HIGH SPEED LOW COMPLEXITY XOR FREE TECHINQUE BASED DATA ENCODER ARCHITECTURE

[(1)]S.DIVYA BALA, [(2)]S.M.BANUPPRIYA
UG Scholar
Department of Electronics and Communication
Thiagarajar College of Engineering
Madurai.

[(3)] K.KALYANI, [(4)] S.RAJARAM
Assistant Professor, Associate Professor
Department of Electronics and Communication
Thiagarajar College Of Engineering

Abstract— In telecommunication, Long-Term Evolution (LTE) is a standard for high-speed wireless communication for mobile phones and data terminals, based on the GSM/EDGE technologies. Convolution codes find their extensive usage as channel codes in popular wireless standards like 3GPP-WCDMA, 3GPP2-CDMA2000, LTE. The existing convolutional encoder based on XOR increases circuit hardware complexity and delay time which leads to poor performance . This paper presents a new XOR-Free architecture based area efficient convolutional encoder for 3GPP2 and LTE wireless standards. The proposed approach completely removes the XOR-processing operation of a chosen non-systematic, feed-forward generator polynomial and reduces the logical operators in the encoder. The proposed design uses Read-only memory (ROM) with a pre-processed addressing operations to reduce ROM size nearly by 50%. Thus results of the new architecture reduces the area complexity and encode processing time as compared to conventional method. The VHDL coding for the proposed encoder is written, simulated and synthesized using Quartus II version 10 tool. The Hardware co-simulation of the architecture is first validated and then implemented with Altera CYCLONE II FPGA.

Keywords—3GPP2, LTE, ROM, CONVOLUTIONAL ENCODER, FPGA.

I. INTRODUCTION

Convolution codes find their extensive usage as channel codes in popular wireless standards like 3GPP-WCDMA, 3GPP2-CDMA2000, LTE, and IEEE-802.11 and software defined radio and cognitive radio applications. Convolutional codes are usually preferred over Block codes due to its economical soft decoding capability and inherent higher coding gain[1-2]. The optimization of XOR operation in convolution encoder is the main concern while implementing polynomials over Galoi field. The existing convolutional encoder based on XOR increases circuit hardware complexity and delay time which leads to poor performance [3-5]. Existing convolutional cannot be reconfigured by modifying polynomial of same constraint length and code rate [6-9].

Thus the objective of this work is to design an XOR free convolution encoder architecture using Hardware Description Language (HDL). Then modify the same architecture to reduce the area complexity and encode processing time by making use of memory sharing ROM architecture using Hardware Description Language (HDL). To simulate these architectures using Quartus II version 10 tool and implement on Altera CYCLONE II FPGA. The hardware complexity and encoding time for this proposed XOR free encoder is compared with conventional encoder to show the efficiency of proposed work.

The paper is organized into five sections, section 1 deals with the introduction, section 2 deals with the conventional convolutional encoder. Section 3 covers in depth with the XOR free convolutional encoder. Section 4 states the result and discussion, whereas Section 5 provides a conclusion

II. CONVENTIONAL CONVOLUTIONAL ENCODER

2.1 CONVENTIONAL XOR BASED CONVOLUTIONAL ENCODER

A convolutional encoder is a discrete linear time-invariant system, used to encode the data prior to transmission, over a channel. The convolutional encoder is realized with a shift register (SR) using delay elements and modulo-2 adders (XOR gates). In the convolutional process, the main operation is

multiplication, which is implemented using shifts and adders. The multiple addition operation increases complexity and consumes a significant amount of power. Hence the mitigation of such complexities by reducing the number of logical operators is the prime focus in the encoder design.

Convolutional codes are often characterized by the base code rate and depth (or memory) of the encoder[n,k,K].The basic code rate is typically given as n/k, where n is the input data rate and k is the output symbol rate. The depth is often called the "constraint length" k, where the output is a function of the current input as well as the previous K -1 inputs.

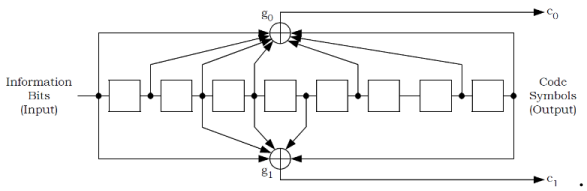

Fig 1

Fig 1 represents the convolution operation it is realized using a deterministic finite state machine (DFSM). Its hardware implementation requires a combinational circuit and memory elements.

This paper describes a new algorithm to implement a convolutional encoder of a chosen generator polynomial having constraint length (K<a) for popular wireless standards. The approach reduces the standard polynomial into LUT. Convolutional operation is realized using a deterministic finite state machine(DFSM).

This existing convolutional encoder based on XOR increases circuit hardware complexity and delay time which leads to poor performance . This paper presents a new XOR-Free architecture based area efficient convolutional encoder for 3GPP2 and LTE wireless standards. The proposed approach completely removes the XOR-processing operation of a chosen non-systematic, feed-forward generator polynomial and reduces the logical operators in the encoder as explained below.

III.XOR FREE CONVOLUTIONAL ENCODER

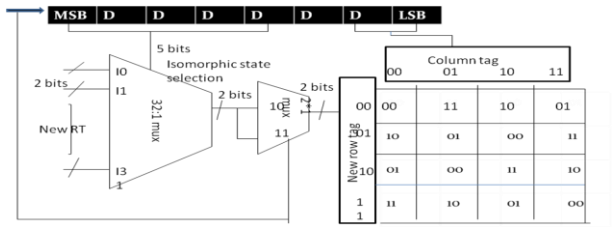

Fig 2

XOR free encoder process is a new way of data encoding and decoding architecture. This methodology is mainly used for error correcting process, by designing a encoder architecture based on parallel form and to reduce the hardware complexity level. XOR free encoder is to minimize logical gate count level and to improve encode processing time.

The existing XOR free convolution encoder architecture includes shift registers with K-1 memory elements and multiplexers to decode the encoded sequences from the ROM. It has 8 input bits, and 5 bit for selection line. It makes use 4*4 memory allocation process**.** The code assumes an input bit as static logic "0," for the possible combinations of shift register state and convolve them to compute the encoded output. Here a 32:1 MUX is used to select new RT from one of its inputs. In the second stage a 2:1 MUX is used to process the output of the first MUX in accordance with the input as shown in figure 1.8.In this architecture 4*4 ROM memory is used. This 4*4 ROM increases the hardware complexity of the existing XOR Free convolutional architecture.

3.1 PROCESS OF EXISTING XOR FREE CONVOLUTIONAL ENCODER

The process of existing XOR free convolution encoder architecture consists of following steps. They are 1.Selection of MUX logic 2.XOR free MUX controller 3.Data compression ROM function 4.Register allocation

3.1.1 SELECTION MUX LOGIC

Assign the information bit as logic "0", for all possible encoder states (S) whose next state follows previous states incremented by one. Compute the output response using basic convolutional encoder.

3.1.2 XOR FREE MUX CONTROLLER

Group the encoder states based on their similar encoded output with possible values as {00, 01, 10, and 11}. Splits the encoder state assignment bits into two subparts. The first is a row tag (RT) as most significant bits and column tag (CT) as the least significant bits.

3.1.3 DATA COMPRESSION ROM FUNCTION

The convolved outputs are stored in ROM architecture. It makes use of 4*4 memory allocation process**.** The addresses of ROM are RT and CT, formed by current input bit and the previous state of shift register. The outputs of the ROM are the encoded sequences which can be taken in serial or parallel sequence as per the requirements.

3.1.4 REGISTER ALLOCATION

The Register is used to store the updated data accumulated in every clock cycle. The Convolved outputs are stored in corresponding memory registers. The above process increase the encoding processing time of this existing XOR . Thus existing architecture increases hardware complexity and

encode processing time. This can be reduced in the present work explained below.

IV. REDUCED ROM BASED XOR FREE CONVOLUTIONAL ENCODER ARCHITECTURE

The modified architecture uses 8*1 MUX at the final stage. This ROM based architecture can be reconfigured by modifying polynomial of same constraint length and code rate. Reduced ROM is more advantageous than previous architecture that it consumes less number of LUT's ,flip flop, flip flop LUT pairs, BUFG.

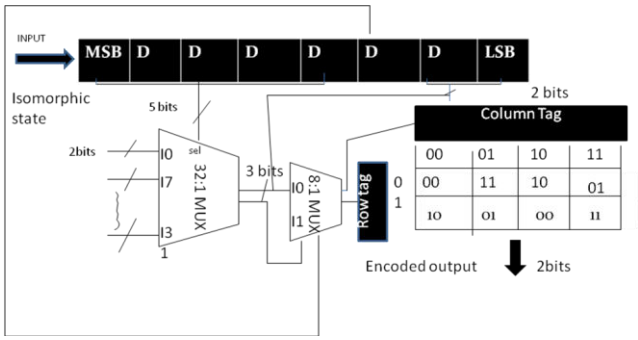

Fig 3

This proposed design uses Read-only memory (ROM) with a pre-processed addressing operations reduce ROM size nearly by 50%.

4.1 DESIGN FLOW OF REDUCED ROM ARCHITECTURE

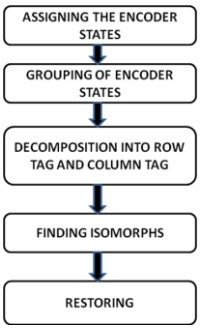

Fig 4

4.1.1 Assigning of encoder states

Assign the information bit as logic '0', for all possible encoder states (S), whose next state follows previous state incremented by one. Compute the output response using basic convolution process. The encoded output with its encoding states is then arranged in a tabular form.

4.1.2 Grouping of encoder states

The encoded output states are grouped based on their similarity.

4.1.3 Decomposition

The encoded assignment states are splitted into two sub parts. The First is a Row Tag (RT) as most significant bits and Column Tag (CT) as the least significant bits.

4.1.4 Finding Isomorphs

The row tag pairs which has the same output of column tag pair are merged to create array(ROM) to create isomorphs.

4.1.5 Restoring

To regain the full functionality for any input bit, i.e. Logic '0' as well as logic '1', the even and odd concept is used output is flipped

V. EXPERIMENT AND RESULT ANALYSIS

The proposed architecture was designed in VHDL and synthesized by Altera Quartus II version 10 tool and implemented in Altera Cyclone II FPGA Board using VHDL code.

5.1. Simulation results

a) Simulation Result for Selection of MUX logic

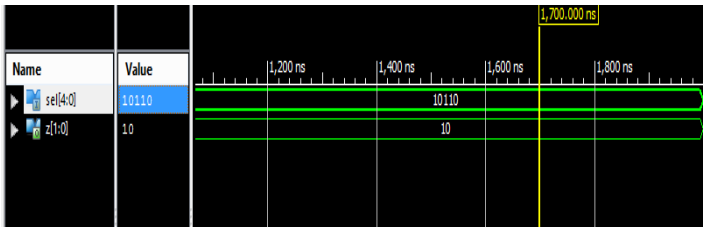

Fig 5 simulation result for selection of MUX logic

The selection input for multiplexer logic is 10110.The output of multiplexer is represented by z in figure 5. Figure 6 represents the output of 8:1 multiplexer. Figure 7 denotes the output of RAM memory allocation .Figure 8 is the overall output of the XOR free convolutional encoder. The input for XOR free mux is select input=10110 and the output is z=10

b) Simulation result for XOR free MUX controller

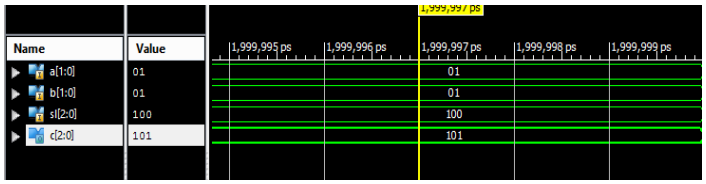

Fig 6:Simulation result for XOR free MUX

c) Simulation result for Data compression ROM function

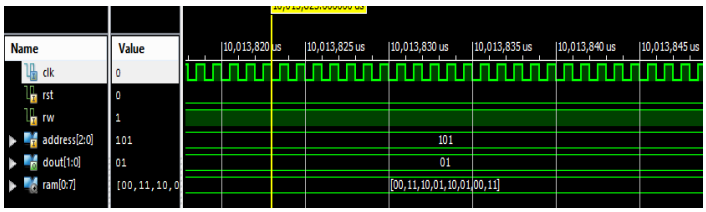

Fig 7 : Simulation result for RAM Memory allocation

The input is the address=101 and the output is dout=01

d) Simulation result for XOR free convolution encoder

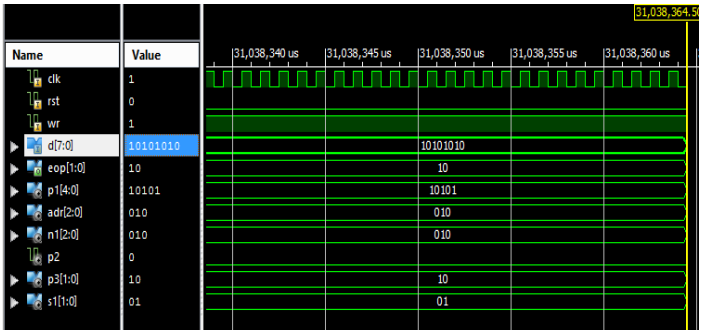

Fig 8:Simulation result for XOR free convolutional encoder

The input for ROM architecture is d=10101010 and the output is eop=10

e) RTL view of XOR free convolutional encoder

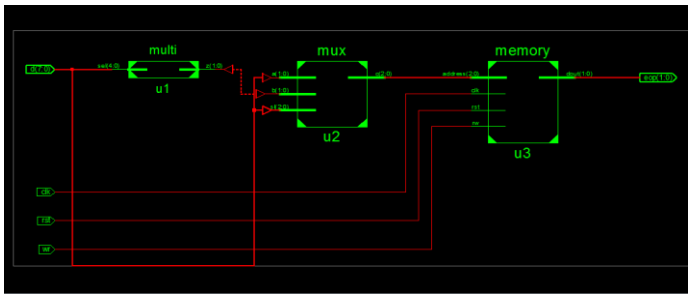

Fig 9: RTL View of XOR free convolutional encoder

The 1st block represents the 32*1 mux and the 2nd block represents the 8*1 mux and the 3rd block represents the Rom architecture

f) Technology diagram of XOR free convolutional encoder

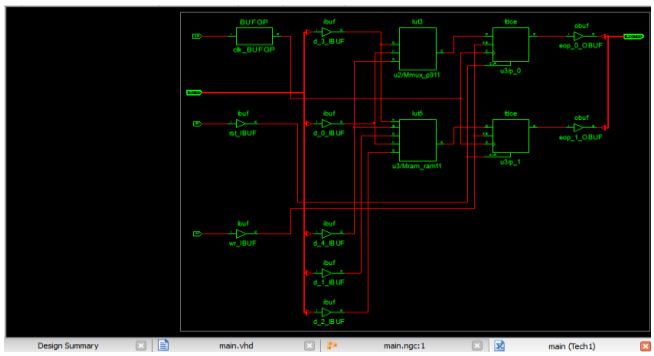

Fig 10 Technology Diagram for XOR free Convolutional Encoder

The blocks in the diagram represents the BUFG(Global Buffer) and the LUT's used in the architecture.

5.2 Synthesis Result

The proposed method is synthesized and implemented using Quartus II version 10 tool and Altera DE1 CYCLONE II FPGA BOARD as a target device to compute the area complexity and computational time.

5.2.1 Area Analysis

The synthesis report obtained is used to compare the efficiency of proposed method with the conventional method. Table 1 shows the comparison of logic utilization of the proposed method with the conventional one. It is evident that the proposed method reduces the hardware resources.

Table-1

LOGIC RESOURCES	AVAILABLE RESOURCES	EXISTING WORK	PROPOSED WORK	REDUCTION PERCENTAGE
FFs	19,000	12	02	83.3%
LUTs	63,400	03	02	33.33%
LUT FF pairs	15,580	13	10	23.07%

5.2.2 Time Analysis

The Frequency of operation and time required for Encoding Process of the existing and proposed convolutional encoder are given by:

Table-2

PARAMETER	EXIXTING WORK	PROPOSED WORK	PERCENTAGE
FREQUENCY(MHZ)	444.6	1261	64.71%
TIME(ms)	22.4	0.7	96.8%

V. CONCULUSION

Convolutional encoder plays a vital role in the field of telecommunication. FPGA based XOR free convolutional encoder is done in this project. It consists of three blocks selection of MUX logic, XOR free MUX and ROM memory. VHDL coding for MUX logic, XOR free MUX and ROM memory are written and simulated using Quartus (II) software. Then these blocks are implemented by downloading those codes in ALTERA FPGA DEI EP2C2C70F484 (CYCLONE II) board. Thus the proposed work reduces the number of flip flops by 16.6%, LUT by 66.6% and LUT FF pairs by 76.9% compared to existing and the speed of the proposed work increases by 3.125%.

References

[1] E. Arikan, "Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memory less channels,"IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] R. Mori and T. Tanaka, "Performance of XOR Free Encoders with the construction using density evolution,"IEEE Commun. Lett., vol. 13, no. 7, pp. 519– 521, Jul. 2009.

[3] S. B. Korada, E. Sasoglu, and R. Urbanke, "XOR Free Encoders: Characterization of 2011, pp. 1–5.

[4] K. Chen, K. Niu, and J. Lin, "Improved successive cancellation decoding of XOR Free Encoders,"IEEE Trans. Commun., vol. 61, no. 8, pp. 3100–3107,Aug. 2013.

[5] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for XOR Free Encoders," IEEE Trans. Signal Process., vol. 61, no. 2, pp. 289–299, Jan. 2013.

[6]G.Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation,"IEEE J. Sel. Areas Commun., vol. 32, no. 5, pp. 946–957, May 2014.

[7] C. Zhang and K. K. Parhi, "Latency analysis and architecture design of simplified SC polar decoders,"IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 2, pp. 115–119, Feb. 2014.

[8] M. Ayinala, M. J. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding transformation,"IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 20, no. 6, pp. 1068–1081, Jun. 2012.

[9] K. K. Parhi, "Calculation of minimum number of registers in arbitrary life time chart," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process

[10] C. Y. Wang, "MARS: A high-level synthesis tool for digital signal processing architecture design," M.S. thesis, Dept. Elect. Eng., University of Minnesota, Minneapolis, MN, USA, 1992