Minimum Period	5.2nS		3.6nS		3.3nS	
	POWER SUMMURY					
Total Thermal Power Dissipation	48.04mW		195.75mW		108.90mW	

Table 2: the Comparison of Hardware and Timing and power Performance of RS Decoder

6. CONCLUSION

In this paper, FEC Encoder and Decoder has been presented to remove error in wireless communication.FEC Encoder and 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 hardware description language,VHDL and synthesized for the FPGA chip CYCLONE family. The result shows that the Encoder could operate at a max frequency of 357.78 MHz for FPGA CYCLONE III.By comparing the RS Encoder on different CYCLONE FPGA, We conclude that CYCLONE III FPGA higher maximum frequency and occupies more area of FPGA compared to other CYCLONE family chip, But total thermal power dissipation in CYCLONE III is low compared to CYCLONE II chip. For Decoder The result shows that the Decoder could operate at a max frequency of 299.13 MHz for FPGA CYCLONE III.By comparing the RS Decoder on different CYCLONE FPGA, We conclude that CYCLONE III FPGA higher maximum frequency and occupies more area of FPGA compared to other CYCLONE family chip, But total thermal power dissipation in CYCLONE III is low compared to CYCLONE II 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 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.

REFERENCE

[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 Theor y 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 | [16] Berrou, C., Glavieux, A. and Thitimajhima, P., "Near Shannon limit error correcting coding and decoding: turbo-codes," Proc. Of ICC '93, Geneva, May 1993, pp. 1064-1070. | [17] Berrou, C. and Glavieux, A., "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," IEEE Trans. on Communications, vol. 44, no.October 1996, pp. 1261-1271 | [18] Benedetto, Sergio and Montorsi., Guido "Design of parallel concatenated convolutional codes,. IEEE Transactions on Communications, vol. 44, No. 5, May 1996, pp. 591-600. | [19] Blackert, W. J., Hall, E. K., and Wilson, S. G., "Turbo code termination and interleaver conditions," Electronic Letters, vol.31, No.24, November 1995, pp. 2082-2084. | [20] Design of RS (255, 251) Encoder and Decoder in FPGA | [21] I.S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," SIAM Journal of Applied Mathematics, vol. 8, 1960, pp. 300– 304. | [22] R. J. McEliece, Finite Fields for Computer Scientists and Engineers. Boston, MA: Kluwer Academic, 1987. | [23] S. Kaur "VHDL Implementation of Reed-Solomon code," Thesis,Thapar Institute of Engg, 2006. | [24] Reed Solomon Codes Electrobits | [25] "Reed-Solomon (RS) Coding Overview," VOCAL Technologies, Ltd., Rev. 2.28n, 2010. |