A Novel Performance Analysis of Multiplexer Based Convolutional Encoder

P. Mano
PG Scholar, Department of Electronics and Communications Engineering, PSNA College of Engineering and Technology, Dindigul, India.

M. Krishnamurthy
Assistant Professor, Department of Electronics and Communications Engineering, PSNA College of Engineering and Technology, Dindigul, India.

Abstract – In digital communication, convolutional codes are important to control the errors between the transmitter and receiver. Convolutional code is a type of error correction code (ECC) which can perform both error detection and correction operation between transmitter and receiver side. Here information bits are transformed serially through the architecture which is one of the major advantages as compared to block code method. The convolution encoder uses XOR gate which acts as main element in the convolution process. Convolution code which is performed based on the XOR operation has a main drawback of consuming high power initially. So it becomes necessary to replace the XOR operation with a low power consumption component mainly in low power applications. MUX acts a good alternative to XOR which consumes less power as compared to the XOR gate. In this work, two different type MUXs are involved to perform convolutional encoder operation for the encoding process. From these proposed approach can be reduce the power consumption, circuit complexity and also reduce the read only memory (ROM) size nearly 50%.

Index Terms – Forward Error Correction (FEC), Convolutional Code(CC), Common Sub Expression (CSE) and Error Correction Code(ECC).

1. INTRODUCTION

In communication, a convolutional code is a type of error-correcting and detection code that generates parity symbols via the sliding application of a polynomial function to a data stream. The sliding application represents the 'convolution' of the encoder over the data, which gives rise to the term 'convolutional coding.' Time invariant trellis decoding allows convolutional codes to be maximum-likelihood soft-decision decoded with reasonable complexity. The ability to perform economical high likelihood smooth decision decoding is one of the major advantage of convolutional codes. This is in contrast to classic block codes, which are generally represented by a time-variant trellis and therefore are typically hard-decision decoded. Convolutional code consists of three parameters.

Convolutional codes are mainly characterized by the proper base code rate and the depth of the code or memory of the encoder [n,k,K]. The base 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 mainly known as constraint length (K), thus the output of the process is a function of the present input as well as the previous K-1 inputs. The depth may also be given as the number of memory elements 'v' in the polynomial or the maximum possible number of states of the encoder (typically 2^v).

The base code rate of a convolutional code is generally modified via symbol puncturing. For example, a convolutional code with a 'mother' code rate n/k=1/2 may be punctured to a higher rate of, for example, 7/8 simply by not transmitting a portion of code symbols. The performance of a punctured convolutional code generally scales well with the amount of parity transmitted. The ability to perform economical soft decision decoding on convolutional codes, as well as the block length and code rate flexibility of convolutional codes, makes them very popular for digital communications.

The encoder operates on the incoming message sequence continuously in serial manner [9].The buffer is not needed. The code words generated are non-systematic codes. In convolutional encoder mainly depends on XOR operation [15] which is the main operation for the encoder process. It consumes more dynamic power through entire operation of convolution process[7] & [8] and then using of common sub expressions element (CSE) method the same bit output data produce for the given different inputs can be identified and then it will be eliminated [13] & [10].

2. DESIGN AND IMPLEMENTATION

Convolution operation is realized using a deterministic finite state machine (DFSM). Its hardware implementation requires a combinational and memory device. The discrete convolution for the encoded sequence (CJ) can be expressed in terms of information sequence (I_I) with the generator sequences (G_I) by the following equation [4].

$$C_J=\sum_{i=0}^{M} Ij - i\ Gi \quad \text{..... (1)}$$

Further shift register (SR) based realization of (1) for encoded sequence (C_J) depends upon the length (L) of SR, the present input I_J and M previous input blocks[I_J-1,……,I_{J-M}] to yield(2)

$$C_J=\sum_{\propto=1}^{n} \quad [\sum_{I=0}^{M} Ij - I \propto g\alpha\beta] \ldots\ldots\ldots\ldots (2)$$

The equation (2) can be used to realize a CDMA2000 Convolutional Encoder with g0(x) = (753) and g1(x) = (561) as generator polynomials, having a code rate k=(1/2) and constraint length K=9. The conventional convolutional encoder for CDMA-2000 shown in a figure 2 [3].

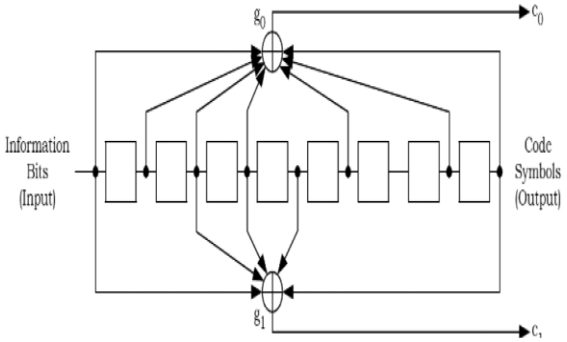

Figure 1 Conventional Diagram

The implementation of convolution encoder depends on following few processes.

1. Assignment
2. Grouping
3. Decomposition and
4. Restoring

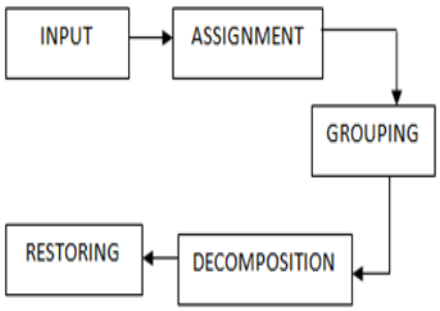

Figure2 Block Diagram of Convolution Encoder Architecture

From the above block diagram the input data can be given to the architecture. Then the data can be performing the basic convolution operation. After that process the similar output data are grouped. From the decomposition process to separate the input data into two subparts (i.e.) row tag (RT) and column tag (CT). After that process the data can be restored and forming a new RT input. With help of assignment a decomposition process the table1 can be formed.

Table 1 Convolution Output

RT \ CT →	00	01	10	11
000000	00	11	10	01
000001	00	11	10	01
000010	01	10	11	00
000011	01	10	11	00
⋮⋮⋮	⋮	⋮	⋮	⋮
111111	01	10	11	00

Table 2 Isomorph States for RT

New RT	Isomorph states (in Decimal)
00	0,1,14,15,22,23,24,25,34,35,44,45,52,51,58,59
01	2,3,12,13,20,21,26,27,32,33,46,47,54,55,56,57
10	4,5,10,11,18,16,28,29,38,39,40,41,48,49,62,63
11	6, 7, 8, 9,16,17,30,31,36,37,42,43,50,51,60,61

After the decomposition process the output data's are grouping based on their isomorph states. Isomorph is nothing but the similar output will produce for given different inputs. Then it form like table2. Finally to find the new RT input, it will be applied to the multiplexer. The new RT table 3 shown [3].

Table3 New RT based on Isomorph

New RT ↓ \ CT →	00	01	10	11
00	00	11	10	01
01	10	01	00	11
10	01	10	11	00
11	10	01	00	11

From these four operations, to find a new row tag (RT), this will be applied to the 32:1 MUX. From the given 8-bit input data the most significant bit (MSB) will be selected and performing the operation. Then the output will flows through 2:1 MUX, depends on the control line it will produce the

output. After that the output data will be stored their corresponding address location in ROM.

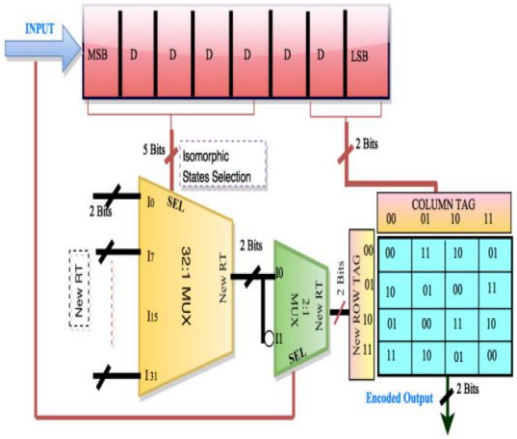

Figure3 Reduced ROM Encoder Architecture

Then the encoder architecture has been further to realize a CDMA2000 Convolutional Encoder with g0(x) = (753) and g1(x) = (561) as generator polynomials, having a code rate k= (1/2) and constraint length K=9 with modifying the 2:1 MUX into 8:1 MUX. In that operation the given 8- bits input data, to use 5 bits (D7 to D3) as the selection for the 32:1 MUX. Based on the selection line it will select one of the New RT Inputs and to produce an output value (2 bit). Then the con-coordinate operation can be performed between 32:1 MUX output data and given 8-bit input data. That operation can be used to form eight input value. Let us consider a= 32:1 MUX output (2bit) and b= D1D0 value from the 8-bit input data. The output of 8:1MUX is 2 bit since, code rate r= (1/2). So, have to perform a Con-coordinate operation. Con-coordination is the process of joining the two bit values.

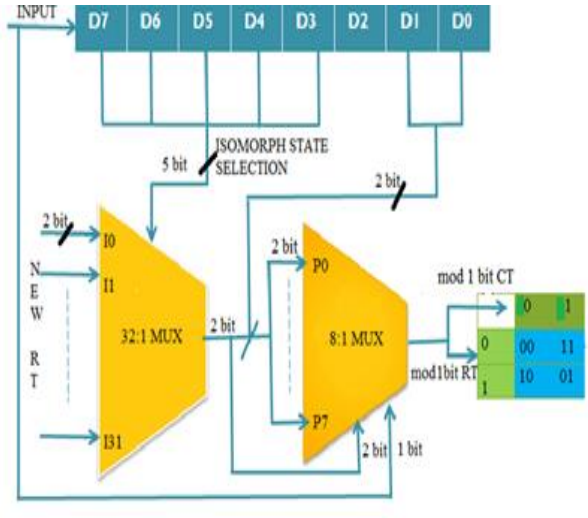

Figure4 MUX Based Convolutional Encoder Architecture

From the above eight con-coordinate equations used to frame the eight (2bit) input. This input 8 input given to the MUX then the output will produce depend on the 3 bit selection line input data, after that the output values are already stored in certain memory location. In that process, to create an array of memory location for the given input and to find the equal memory location of the input bit.

3. RESULT

The MUX based convolutional encoder architecture is implemented on Xilinx using VHDL code. This architecture encoded at a generator polynomial, code rate r=1/2 and constraint length K=9. The comparison of various parameters like number of IOB's, LUT's and frequencies shown in table 4.

Table 4 Comparison Table

PARAMETER	REDUCED ROM XOR-FREE ARCHITECTURE	MUX BASED ENCODER ARCHITECTURE
IOB	11	10
LUTs	03	02
BUFG/BUFGCTRLs	01	01
F(MHz)	585.5	1517.45
T(ns)	1.707	0.659

The Xilinx XPower analyzer tool used to find the dynamic power consumption of the MUX based convolutional encoder architecture at a clocking rate of 1000ns. The overall parameter comparison graph and power consumption shown in figure 5 & 6.

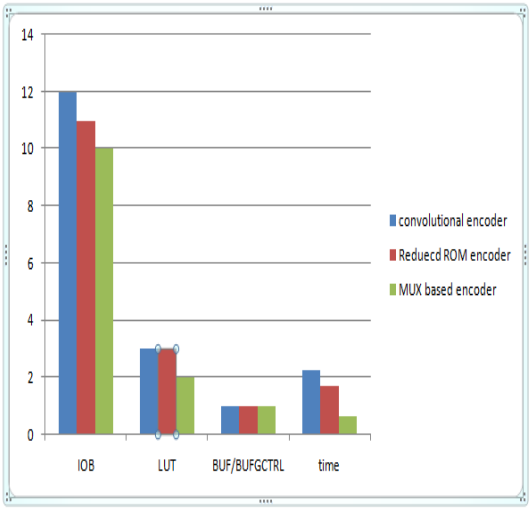

Figure5 Comparison Graph

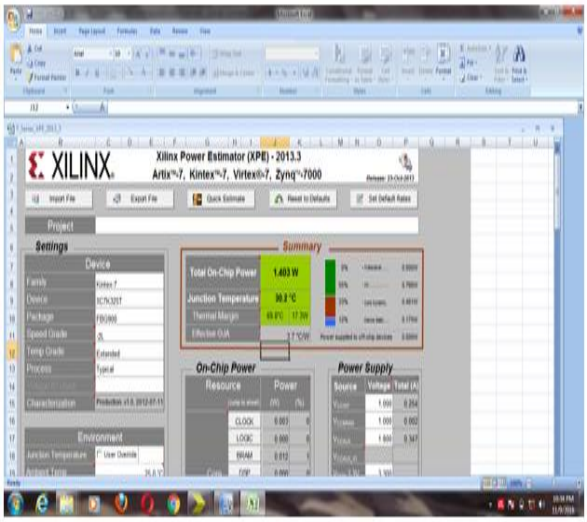

Figure6 Power Calculation

4. CONCLUSION

The result shows the MUX based encoder architecture using common sub-expressions method and parallel processing technique. This technique helps to reduce power consumption and the circuit complexity for data transmission and reception process compare to XOR free architecture. This brief has presented a new partially parallel encoder architecture which is developed for a long XOR Free Encoders. Many optimizations techniques have been applied to derive the proposed architecture. Therefore, the proposed architecture provides a practical solution for encoding a long MUX based Encoder. Finally to get the encoder output data bits with reduce the ROM size nearly 50% using MUX based architecture.

REFERENCES

[1] J. Viterbi, "Convolutional codes and their performance in communication systems," IEEE Transactions on CommunicationTechnology, vol. 19, no. 5, pp. 751–772, 1971.

[2] Y. Yibin, K. Roy, and R. Drechsler, "Power consumption in XOR-based circuits," in Proceedings of the Asia and South PacificDesign Automation Conference (ASP-DAC '99), vol. 1, pp. 299– 302, IEEE,Wanchai, Hong Kong, January 1999.

[3] G. Purohit, Member, K. S. Raju, And V. K. Chaubey, " A New Xor-Free Approach For Implementation Of Convolutional Encoder" IEEE Embedded Systems Letters, Vol. 8, No. 1,Pp22-25, March 2016.

[4] Gaurav Purohit, Kota Solomon Raju and Vinod Kumar Chaubey, "XOR-free implementation of convolutional encoder for reconfigurable hardware", International Journal of Reconfigurable Computing, vol. 2016, Article ID 9128683, 8 pages, 2016.

[5] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions," IEEE Transactions on Computer-Aided Design ofIntegrated Circuits and Systems, vol. 18, no. 1, pp. 58–68, 1999.

[6] J. Dielissen, N. Engin, S. Sawitzki, and K. van Berkel, "Multistandard FEC decoders for wireless devices," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 55, no. 3, pp. 284–288, 2008..

[7] R. Arunajayashree, S. Ramyashri, S. Selvarani and M. Sharmila, "Reconfiguration of Convolutional Encoder by optimizing the XOR"International Conference on Explorations and Innovations in Engineering & Technology (ICEIET - 2016).

[8] G. P. Calzolari, M. Chiani, F. Chiaraluce, R. Garello, and E. Paolini, "Channel coding for future space missions: new requirements and trends," Proceedings of the IEEE, vol. 95, no. 11, pp. 2157–2170, 2007

[9] Huang, J. Li, and M. Chen,"Optimizing XOR-based codes," US Patent 8209577 B2, 2012

[10] S. Lin and D. J. Costello Jr., Error Control Coding, Prentice-Hall, Englewood Cliffs, NJ, USA, 2nd edition, 2004.

[11] M. Igor, "Power consumption analysis of XOR based circuits," Informatics, vol. 1, no. 9, pp. 97–103, 2006.

[12] T. Stock hammer, A. Shokrollahi, M. Watson, M. Luby, and T. Gasiba, Application Layer Forward Error Correction for MobileMultimedia Broadcasting Case Study, Digital Fountain, 2009.

[13] J. Rabaey, Low Power Design Essentials, Springer Science & Business Media, New York, NY, USA, 1st edition, 2009.

[14] S. Benedetto and G. Montorsi,"Role of recursive convolutional codes in turbo codes," Electronics Letters, vol. 31, no. 11, pp. 858– 859, 1995.