International Journal of Engineering & Technology, 6 (4) (2017) 171-180

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v6i4.8205
Research paper

FPGA implementation of RS codec with interleaver in DVB-T using VHDL

Sara Kamar [1]*, A. Fouda [1], Abdelhalim Zekry [2], A. El-Mahdy [1]

[1] *Modern Academy, Egypt*
[2] *Ain Shams University, Egypt*
**Corresponding author E-mail: sara-kamar@hotmail.com*

Abstract

Digital television (DTV) provides a huge amount of information to many users at low cost. Recently, it can be packaged and fully integrated into completely digital transmission networks. Reed-Solomon code (RS) is one type of error correcting codes that can be used to enhance the performance of DTV. Interleaving/deinterleaving process enhances the performance of channel errors by spreading out random errors, very high-speed hardware description language (VHDL) is used in electronic design automation. It can be used as a general-purpose parallel programming language.
This paper presents VHDL program for Reed-Solomoncodec (204, 188) and convolutional interleaver/deinterleaver, used in Digital Video Broadcasting-terrestrial system (DVB-T), according to ETSI EN 300 744 V1.5.1 standard. The VHDL programs are implemented on Xilinx 12.3 ISE and then simulated and tested via ISE simulator then the code is synthesized on FPGA device the results are compared with IP core for Xilinx 12.3 ISE, which gives the same results.

Keywords: *Convolutional interleaver/Deinterleaver; DVB-T; Outer Coding; RS (204,188); VHDL.*

1. Introduction

Digital television can deliver more programs than traditional analog television over any transmission media. This is because digital information can be manipulated in ways never possible with analog television. In the analog TV, many terrestrial networks were government owned because of the large capital cost of network equipment, which gave protection from competition to the existing network operators. However, with digital technologies, there will be a great number of bandwidth available for other program providers and for program makers.
The DVB project has been developed for broadcasting on different media such as cable, satellite, as well as terrestrial channels where it is referred to as the DVB-T standard, by the European Broadcasting Union (EBU) headquarters in Geneva and supported by the European Commission. [1].
An important part of the DVB_ T system is outer encoder/decoder, code, which uses Reed-Solomon code with convolutional interleaver/deinterleaver.
Reed-Solomon codes are blocked error correcting codes used to correct errors on several applications in digital communications and storage, such as DVB where Reed-Solomon encoder takes a block of digital data and adds extra "redundant" symbols.
The Reed-Solomon decoder processes each coded block and correct errors to restore the original data. An interleaver should exist to randomize the effect of channel errors. To spread out the burst errors in time such that it could be better handled by the decoder if they become random errors, Interleaver before transmission and deinterleaver after reception could be used. Since, in all practical cases, the channel memory decreases with time separation. [2]
There were efforts to implement Reed-Solomon encoder and decoder with the same parameter of this paper. [3], [4], [5] but this paper added the integration of RS encoder with interleave, Dieterle to make the whole outer codec block for DVB-T. The implemented devices are compared with the results of corresponding IP core of RS (204, 188) which is already implemented on Xilinx.
The paper is organized as in the following: section (2) includes DVB-T system model, section (3) is about R-S codec implementation. Section (4) covers the convolutional interleave implementation. Section (5) contains the numerical results, and the paper is concluded in section 6.

2. System model

There are two types of the communication systems based on the signals that are transmitted, digital communication systems and analog communication systems. Because of of the possibility of error control and encryption, the digital communication system is often used nowadays. Examples of communication systems in modern life are television system.
Fig. 1 presents the basic block diagram for DVB-T Transmitter where the source coded video data is channel coded by an outer encoder followed by an inner error encoder and then orthogonal frequency division multiplexed before it is ent to the rf front end. The DVB-T receiver performs the inverse operations on the received signal from the rf front end to extract the video data. Accordingly, the receiver block diagram is shown in Fig. 2 where the outer decoder comes after the OFDM demodulation and the inner decoder.

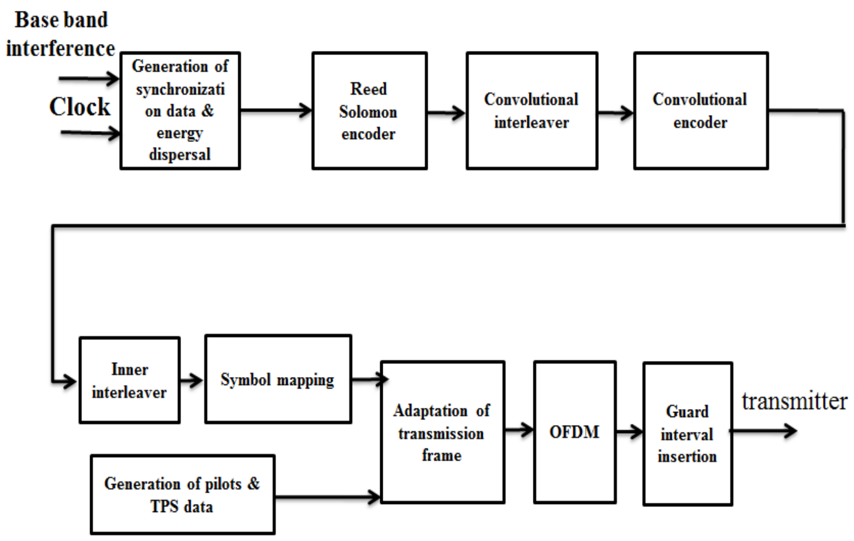

Base band interference
Clock
Generation of synchronization on data & energy dispersal
Reed Solomon encoder
Convolutional interleaver
Convolutional encoder
Inner interleaver
Symbol mapping
Adaptation of transmission frame
OFDM
Guard interval insertion
transmitter
Generation of pilots & TPS data

Fig. 1: Block Diagram for DVB-T Transmitter.

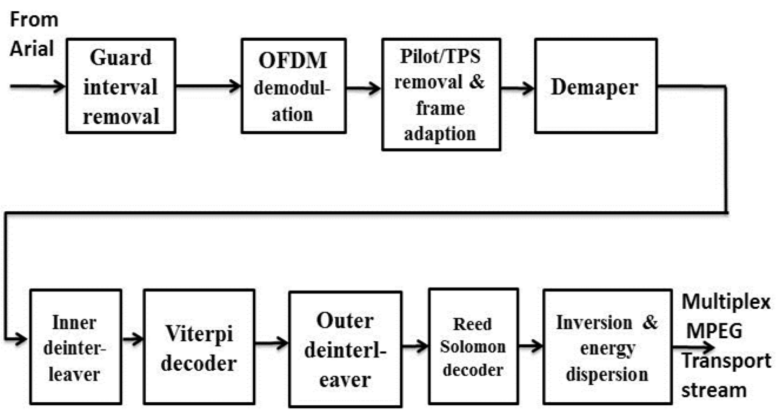


Fig. 2: Block Diagram for DVB-T Receiver.

3. Reed-Solomon code

Reed-Solomon codes are systematic, non-binary, linear, block cyclic code. A systematic code generates code words that contain the message symbols with unchanged form, Non-binary codes are dealing with symbols that consist of several bits while a code is linear if the addition of any two valid code words also results in another valid codeword. Block code means the encoder processes a block of message symbols and then outputs a block of code word symbols. The encoder makes mathematical operations to the message symbols to generate the parity symbols which will be added to the message symbols to generate the codeword. Finally, the code is acyclic code if a circular shift of any valid code word also produces another valid codeword. [2]

RS codes are generally represented as an RS (n, k), with m-bit symbols, where m is any positive integer having a value greater than 2.

k is the number of data symbols being encoded.

n is the total number of code symbols in the encoded block.

For the most conventional R-S (n, k) code:

$$(n, k) = (2^m\text{-}1, 2^m - 1 - 2t) \quad (1)$$

Wheretis the symbolerror correcting capability of the code, and n - k = 2t, is the number of parity symbols.

A codeword based on these parameters is shown diagrammatically in Fig. 3.

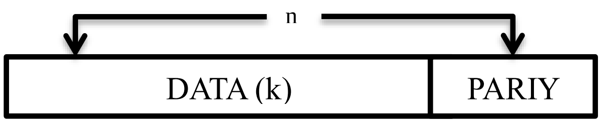


Fig. 3: Reed-Solomon code word.

The most important advantage of Reed-Solomon codes over binary linear block codes is the ability to correct the burst errors.

3.1. Reed-Solomon codes

Reed-Solomon constructed from extended Galois field GF (p^m) with p^m elements from the original finite field GF (p) which contains p (prime) elements. The 0’s and 1’s representing the binary field GF (2) is a subfield of the extended field GF (2^m).

There is a new symbol (α) which represents the additional elements in the extended field. A power of α represent each nonzero element in GF (2^m).Starting with the elements {0, 1,α}, an infinite set of elements, F, is generated by multiplying the last element by α. so, the elements of the finite field GF$(2^m)$are given by [6]:

$$\mathrm{GF}(2^m) = \{0, \alpha^0, \alpha^1, \alpha^2 \ldots \ldots\ldots, \alpha^{2^m-2}\} \quad (2)$$

3.2. Reed-Solomon encoding

Converting an input message into a corresponding codeword is the encoding process, where each codeword ∈GF (2^m). A polynomial g(x) of degree n-k with coefficients from GF (2^m) generates the RS codeword. Adding the parity symbols to the information

symbols is forming the codeword. Any RS encoder design performs two operations, division and shifting.
According tot-errors correcting RS code, generator polynomial is given by [6]:

$$g(x)=(x+\alpha^{i})(x+\alpha^{i+1})\ldots\ldots(x+\alpha^{i+2t-1}) \quad (3)$$

Where α is a primitive element in GF (p^{m}). The parity polynomial p(x) is expressed by

$$p(x) = m(x)\, x^{n-k} \text{ modulo } g(x) \quad (4)$$

The resulting code word polynomial u (x):

$$u(x)=p(x) + m(x)\, x^{n-k} \quad (5)$$

Where p (x) represents the parity symbols, and m (x) the message symbols in polynomial form.
These operations can be implemented using Linear Feedback Shift Register [7], as depicted in Figure 4

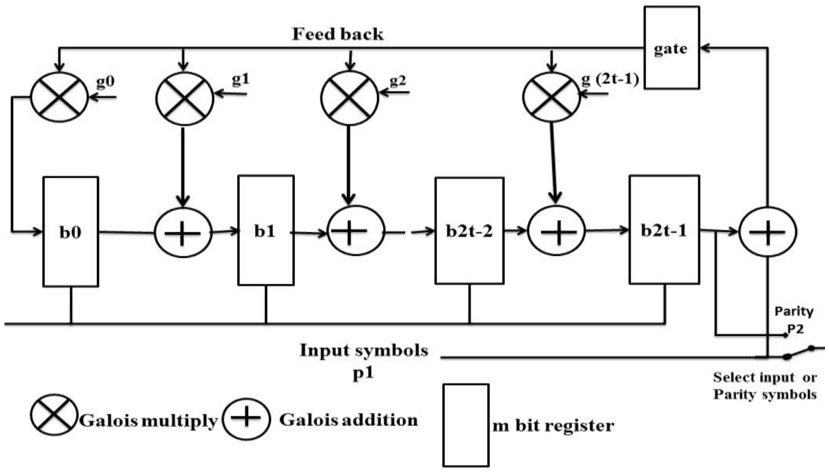


Fig. 4: RS Encoder Circuitry.

The working procedure of the circuit in Fig. 4 is described as: The initial states of the registers b0 ~ b2t-1are all 0. At the beginning, the switch S is on p1. At this moment, the input sequence m(x) is transmitted to two destinations, one to the output, and the other, after multiplied byX^{n-k}, to the division circuit.
After k clock cycles, the whole message sequence m(x) is sent through the circuit, and the division arithmetic is executed. The values saved in the registers are the coefficients of the remainder r(x), i.e. the 2t parity check symbols.
At the k+1 clock cycle, the switch is switched to p2. Then, the data into the registers are shifted out one by one. After 2t=n-k clock cycles, all of the data are transmitted to the output end and from the codeword u(x) along with the k information symbols of the message sequence m(x). This is the encoding of one block of code. To encode the next block one resets all registers b0~b2t-1 to the initial states of 0 and repeats the above procedures to perform the encoding of the next block of code. [7]

3.3. Reed-Solomon decoding

According to Fig. 5, the decoder consists of the following parts:

1) Syndrome S(x) Calculation.
2) Solving error location polynomial σ (x) and error-value evaluator polynomial ω (x) using Berlekamp-Massy algorithm.
3) Find the roots of σ(x) using Chien search.
4) Find the error values using the Forney Algorithm.
5) Correct the received word C (x) = e(x) + r (x), where e(x) is error symbol polynomial and r(x) is the received polynomial.

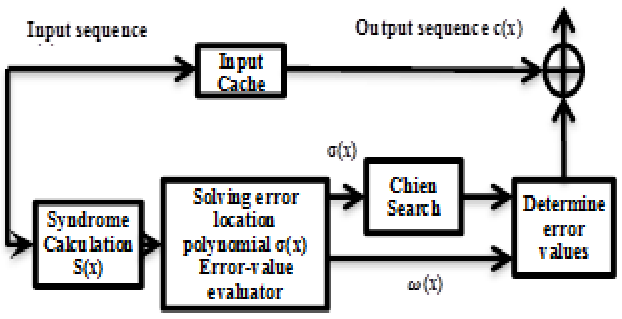


Fig. 5: Block Diagram of the Decoder.

3.3.1. Syndrome calculation

The first step in decoder process is to determine whether the received code word (r) is a valid member of code word set, by performing a parity check on received code word (r) to calculate syndromes. If(r) is a member of the code word, then the syndromes S has value 0. If any syndrome has value, then there is an error [6].
The syndrome Si is computed from the expression:

$$S_i = r(x)|_{x=\alpha^{i}} = r(\alpha^{i}) \; i=0, 1\ldots \text{n-k-1} \quad (6)$$

If the resultant S_i=0, then there is no errors in the received signal, otherwise error occurred.

3.3.2. Key equation solving

This requires solving 2t simultaneous equations, one for each syndrome. There is an equation that is known as Key-equation used for this purpose and is given as [3]

$$\{\sigma(x)\, S(x) + \omega(x)\} \bmod (x^{2t}) = 0 \quad (7)$$

Where σ(x) is the error locator polynomial and ω(x) is the error evaluator polynomial, There are two techniques for solving the key equation; the Berlekamp algorithm [8], and Euclidean algorithm, [8], [9]. The technique used in this paper is the Berlekamp-Massey algorithm because this algorithm has less hardware complexity.

3.3.3. Chien search

After calculating σ(x), its roots should be found, which is a polynomial whose roots are constructed to be the inverses of the locations where the errors occurred. To find the roots of error location polynomial we substitute all the finite field elements in the error location polynomial σ(x) and fined the condition $\sigma(\alpha^i)$ =0. The Chien search is effective algorithm to do this search. [10], [11].

3.3.4. Forney algorithm

After the errors are located, the error locations can be substituted into the syndromes to obtain the error value. Since the calculation scale is too large for matrix inversion, Forney's algorithm, [12], [13] is normally adopted to determine the error values.

3.3.5. Correcting the received word

After calculating the error locations and error values, at each error location, the received symbols are XORed with the error symbol. In this way the decoder corrects any error.

4. Convolutional interleaver

One of the most popular ways to correct burst errors is to take a code that works well on random errors and interleaves the bursts to "spread out" the errors so that they appear random to the decoder. There are two types of interleaves commonly in use today, block interleave and convolutional interleave. This paper is concerned with convolutional interleave.

Convolutional interleaves to have been proposed by Ramsey and Forney. The structure proposed by Forney appears in Figure-6. The code symbols are sequentially shifted into the bank of Nth registers. Each successive register provides J symbol more storage than did the preceding one. The zero register provides no storage where the symbol is transmitted immediately. With each new code symbol, the commentator switches to a new register, and the new code symbol is shifted in while the oldest code symbol in that register is shifted out to the modulator or transmitter. [6]

After the (N- 1)th register, the commutator returns to the zero register and starts again. The deinterleaver performs the inverse operation, and the input and output commutator for both interleaving and deinterleaving must be synchronized. [6].

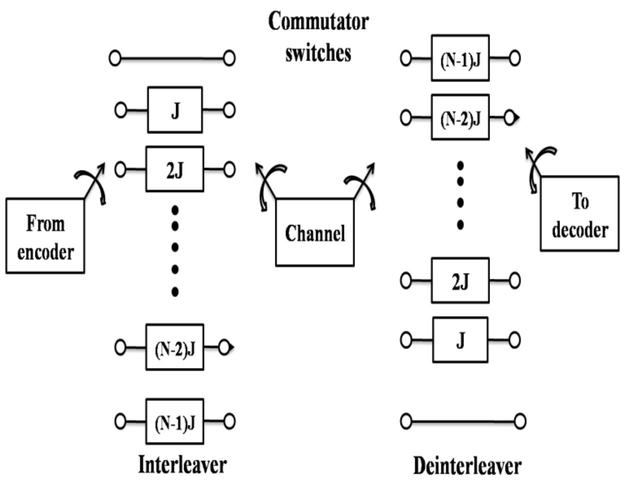


Fig. 6: Shift Registers Implementation of Interleaver/Deinterleaver. [5].

5. VHDL modeling and simulation results

5.1. System specifications

The specification of RS codec according to European Telecommunications Standards Institute (ETSI) is RS (204,188, t = 8) shortened code, derived from the original systematic RS (255,239, t = 8) code. The interleaver performs a convolutional byte-wise interleaving process based on the Forney approach which is compatible with the Ramsey type III approach. The interleaver is composed of I = 12 branches, cyclically connected to the input bytestream by the input switch. Each branch j shall be a First-In, First-Out (FIFO) shift register, with depth j × M cells, where M = 17 = N/I, with N = 204. [14]

These functions are implemented using VHDL code and simulated on Xilinx ISE 12.3. Then the code is synthesized on Virtex 6 (XC6VLX240T) FPGA. The design is verified by the loop back technique where the data output from the decoding process is identical to that input to the encoder. In addition, RS (204,188) encoder simulation result is compared with IP core RS (204,188) which already implemented on Xilinx.

For the validation of the design, its results for RS encoder are compared with that of the IP core of Xilinx.

The Code Generator Polynomial for RS (204,188) for DVB-T is given by [13]:

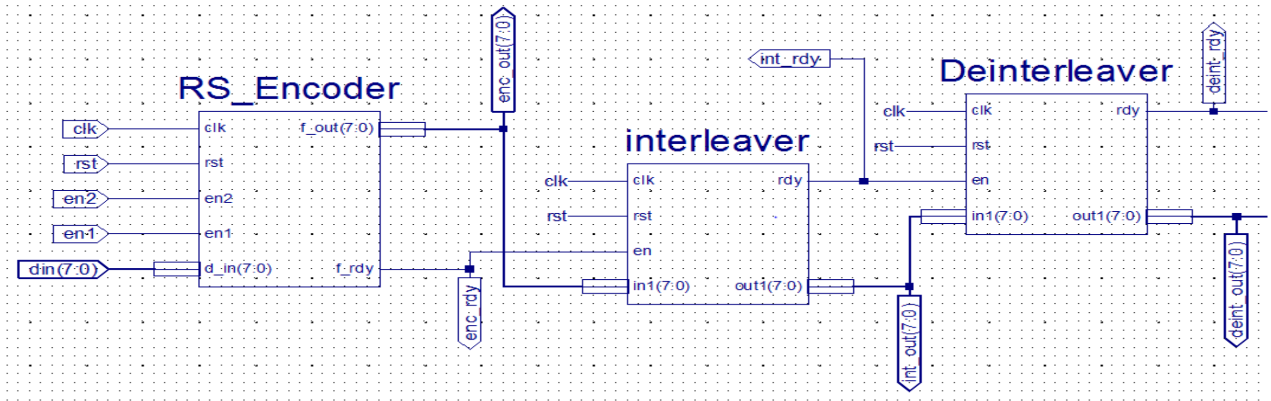
$$g(x) = (X+\alpha^0)(X+\alpha^1)(X+\alpha^2)\ldots\ldots\ldots(X+\alpha^{15}) \qquad (8)$$

And its Field Generator Polynomial follows the expression [14]:

$$p(x) = X^8 + X^4 + X^3 + X^2 + 1 \qquad (9)$$

5.2. System modeling and simulation

5.2.1. The integrated full system model



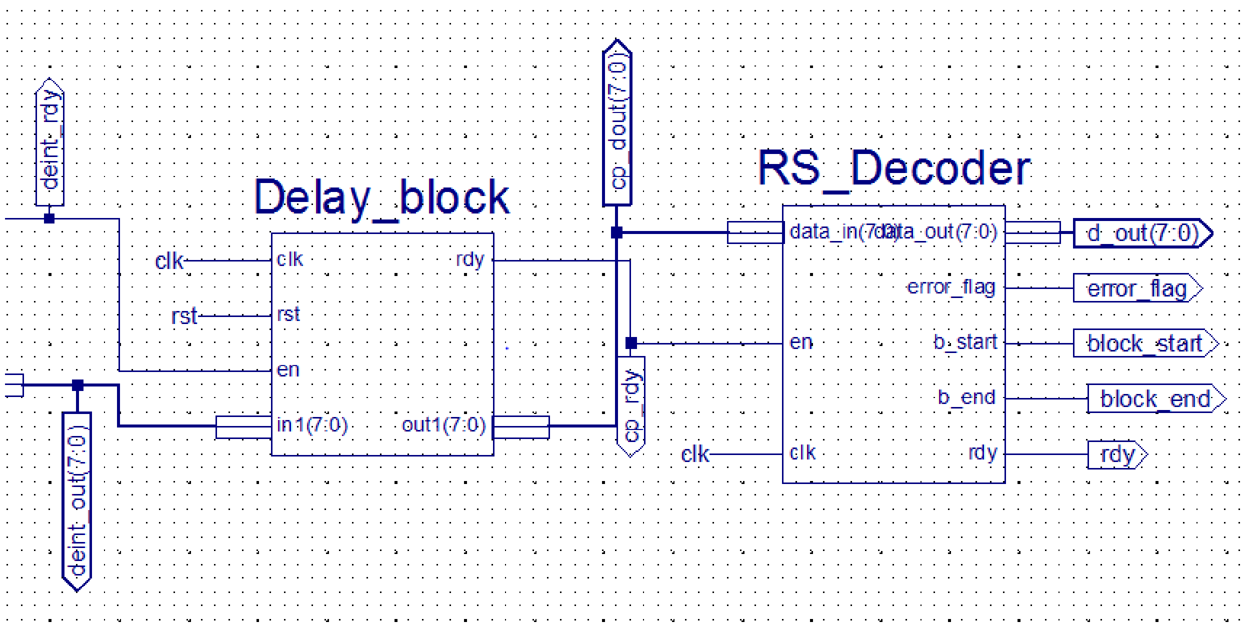


Fig. 9: Overall Integrated System.

The designed RTL model of the integrated outer codec is depicted in Fig.9. It consists of the model for the RS encoder, the convolutional interleaver, the convolutional deintleaver, a delay block and the RS decoder. The logic transition occurs at the rising edge of the clock “clk”and the “rst” is the reset button for all building blocks. The pin assignment of all blocks is indicated in the same Fig.9.
The functions of the pins can be described as follows: (en1) the encoder starts working then (en2) the parity come out, (in(7,0)) is input to encoder, (enc_out(7,0)) is the output for encoder and input for interleaver, (enc_rdy) the output come out and interleaver ready to take data, (int_put(7,0)) output for interleaver and input for deinterleaver, (int_rdy) the interleaver out data, and deinterleaver ready to take it, (deint_out(7,0)) out for deinterleaver and input for Delay block(which use for synchronization), (deint_ rdy) deinterleaver ready to output data and Delay block ready to accept it, (cp_out(7,0)) Delay block out and decoder input, (cp_ready) Delay block is outputing data and the decoder starts work, (d_out(7,0)) decoder out, (block_start) decoder starting data output, (block_end) decoder is finished, (rdy) the decoder ready for outputing data, (error_flag) indication if there errors at the decoder input. The encoder block receives an input of k symbols=188 byte sequentially, encodes then by adding extra redundant bits parity symbols of 16 bytes. Then the n output bytes of the encoder of 204 bytes are entered to interleaver to spread out errors in timeand the output of interleaver is still 204 bytes. The next block is deinterleaver which returns back the n bytes to their original arrangement. The delay block input is as the same as the output of deinterleaver. The decoder input must be continuous without delay but the deinterleaver takes delay time to exit its output. The function of delay block is to store the output of deinterleaver and transmit it to decoder.The final block is the decoder and it corrects the errors occurred by the channel. The detailed operation and the resulting waveforms for all functional blocks are presentedin the next section.

5.2.2. The waveform for the overall system

Figure (10.1) presents the input/output data for the encoder and the output of interleaver and their operations are as follow:
At the rising edge of the enable to the encoder (en1=1) the data is entered to the encoder which in this example is 33Hex for the first byteand 55Hex for the other 187 bytes. Then the encoder starts doing its operation. After finishing its operation it raises an output flag encoder ready (enc_rdy=1), which mean that the encoder is ready to output its data through (enc_out[7:0]) pins which is the input for interleaver. At the next rising edge the interleaver starts doing its operation then it is ready to out data through (int_out [7:0]) pins, which also is the input for deinterleaver. The output (int_rdy=1) is a flag to indicate that the output of the interleaver is available. The deinterleaver waveform is in the next figure.
Fig.10.2 presents the output waveform of the deinterleaver and its operation is as follows:
After the data entered to deinterleaver, it starts doing its operation and then when the flag (deint_rdy) pin is equal one, it becomes ready to deliver its output through (deint_out[7:0]) pins. It is clear that the data has its original order, 33hex then 55hex, 55hex …etc. The next figure presents the encoder parity and delay block output. This validates the correct logical operation of the whole system as the original data is restored after loop back.

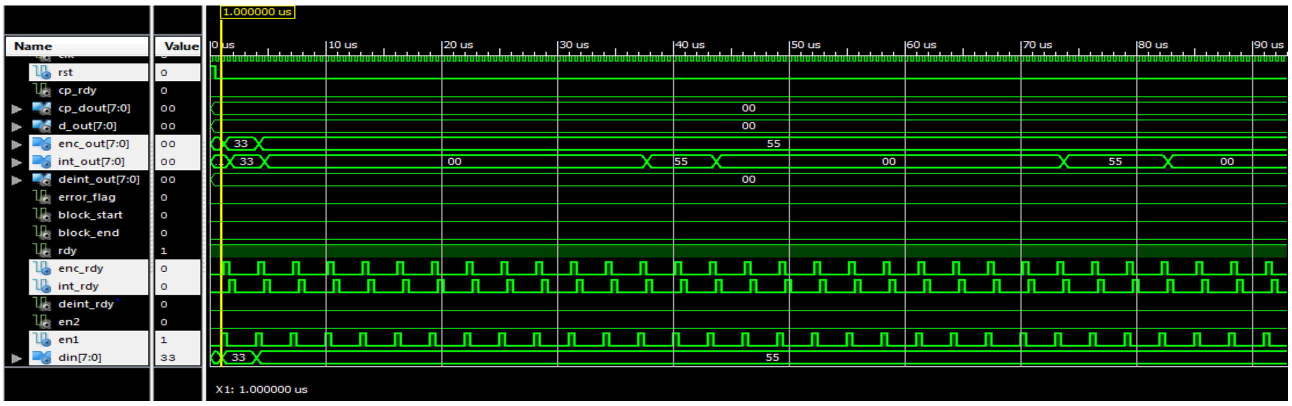


Fig. 10.1: The System Waveforms showing Data Flow to and from the Encoder.

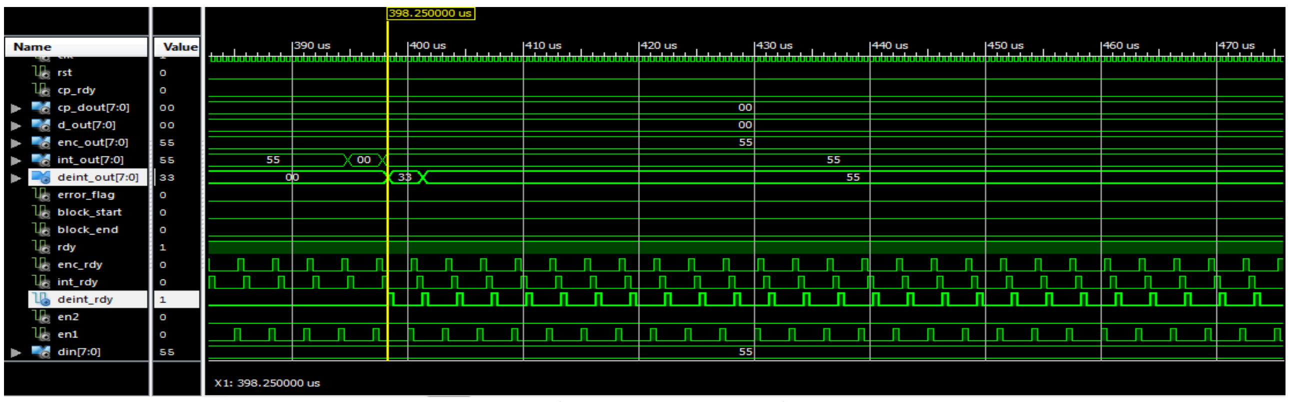


Fig. 10.2: The system Waveform Showing Data Flow from the Deinterleaver.

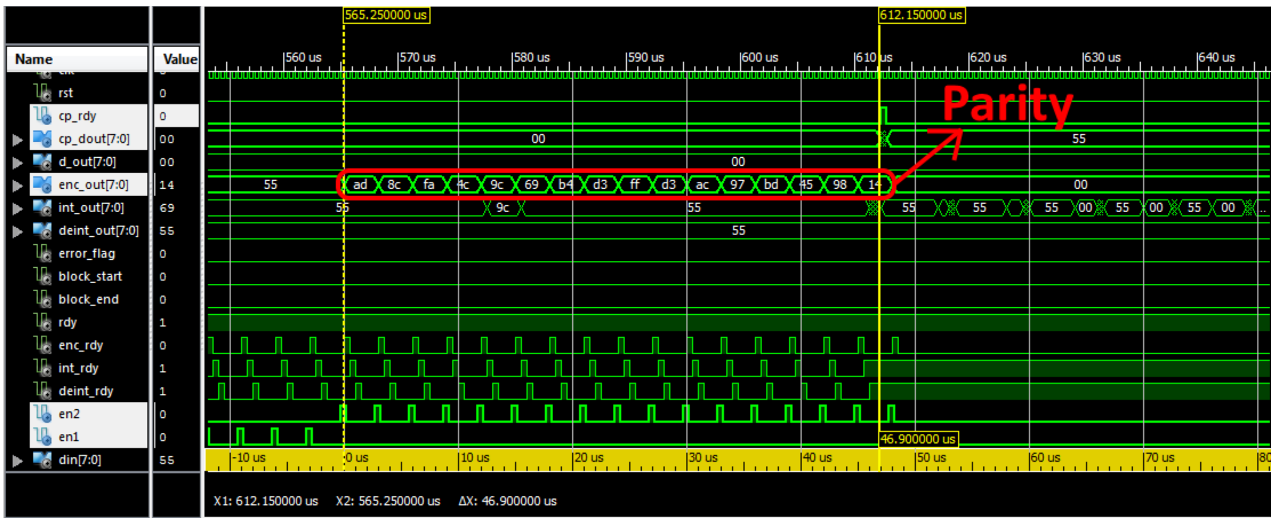


Fig. 10.3: The Overall System Waveform Showing the Parity for The encoder.

Fig. 10.3 depicts the value of parity which has been calculated by the encoder and the output for delay block and their operations are as follow:
After all188 byte input data is output the (en1) bin becomes equal zero and (en2) bin becomes equal one to make the switch go to position 2 (refer to section (2.2)) and the 16 byte parity are output from encoder and its values as clear from the figure are (ad,8c,fa,4c,9c,69,b4,d3,ff,d3,ac,97,bd,45,98). For the delay block, (cp_rdy) bin becomes equal one when the delay block output is ready to output data through (enc_out [7:0]) pin. Fig. 10.4 is presenting the parity outputting from the dent_out [7:0] pin of the deinterleaver.

The deinterleaver finish its job after returning back the parity of encoder as the same arrangement of the encoder; then (deint_rdy) pin becomes equal to zero when deinterleaver returns back the last symbol of the parity. The next figure is about decoder.
Fig. 10.5presents the output waveform of the data from the decoder where after the decoder corrects all possible errors , (Block_start) pin goes high to indicate that the decoder has been finished its work and output starts to come out through (d_out [7:0]) pin. It is clear from the figure that the decoder output is equal to the input for encoder which is 33Hex for first byte and the last 187 bytes are equal to 55Hex. The parityappearsin the next figure, Fig. 10.6.

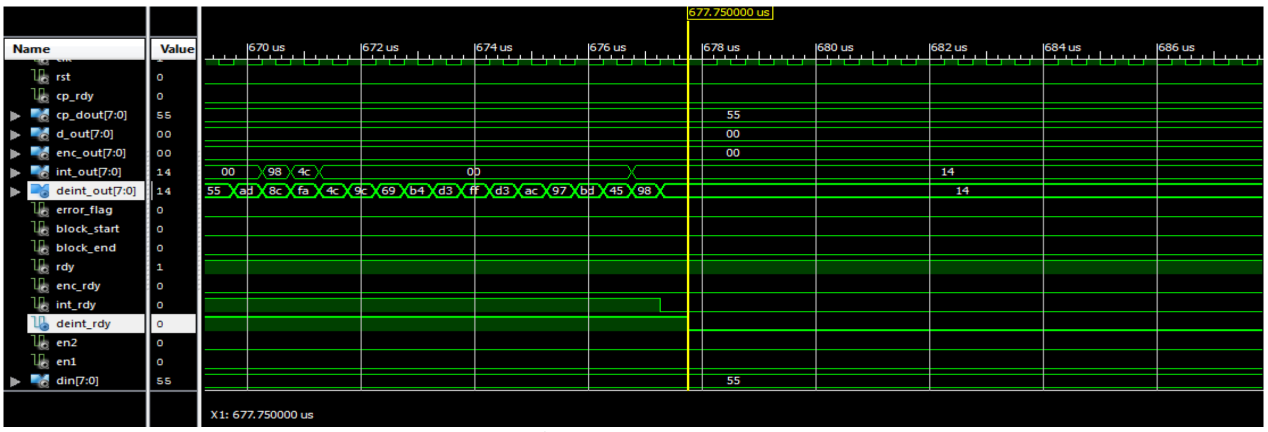


Fig. 10.4: The System Waveform Showing the Deinterleaver Outputting Parity.

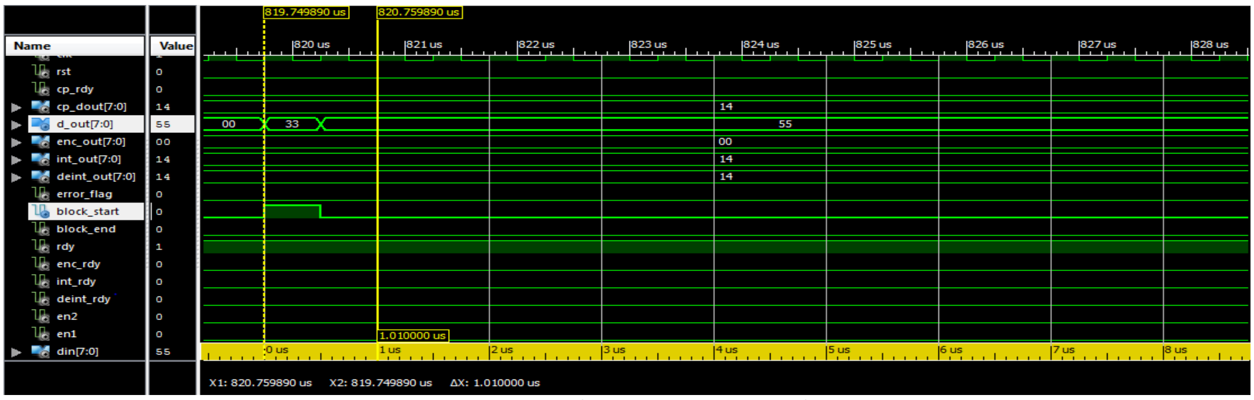


Fig. 10.5: The System Waveform Showing Data Flow from Decoder.

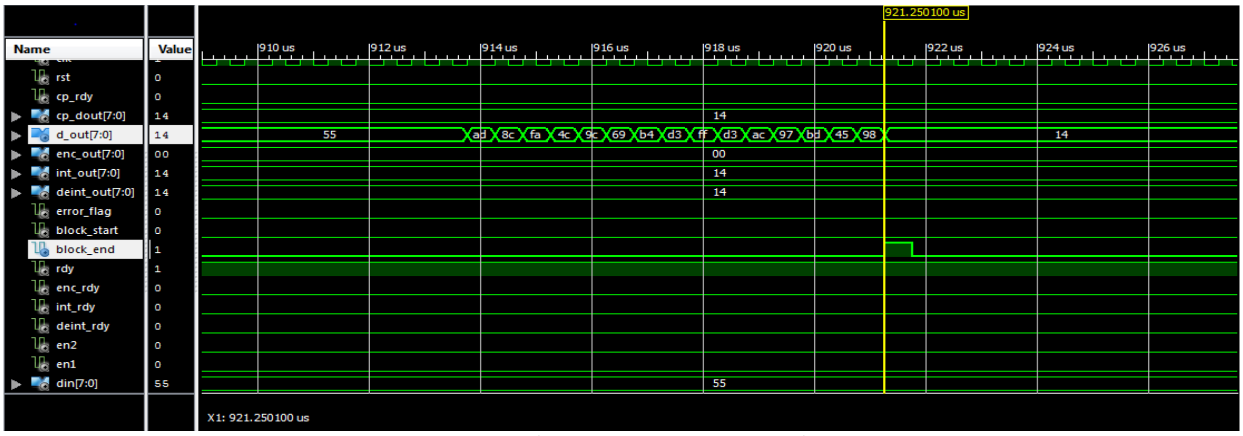


Fig. 10.6: The System Waveform Showing the Parity Flow from The decoder.

Fig. 10.6 presents the remaining output for the decoder. To make sure that the decoder corrects all errors at the input data its output must be as same as the encoder input and the parity which is added by the encoder and this figure depicts the remaining input and the parity from the decoder which is the same as that of the encoder proving the logical correct operation of the designed system.

5.2.3. The waveforms for the IP core for the RS codec

Fig 11.1 presents the input and the output for the RS IP core of Xilinx. To verify our RS codec design we compared its performance with that of the RS IP core from Xilinx. The same test vector was used and so it is expected to get the same output vectors of our design if our design is logically correct.Fig.11.1 shows that the input for IP core is the same as the input for our design, and also the output is the same as our design. .

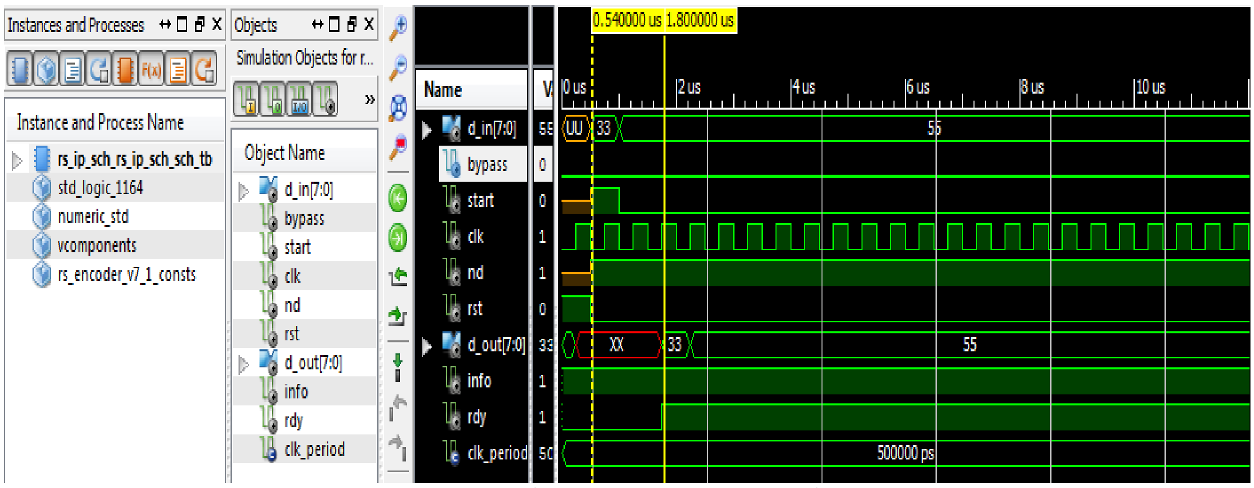


Fig. 11.1: The RS IP Core Waveform Showing Data Flow to and from IP Core Encoder.

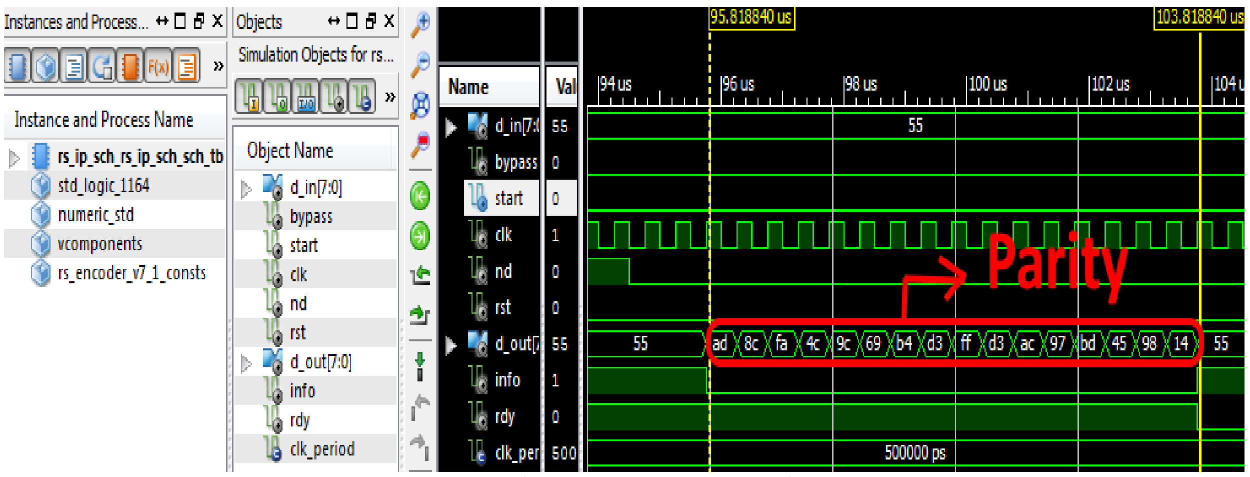


Fig. 11.2: The IP Core Waveform Showing the Parity Flow from IP Core Encoder.

Fig.11.2 shows also that the parity for IP core is as same as the parity for our encoder.

5.2.4. Testing the error correct ability of the designed RS codec

The simulation results depicted in Fig 13.1 shows that it is assumed that the original data is intentionally corrupted at the first 8 bytes of the data. The function of decoder is to restore back the data in correct form.Fig.13.1 shows that the input data become cc Hex at the first byte and aaHex at the following 7 bytes which are the maximum correcting capability for decoder (section 2).

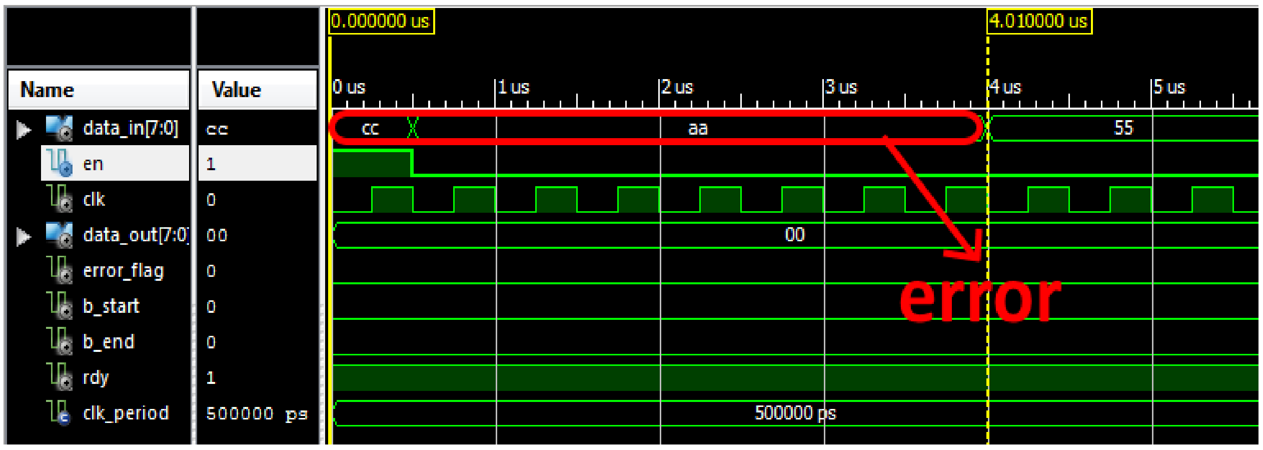


Fig. 13.1: The Decoder Waveform with 8 Bytes Error Showing that Error Occurred Attransmission.

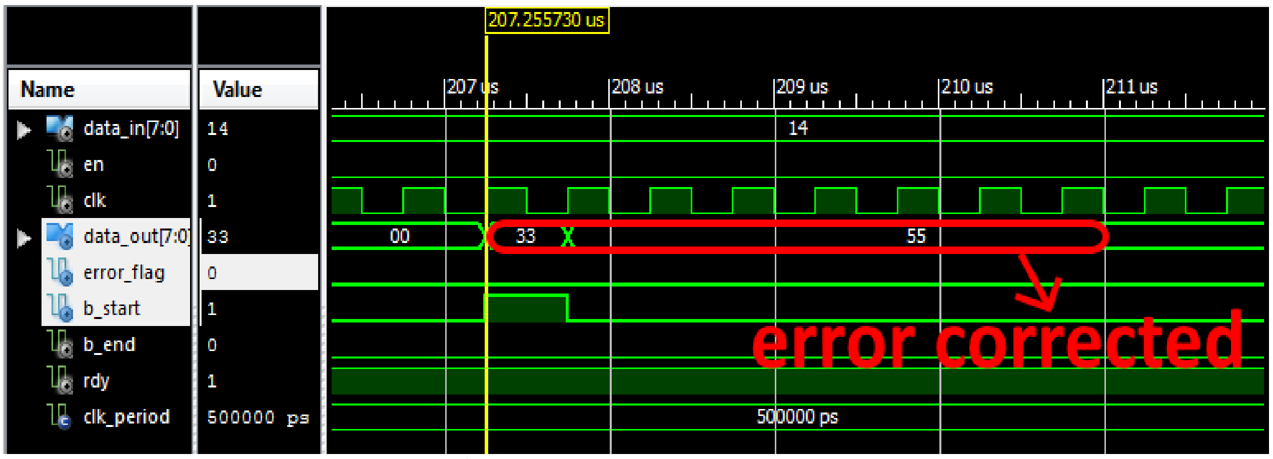


Fig. 13.2: The Decoder Waveform with 8 Bytes Error Showing the Error Correcting by Thedecoder.

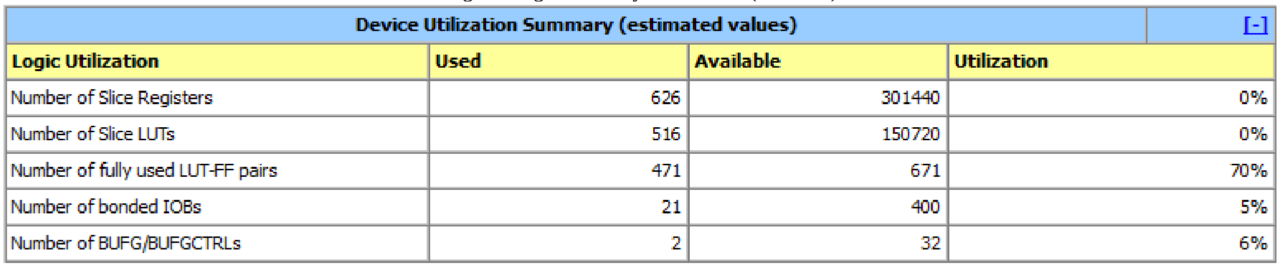
Fig. 13.2 shows the out put for decoder which shows that the error has been corrected and the original data has been restored.

5.2.5. Logic design summary

One of the most important characterizations of the design is a logic design summary which indicates the utilization of the available logic block in the FPGA chip. The Reed-Solomon codec for RS (204,188) 188 and interleave for DVB-T are implemented in VHDL to encode and decode symbols for reliable communication. The logic design summary for our RS encoder is listed in the table (1), while that of IP core R-S encoder is listed in the table (2) for the sake of comparison. The logic design summary of the overall outer coded system is listed in Table (3) it must be mentioned that all designs are implemented on Virtex 6 (XC6VLX240T).
From Table (1) clearly the implementation for our encoder consumes as m all fraction of all resources of vertix 6 FPGA chip, except the lookup tables. The encoder utilizes about 70% from the

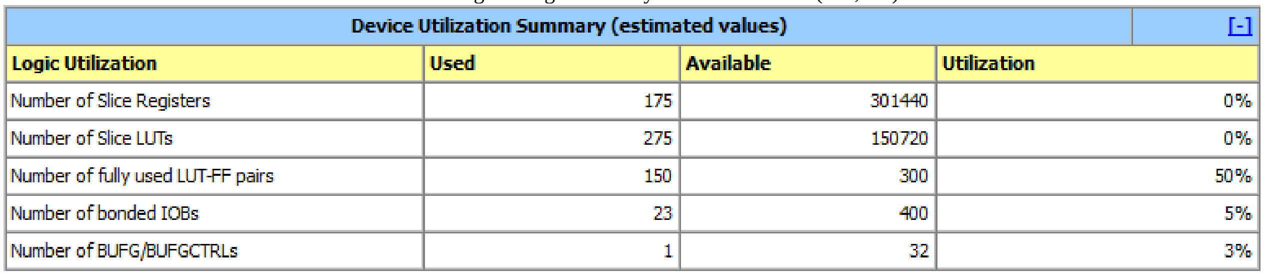
look-up table resources. There are two ways for FBGA to implement any logic block, either by using look-up table or by using functional blocks. The easy way is to use look-up tables, and so synthesis on FBGA starts using this way if the resources are enough. From Table (2), the IP core uses only 50% from look-up table resources. As the encoder alone uses the look-up table resource, so when implementing all blocks of the outer codec simultaneously uses more the second way for synthesizing the design, to reduce the usage for look up tables. So, from Table (3), we find that for the overall system, the percentage for using the look-up tables is less than that for encoder alone. One notice also from the given design summary that our design consumes more hardware than the IP core design.

Table 1: Logic Design Summary for Our RS (204, 188) Encoder.



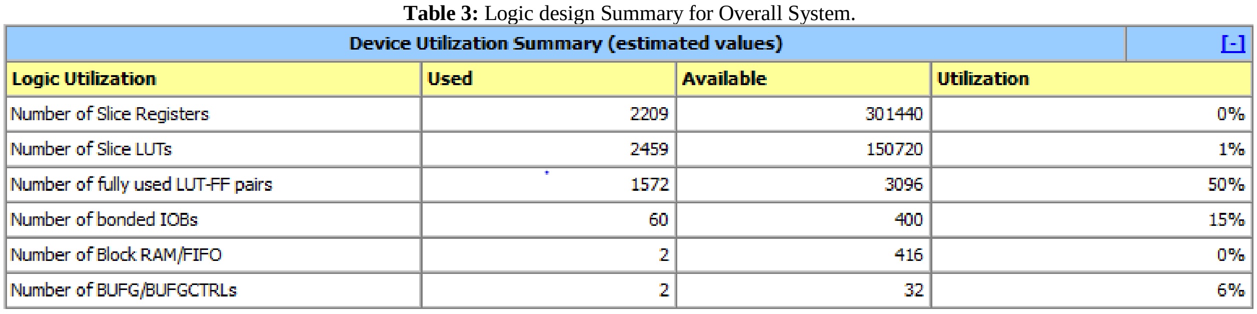
Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	626	301440	0%
Number of Slice LUTs	516	150720	0%
Number of fully used LUT-FF pairs	471	671	70%
Number of bonded IOBs	21	400	5%
Number of BUFG/BUFGCTRLs	2	32	6%

Table 2: Logic Design Summary for IP Core of RS (204,188)



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	175	301440	0%
Number of Slice LUTs	275	150720	0%
Number of fully used LUT-FF pairs	150	300	50%
Number of bonded IOBs	23	400	5%
Number of BUFG/BUFGCTRLs	1	32	3%

Table 3: Logic design Summary for Overall System.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	2209	301440	0%
Number of Slice LUTs	2459	150720	1%
Number of fully used LUT-FF pairs	1572	3096	50%
Number of bonded IOBs	60	400	15%
Number of Block RAM/FIFO	2	416	0%
Number of BUFG/BUFGCTRLs	2	32	6%

5.2.6. Timing summary

The timing report containing the timing summary of the design completes the characterization of the logic design. Here, the outer codec of a DVB-T system consisting of Reed-Solomon codec for RS (204,188) and interleaver for DVB-T are implemented in VHDL. The timing summary for our RS encoder, IP core RS encoder,synthisized on Virtex 6 (XC6VLX240T),are shown in Table (4). It is clear that our RS encoder has appreciably higher speed than that of the IP core RS encoder. This is an expected result since the higher speed of our design is on the cost of more hardware.

Table 4: Timing Summary for Our RS Encoder, IP Core Rs Encoder, and the Overall System

Time summary	Our RS encoder	RS IP core encoder	Overall system
Speed grade	-3	-3	-3
Minimum period	1.234ns	1.905ns	2.780ns
Maximum frequency	810.307MHZ	524.990MHZ	359.751MHz

6. Conclusion

Reed-Solomon codes and convolutional interleave are used for error detection and correction for reliable DVB-T. The encoder divides the incoming data stream into blocks and processes each block individually by adding redundancy. And the decoder processes each block individually and it corrects errors introduced in the received data. Interleaver spreads out the burst errors in time and deinterleaver rearrange the data thus to be handled by the decoder. These functions are implemented using VHDL code and simulated on Xilinx ISE 12.3. Then the code is synthesized on Virtex 6 (XC6VLX240T) FPGA. The design is verified by the loop back technique where it is found that the data output from the decoding processes identical to that input to the encoder. In addition, RS (204, 188) encoder simulation results are compared is with IP core RS (204, 188) which already implemented on Xilinx. It is found that both designs give the same functional performance results which validate our design. While our design consumes the larger area from the chip, it is faster than the IP core from Xilinx.
This work shows that the other building blocks of the DVB-T system can be implemented in a similar way done here for the outer codec. In this, way can construct the physical layer of the DVB-T system, and this really is one of the software radios defined radio implementations.

References

[1] O'leary, S. (2000). Understanding digital terrestrial broadcasting. (p. 1:14) London, Boston: Artech House.

[2] Sandeep Kaurs "VHDL implementation of Reed – Solomon codes" Thesis of master of Engineering in electronics and communication engineering, Thapar institute of engineering & technology, Deemed university, PatialaA – 147004, 2006.

[3] Priyanka Dayal, R. K. (2014). "Implementation of Reed-Solomon codec for IEEE 802.16 network using VHDL code". (p. 452:455). India: International Conference on Reliability, Optimization and Information Technology.
[4] Bhawna Tiwari, RajeshMehra, “FPGAImplementation of RS Codec (Dayal, "Implementation of Reed Solomon CODEC for IEEE 802.16 network using VHDL code, 2014)for Digital Video Broadcasting”, VSRD-IJEECE Journal,Vol. 2 (2), 68-77, 2012.
[5] Lamia, M.Fouraty, “A reconfigurable FEC system based on Reed Solomon codec for DVB and 802.16 network” Electronic and information technology laboratory (L.E.T.I), Sfax national engineering school, 3038 Sfax Tunis.
[6] Sklar, B., Digital Communications: Fundamentals and Applications, (p 437:468) Second Edition (Upper Saddle River, NJ) Prentice-Hall, 2001).
[7] Haoyi Zhang “Reed-Solomon code (204, 188) encoder/decoder design, synthesis and simulation with QUARTUS ⅱ”A graduate project submitted in partial fulfillment of the requirements For the degree of Master of Science in Electrical Engineering, December, 2013.
[8] L.H. Charles Lee “Error control block codes for communication engineer”. (p. 120:163) London, Boston, Artech House
[9] Aqib. Al Azad, Minhazul. Huq, Iqbalur, Rahman Rokon, “Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog”, International Conference on Advancements in Electronics and Power Engineering (ICAEPE'2011), Dec., 2011.
[10] R.T. Chien, Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghem Codes, IEEE Transactions on Information Theory, Vol. 10, No. 10, 357-363, 1964. https://doi.org/10.1109/TIT.1964.1053699.
[11] Shu. Lin., Danial J. Costello, Jr. “Error control coding: Fundamental and application” (p.151:174), Prentce- Hall, Inc. Engelwood Cliffs, New Jersey 07632.
[12] G.D. Forney, Generalized Minimum Distance Decoding, IEEE Transactions on Information Theory, Vol. IT- 12, No. 2, 125-131.
[13] G.D. Forney, On Decoding BCH Codes, IEEE Transactions on Information Theory, 549-557.
[14] ETSI EN300744“DigitalVideoBroadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television”. Final draft, V1.5.1 (2004-06), 1-64.