Decimal Convolutional Code and its Decoder for Low-Power Applications

Ali Ghasemi Khah [1*], Yosef Seifi Kavian [2]
1- Department of Electrical Engineering, Shahid Chamran University, Ahvaz, Iran.
Email: ali_ghasemikhah@yahoo.com (Corresponding author)
2- Department of Electrical Engineering, Shahid Chamran University, Ahvaz, Iran.
Email: y.s.kavian@scu.ac.ir

Received: August 2018 Revised: September 2018 Accepted: November 2018

ABSTRACT:
P. Elias proposed convolutional coding at 1955. Convolutional encoders have very simple structure but their decoders are very complex and power consumer. Power consumption and error correction of Convolutional Codes, will be enhanced by increase in their constrain length, therefore there is always a trade-off between Power consumption and error correction. In Convolutional Codes, the code specifications remain constant in each frame. If the specifications are changed during each frame in a code, a new code with new performance and specifications is created. This paper, aims to evaluate this issue for the first time and compare its performance with Convolutional Codes. This new code is named “Decimal Convolutional”. If in a decimal convolutional code, constrain length is changed during each frame, the generated code will be a convolutional code with “decimal constrain length”. In this paper, a convolutional code with decimal constrain length is introduced, encoder and Viterbi decoder structure is explained for it and its specification is compared with convolutional code. Using this code, an optimized constrain length can be obtained and relative power consumption of decoder can be also reduced. The proposed design blocks are described by VHDL and they are implemented on Xilinx Spartan3, Xc3s400 FPGA chip.

KEYWORDS: Decimal Convolutional Code, Viterbi Decoder, FPGA, Low Power.

1. INTRODUCTION

Many communication systems, such as mobile phones and satellite transmission systems, use algorithms to correct errors of received data, since error correction coding better utilizes the band-limited channel capacity than special modulation schemes [1]. Convolutional Codes are powerful and High performance in many communication systems. Constrain length and data rate are the important specifications of convolutional codes. By enhancing constrain length and reducing the data rate, the error correction performance will be improved, but instead the chip area and the decoder power consumption will be increased [2-4]. In convolutional coding, the code specifications won’t be changed during a data frame. According to convolutional coding, a new coding method is introduced in this paper, in which, the code constrain length varies in different stages of each frame. This new convolutional code is named “decimal constrain length convolutional code”. Among the methods that are provided for decoding the convolutional codes, “Viterbi algorithm” has a higher error correction performance. In this paper, the Viterbi decoder is designed and implemented for the mentioned code.

2. THE CONVOLUTIONAL CODE WITH DECIMAL IN CONSTRAIN LENGTH

For evaluating the convolutional code with decimal in constrain length, we examine a type of these codes that their constrain lengths will be decreased one unit in the middle of each frame. In the following part of this paper, wherever we name the decimal convolutional code, we mean this kind of these codes. In **Fig. 1**, a sample of decimal convolutional encoders is indicated, in which the outputs are determined in the first 9 stage of each frame by 3 flip flaps and then by 2 flip flaps. In the other word, in the first 9 stages, constrain length of code is equal to 4 and from the10th stage it will be decreased to 3. The 10th stage is named as “Break Stage” (BS).

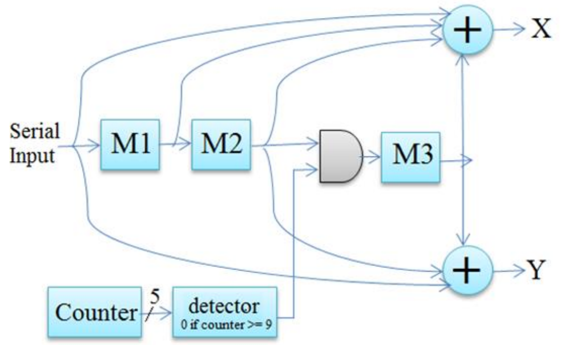

Fig. 1. Convolutional encoder with decimal in constrain length of code.

The indicated encoder in **Fig. 1** is used as sample encoder in next sections. Trellis diagram of sample encoder is indicated in **Fig. 3**. As it is indicated in trellis diagram, the number of diagram states is decreased to half due to the reduction in one of the encoder flip flaps in the BS. This leads to a reduction in chip area and Power consumption in decoder. For example, in Viterbi decoder, the calculation volume and the dynamic power of the PMU section, the number of cells in SMU section and the number of the components in TB section are decreased to half from BS. Therefore, it is expected that the chip area, the calculation volume and the dynamic power of Viterbi decoder are decreased to half from BS.

The generated code by sample encoder has no constrain length equal to 3 or equal to 4. It is in the middle of these two values and it has a decimal constrain length. Considering the 30 bits frames, the constrained length of its code is obtained from (1).

$$K = 3 + \frac{9}{30} = 3.3 \qquad (1)$$

That K is constrain length. The constrain length of decimal convolutional encoder, with a structure similar to sample encoder, is obtained from (2).

$$K = K2 + \frac{BS - 1}{L} \qquad (2)$$

That K2 is number of encoder flip-flaps after BS and L is frames length. In sample encoder, the value of m3 flip flap, from the 9th stage, has no effect in encoder output. The value of this flip flap in 9th stage is named “latent bit”, due to its performance on the code error correction performance, despite removing it from the encoder. This is evaluated in the following sections. In trellis diagram, the latent bit is the MSB bit of the encoder state in 9th stage which is indicated in **Fig. 2**.

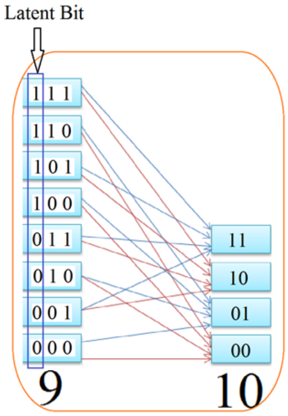

Fig. 2. Latent bit in trellis diagram.

One of other decimal coding advantage is that, due to the reduction in constrains length of code in the middle of each frame, one of the required reset bits is reduced, so we can send the information more than 1 bit in each frame and therefore, the data rate is increased.

3. THE PERFORMANCE OF CONVOLUTIONAL CODE WITH DECIMAL IN CONSTRAIN LENGTH

For evaluating the decimal convolutional code, the decoder Power consumption and error correction is assessed based on its two adjacent convolutional codes. A MATLAB code is driven to evaluate the BER for error correction performance of the decimal convolutional code, K=3.3 and its adjacent convolutional codes, K=3 and K=4. BER curve is shown in **Fig. 4** vs. SNR using AWGN channel and BPSK modulation.

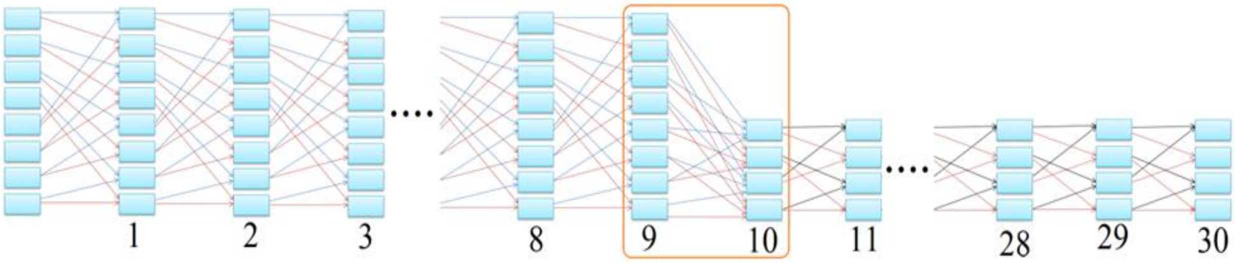

Fig. 3. The trellis diagram code with the constrain length of 3.3 for 30 bit frames.

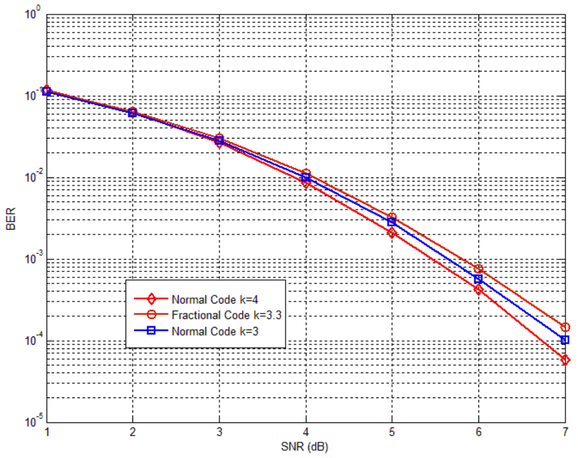

Fig. 4. BER curves for the code with constant lengths of 3, 4 and 3.3.

As it can be seen in **Fig. 4**, performance of decimal convolutional code has not an appropriate error correction, because, its performance in large SNRs is worse than the convolutional code smaller than itself (k=3). This performance is due to ignoring of latent bit in encoder side. If the latent bit is preserved and available in decoder side, the BER will fit that is shown in **Fig. 5**.

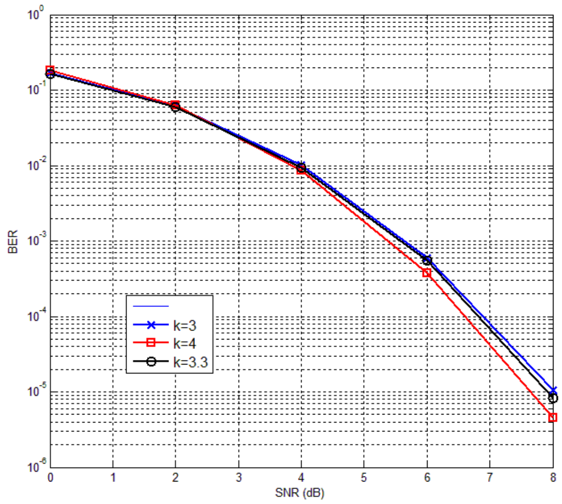

Fig. 5. Performance of decimal code error correction with preserving latent bit.

Fig. 5 indicates that, if the latent bit is completely preserved, the convolutional code with decimal in constrain length will have a good performance. The **Fig. 5** diagram was for preserving the latent bit, but in practice this bit cannot be preserved completely due to the channel noise. There can many methods provided for relative preserve of latent bit. The simple method iterative coding was used in this paper, in order to preserve the latent code. In **Fig. 6**, the convolutional code performance with decimal constrain length of 3.3, with iterative coding 4 for preserving latent bit is indicated with performance of convolutional codes which indicate the improve in decimal code performance.

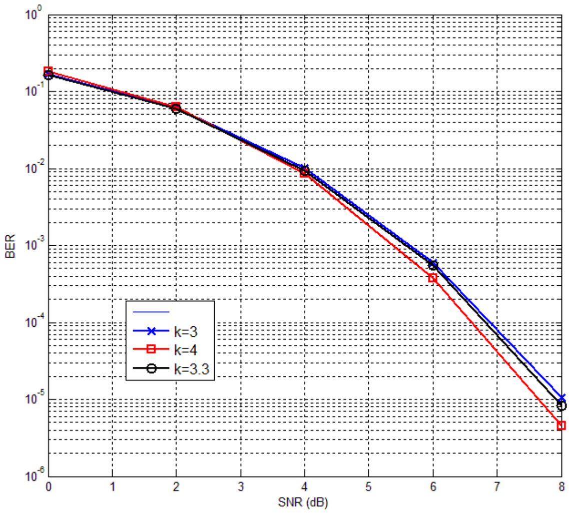

Fig. 6. Convolutional code error correction performance after preserving the latent bit by iterative coding 4.

The result of simulating for recent method in SNR=4dB point is in **Table 1**. According to **Table 1** , (3) and (4), code with constrain length of 3.3 to the code with constrain length of 3, has 44% better relative performance and to the code with constrain code of 4, has 56% weaker relative performance.

Table 1. BER for decimal convolutional code and its adjacent convolutional codes in SNR = 4dB

	K = 3	K = 4	K = 3.3
BER	1015×10^{-5}	858×10^{-5}	946×10^{-5}

$$relative\ improvement = \frac{1015-946}{1015-858} \times 100\% = 44\% \qquad (3)$$

$$relative\ attenuation = \frac{946-858}{1015-858} \times 100\% = 56\% \qquad (4)$$

Preserving the latent bit by iterative coding method, although improves the performance and weakens the error, due to sending extra data with main data will reduce the data rate. In fact, in this method, there is a trade-off between the error correction performance and data rate. For evaluating the idea of decimal code, decimal decoder and encoder are implemented as hardware. In the following, the method of implementing the convolutional decoder with decimal in constrain length, is explained.

4. CONVOLUTIONAL DECODER IMPLEMENTATION WITH DECIMAL IN CONSTRAIN LENGTH

In previous Section the code with decimal in constrain length, was assessed In terms of error correction function. It was also said that due to reduced calculations and decimal decoders' hardware, it is expected that their power consumption decreases, but to implement a decimal decoder, Additional hardware is required which is the consumer of power. Thus for appropriately assess the power consumption of the proposed code, a Viterbi decimal decoder and a decimal encoder, have been implemented. BMU of decimal Viterbi decoder is similar to BMU in Viterbi decoder. A description of other convolutional Viterbi decoder blocks with constrain length 3.3 is given later.

4.1. Path Metric Unit

Viterbi decoder finds the encoder path using the maximum-likelihood method. In this method the Viterbi decoder calculates the cost of all the paths, and picks the smaller path as the right path, this is conducted by PMU [5]. The PMU consists of ACS blocks, according to trellis diagram in **Fig. 3**, in the BS, the nodes of trellis diagram have 4 inputs and the rest of stages have 2 inputs. So ACS blocks of decimal Viterbi decoder, in the BS, should choose the shortest path among the 4 possible paths, which requires a structure as in **Fig. 7**. This structure is different from normal ACS structure with two inputs that is shown in **Fig. 8**.

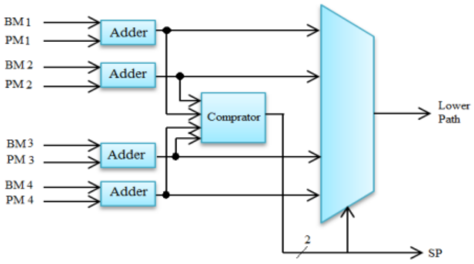

Fig. 7. The structure of ACS block for the break stage

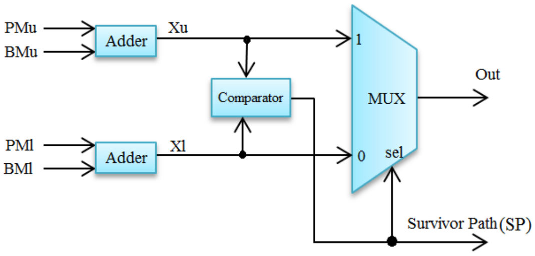

Fig. 8. The structure of ACS block with two inputs

A different structure means to add a new component and consequently increasing the chip area and power consumption. In this paper, to solve this problem, a structure is provided that uses the available normal ACSs with two inputs, to realize the ACSs with four inputs, Shown in **Fig. 9**.

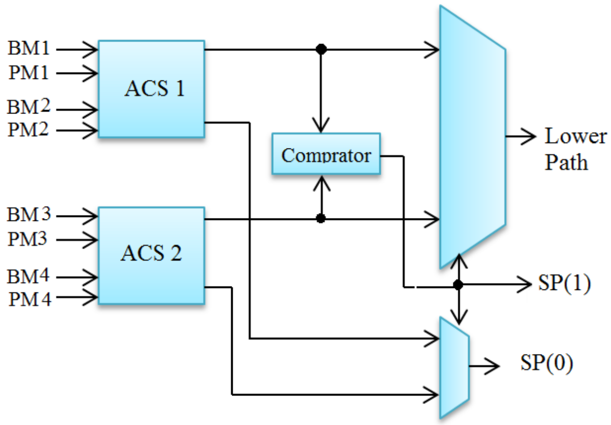

Fig. 9. The structure of low -consumption ACS offered for break stage.

4.2. Survivor Memory Unit

States of trellis diagram in BS are reduced by half. So the number of nodes in which the shorter path is chosen also reduces by half, and memory required for storing these routes is also reduced by half. Reducing the SMU, will leads to reduction of chip area and power consumption.

4.3. Trace Back Unit

The schematic view of TB unit of decimal decoder is shown in **Fig. 11**. The main components comprising the TB unit are cell-1 and cell-2.

The structure of components cell-1 and cell-2 is shown in **Fig. 10**. Also In this unit, components decline after the BS, leads to reduction of chip area and power consumption.

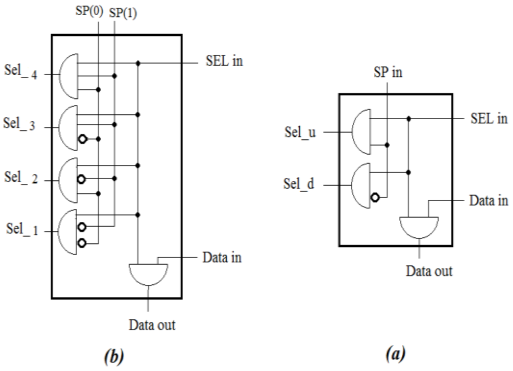

Fig. 10. Components of TB, a) cell-1, b) cell-2

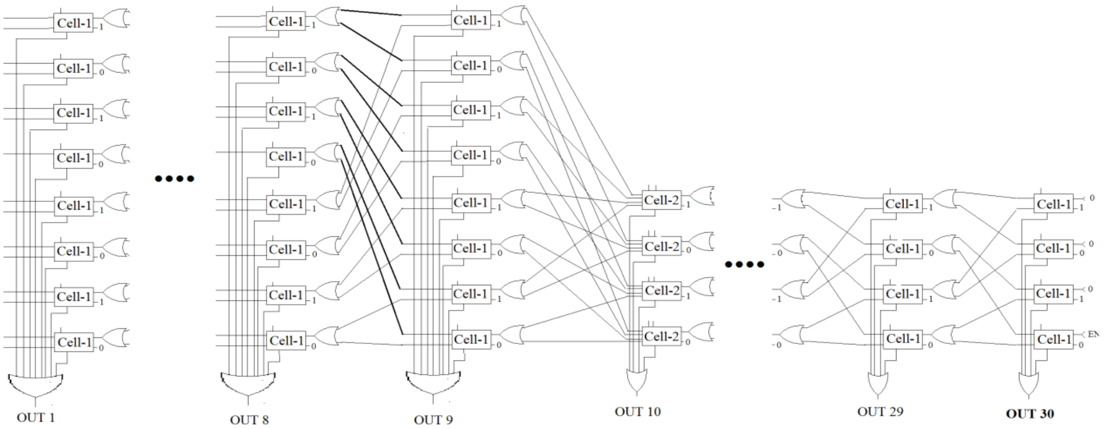

Fig. 11. Structure of TB unit of decoder with decimals in constrain length.

5. SIMULATION AND EXPERIMENTAL RESULTS

To test the designed decoder, we used the test bench that shown in **Fig. 12**. The “Serial Input Generation” block, generates the serial inputs for the encoder. With each clock pulse, one bit of data is placed randomly on output of the block. LFSR structure is used to generate random data [6]. Clock signal for this block is produced by “CLK1 Generator” block which is a division of the system clock. In experiments carried out in this paper, the value of this division is selected 2. “CLK2 Generator” Block produces the delayed signal of CLK1 and “CLK3 Generator” block produces delayed signal of CLK2, Which provides the clock signal for the next blocks.

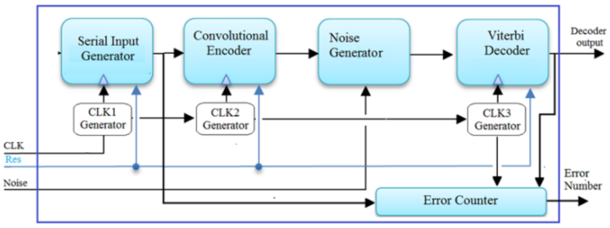

Fig. 12. test bench for testing the Encoder.

All test bench blocks in **Fig. 12** are described with VHDL language. Due to high usage rate of codes with the constrain length 6 and 7, Decimal code tested, was selected between this two codes and has a constrain length 6.3 and the data rate ½. The Isim simulator was used to simulate the test bench. Matching the decoder output with the Input of encoder, represents the correct description of decode blocks which this matching is shown in **Fig. 14**, for decimal code 6.3.

Described decoders are synthesized by Xilinx ISE12.1and they are implemented on a Xilinx Spartan3, Xc3s400 FPGA. The FPGA kit is shown in **Fig. 13**.

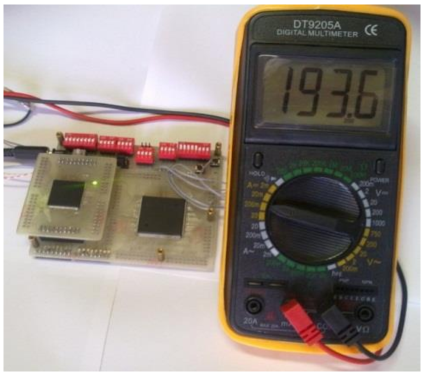

Fig. 13. FPGA test kit board.

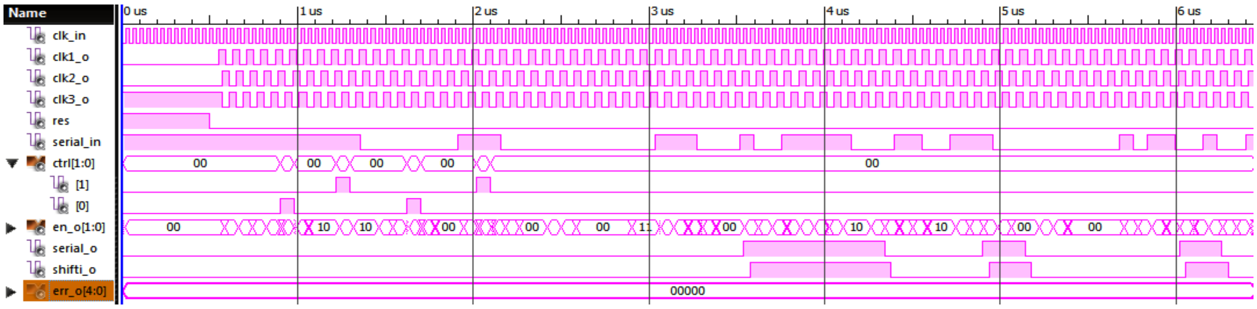

Fig. 14. The coded data recovery by the decoder with decimal in constrain length.

Using the MATLAB simulator, the BER value in SNR=4dB, is calculated for codes with constrain length 6, 7 and 6.3, and with measuring results of their power consumption is shown in **Table 2**.

Table 2. BER simulation and experimental power results

	K = 6	K = 7	K = 6.3
Dynamic Power (mW)	215	345	310
BER	57×10^{-4}	37×10^{-4}	50×10^{-4}

$$relative\ power\ consumption\ improvement = \frac{1015-946}{1015-858} \times 100\% = 27\% \quad (5)$$

$$relative\ performance\ attenuation = \frac{946-858}{1015-858} \times 100\% = 65\% \quad (6)$$

According to (5) and (6) decimal code with a constrain length of 6.3, to Improve 27% of relative power consumption of decoder, its relative error correction Performance, is decreased by 65%. To evaluate the speed performance of designed decoder, propagation delay of decoder units with constrain length 7 and 6.3 are estimated with Xilinx ISE12.1 and they are shown in Table 3.

Table 3. Propagation delay of decoder units for constrain lengths 7 and 6.3 (nS)

	K=7	K=6.3
BMU and PMU propagation delay	3.3	5.5
TB unit propagation delay	26.7	24.65

Propagation delay of BMU and PMU In decimal decoder has increased which is because of increased additional component to break the code. But the delay of TB In decimal decoder has been reduced. According to the fact that maximum frequency factor, is the TB unit's propagation delay, Frequency performance has been better in decimal decoder. The new decoder's speed improvement is provided in (7).

$$Frequency\ improvement = \frac{26.7-24.65}{26.7} \times 100\% = 7.7\,\% \quad (7)$$

6. CONCLUSION

Power consumption and chip area in convolutional codes decoders, increases exponentially with increasing the constrain length. The increase is significant in the constrain lengths greater than 9, and in some applications is limiting. Decimal code is a solution to fix this limitation; in applications where a code with constrain length N does not have the required error correction Performance, And the code with length of N+1 has an unauthorized Power consumption and chip area, using a code with decimal constrain length, is a good option.

In this paper convolutional code with the decimal constrain length was introduced, and for a code and sample conditions, the relative 27% improvement of decoder's power consumption and relative 65% weakening of error correction Performance was gained, The frequency Performance also improved by 7.7%. Weakening the error correction Performance was because the removal of latent bit which to retrieve it in the decoder, we used the primary method of repetition coding. By using better methods, the error correction performance can be get better. In this paper only a specific type of convolutional codes with decimal constrain length was dealt with. Perhaps other types can be suggested that have a more optimal performance. One outstanding advantage of decimal coding is their high diversity which allows the designers to implement an optimal coding for their designs. Also the concept of decimal can be employed in the characteristics of other convolutional codes (e.g. decimals in the data rate) and generate various codes.

REFERENCES

[1] U.Meyer-Baese, **"Digital Signal Processing with Field Programmable Gate arrays,"** *Third Edition , Springer*, 2007.

[2] P.Sweeney, **"Error Control Coding,"** *Wiley*, 2002.

[3] R.H. Morelos-Zaragoza, **"The Art of Error Correcting Coding,"** *Wile*y, 2002.

[4] S.Lin and D.J.Costello , **" Error Control Coding,"** *Prentice-Hall* ,1983.

[5] J.C. Moreira, **"Essentials of ERROR-CONTROL CODING,"** *Wiley*, 2006.

[6] Z. Navabi, **" Digital System Test and Testable Design,"** *Springer*, 2011.

[7] S.Ranpara, **"On a Viterbi decoder design for low power dissipation, "** *M.Sc, Thesis, Virginia Polytechnic Institute and State University*, 1999.