Maha A. Fleah

Control and Systems Eng.,
University of Technology
Baghdad, Iraq.
60715@student.uotechnology.edu.iq

Qusay F. Al-Doori

Control and Systems Eng.,
University of Technology
60022@uotechnology.edu.iq

State Space Parallelization Method for a 16-Bit Turbo Encoder

Abstract- *Turbo codes are widely used in digital communication systems. Their ability to reach the Shannon channel capacity made it the choice for most of the communication systems. Due to the huge amount of the transmitted data, there is a need to increase the processing speed of the encoders. The researchers used the state space technique to enhance the throughput of the turbo encoder. They apply it to increase the turbo encoder throughput from one bit per cycle up to 8 bit per cycle. The researchers applied the state space method to a three-flip flop, eight state Recursive Systematic Convolution Code circuit to achieve their goal. In this paper, we explored the state space technique and applied it to a four flip-flop Recursive Systematic Convolution Code circuit so that we can achieve a throughput of 16 bit per cycle. The circuit was designed and tested using MATLAB then implemented using FPGA to verify its operation.*

Keywords- *Turbo encoder, parallel concatenated, Recursive Systematic Code (RSC).*

1. Introduction

Channel coding is used to increase the reliability of the communication system by reducing the error in the received data. It accomplishes this task by adding redundant bits to the transmitted data so that the receiver can detect the error and correct it [1]. One of the essential error detection/correction circuits is the turbo encoder circuit. Turbo code provides a reliable transmission with a performance that reaches the Shannon limit [2,3]. The recent development in wireless communication that is used in devices like smartphones, notebooks, Personal Digital Assistant (PDA), increased the strain of the network capacity [2,4]. To overcome such a problem, the processing speed of the transmitted data should be increased by parallelizing the serial processes in the communication system [4]. One of the serial processing stages of the channel encoding in the wireless communication system is the turbo encoder [3].

Several researchers dealt with increasing the throughput of the turbo decoder. However few of them dealt with increasing the throughput of the turbo encoder. The throughput of the turbo decoder could be increased by the execution of the radix-4 algorithm [4,5]. Radix 4 is based on performing two processing steps /cycle so that the processing speed is increased and the clock latency is decreased [4,5]. On the other hand, the turbo encoder throughput can be increased using the state-space method [6,8].

The state-space method is used in other types of encoders to paralyze the circuits. Researchers such as [9-11] used the method to increase the throughput of the Cyclic Redundancy Check circuit. The use of the sate space method to paralyze the turbo encoder circuit was introduced by Purwita, Setio, and Adiono in 2011 [6]. They implement the method on a three flip-flop, 8 states Recursive Systematic Code (RSC) circuit such that the new circuit can generate 8 output bits per cycle. Their work was based on the Long-Term Evolution (LTE), technique where the turbo encoder is needed as the 8 states RSC.

In this work, we extend the method used by [6] so that it can generate a parallel representation for a 16 state 4 flip-flops RSC circuit. The new circuit could be used in satellite communication systems [1,12].

The paper is organized as follows: Section 2 describes the Turbo encoder structure. Section 3 describes the state space representation of the new architecture of the Turbo encoder. The results of the proposed algorithm are shown in section 4, and section 5 concludes the paper. The paper organized into logical parts or sections. Subsections are not numbered. Any subsection is given a brief heading. The contents include the introduction that should define clearly the nature of the problem, and the references should be made to previously published papers. The theoretical, experimental, results, discussions, and conclusions form the main sections of the paper. Theoretical section extends the analytical

background of the article and develops a new formulation of the problem. Calculations are achieved here using the developed equations, and the modifications should be pointed out. Depending on the suggested research methods, the experimental investigation is achieved, using the testing instruments or design and manufacturing a test rig. Materials and methods are detailed here. In the results and discussions section, the significance of the obtained results should be pointed out, and the citations and the discussions of the literature should be avoided in this section. Sometimes results and discussions are combined in one section.

2. The Structure of Turbo Encoder

Turbo encoder can encode the input data stream and generates the binary output data so that the output is concatenated with the transmitted data. This operation will prevent the noise from affecting the data message [1]. The general structure of a turbo encoder is shown in Figure 1, which shows that it consists of two RSC concatenated in parallel and one interleaved separated between them [13]. The purpose of using the interleaved is to rearrange the input data stream sequence; hence, the data stream is protected against error bursts. This will improve the error-correcting performance [14]. The interleaved makes the output of the two encoders completely different. Thus, even if the first encoder generates an output that will suffer from a high error rate, the other encoder will generate an output that will withstand a low error rate; this will decrease the error and improve the performance of the communication system [15]. There are many interleaved types, but the most general one which is being used is the random interleaved (pseudo interleaved) that randomize the binary input sequence depending on the given permutation pattern [16,17].
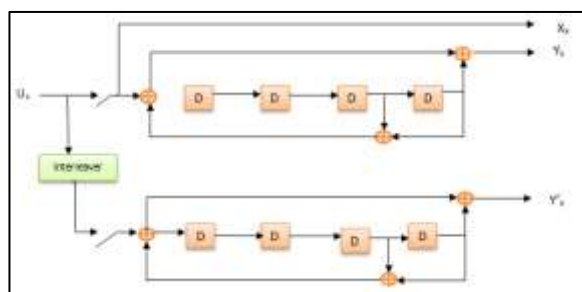

Figure 1: General structure of a rate 1/3 Turbo encoder

The RSC codes that are used in this design is of a rate equals to 1/3 and constraint length equal to 5. It is a 16 states encoder since there are 4 D Flip

Flop registers, these registers will generate 16 states (2^4). This type of encoder is useful and used in satellite communications [1,12].

3. State-Space Parallel Turbo Codes

The RSC is based on the Linear Feedback Shift Register (LFSR) in which the binary input (Uk) is propagated through the shift register flip-flops and XORed to produce the required output. The performance and the throughput of the LFSR can be increased by using the state space algorithm [7,18]. The state-space is a mathematical way that is used to implement the circuit in paralle, and also it is used to simplify the parallelism operation [9].

Figure 2 shows the 16 bit RSC circuit in which the parity bit (Y_k) is the output of the LFSR, and the systematic bit (X_k) is taken directly from the input (U_k). The circuit states (X_t) changes as the time (t) is increased to (t+1).

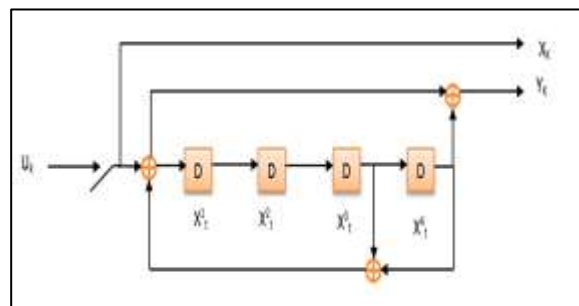**Figure 2: The 16 Bit RSC Code**

The 16 RSC encoder transfer function, driven from the circuit of Figure 2 is given by:

$$G(D) = (1, \frac{g_1(D)}{g_2(D)}) \qquad (1)$$

Where 1 represents the systematic output which is the same as the input, g_1 is the feed forward output, and g_2 is the feedback from the output to the input of the RSC encoder [15].

g_1 (D), and g_2(D) are given by:

$$g_1(D) = 1 + D^4 \qquad (2)$$

$$g_2(D) = 1 + D^3 + D^4 \qquad (3)$$

The turbo encoder inputs are represented as u_0, u_1,...........U_{k-1}. The outputs from the first RSC are Y_0, Y_1,.............Y_{k-1} and the outputs from the second interleaved RSC are Y_0', Y_1',................ Y_{k-1}', where k is the number of the input code word.

The state space equations driven from Eq. (2) and Eq. (3) are:

$$X_{t+1} = A.Xt + B. Ut \qquad (4)$$
$$Yt = C.Xt + D. Ut \qquad (5)$$

Where X_{t+1} is the next state, Xt is the current state, Yt is the output of the system and (A, B, C, D) are the state transition metrics which are given as:

$$X_t=\begin{bmatrix} X_t^1 \\ X_t^2 \\ X_t^3 \\ X_t^4 \end{bmatrix},\ A=\begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},\ B=\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$C = [0\ \ 0\ \ 1\ \ 0]$, and $D= [1]$ (6)

Equation (4) can be rewritten in another form to represent the relation between the previous state and the next state as.

$X_{(t+n)} = A.X_{t+ (n-1)} + B.U_{t+ (n-1)}$ (7)

Where n is positive number.

To find the equation that represents the relation between the previous state, current state and the next state of X_t, equation (7) can be modified as:

$X_{(t+n)} = A^n{}_t + A^{(n-1)}BU_t + A^{(n-2)} BU_{(t+1)} +$
$...............+BU_{(t+(n-1))}$
(8)

The output denoted by Y

$Y_{(t+n)} = C_{(t+ (n-1))} + D\ U_{(t+ (n-1))}$
(9)

By using the multiplication results of the states (A, B, C, and D) and also by using Eq. (8) and Eq. (9) and for $i = 0,1.........,\frac{K}{16}-1$, and $t = 16i$, then we can obtain 16 level convolution encoder equation as show below.

$X^1{}_{(16i)+16} = X^3{}_{(16i)} + X^4{}_{(16i)} + U_{(16i)} + U_{(16i)+4} + U_{(16i)+5} + U_{(16i)+6} + U_{(16i)+7} + U_{(16i)+9} + U_{(16i)+11} + U_{(16i)+12} + U_{(16i)+15}$ (10.a)

$X^2{}_{(16i)+16} = X^1{}_{(16i)} + U_{(16i)+3} + U_{(16i)+4} + U_{(16i)+5} + U_{(16i)+6} + U_{(16i)+8} + U_{(16i)+10} + U_{(16i)+11} + U_{(16i)+14}$ (10.b)

$X^3{}_{(16i)+16} = X^2{}_{(16i)} + U_{(16i)+2} + U_{(16i)+3} + U_{(16i)+4} + U_{(16i)+5} + U_{(16i)+7} + U_{(16i)+9} + U_{(16i)+10} + U_{(16i)+13}$ (10.c)

$X^4{}_{(16i)+16} = X^3{}_{(16i)} + U_{(16i)+1} + U_{(16i)+2} + U_{(16i)+3} + U_{(16i)+4} + U_{(16i)+6} + U_{(16i)+8} + U_{(16i)+9} + U_{(16i)+12}$ (10.d)

$Y_{(16i)} = X^3{}_{(t)} + U_{(16i)}$ (11.a)

$Y_{(16i)+1} = X^2{}_{(t)} + U_{(16i)+1}$ (11.b)

$Y_{(16i)+2} = X^1{}_{(t)} + U_{(16i)+2}$ (11.c)

$Y_{(16i)+3} = X^3{}_{(t)} + X^4{}_{(t)} + U_{(16i)} + U_{(16i)+3}$ (11.d)

$Y_{(16i)+4} = X^2{}_{(t)} + X^3{}_{(t)} + U_{(16i)+1} + U_{(16i)+4}$ (11.e)

$Y_{(16i)+5} = X^1{}_{(t)} + X^2{}_{(t)} + U_{(16i)+2} + U_{(16i)+5}$ (11.f)

$Y_{(16i)+6} = X^1{}_{(t)} + X^3{}_{(t)} + X^4{}_{(t)} + U_{(16i)} + U_{(16i)+3} + U_{(16i)+6}$
(11.g)

$Y_{(16i)+7} = X^2{}_{(t)} + X^4{}_{(t)} + U_{(16i)} + U_{(16i)+1} + U_{(16i)+4} + U_{(16i)+7}$ (11.h)

$Y_{(16i)+8} = X^1{}_{(t)} + X^3{}_{(t)} + U_{(16i)+1} + U_{(16i)+2} + U_{(16i)+5} + U_{(16i)+8}$ (11.i)

$Y_{(16i)+9} = X^2{}_{(t)} + X^3{}_{(t)} + X^4{}_{(t)} + U_{(16i)} + U_{(16i)+2} + U_{(16i)+3} + U_{(16i)+6} + U_{(16i)+9}$ (11.j)

$Y_{(16i)+10} = X^1{}_{(t)} + X^2{}_{(t)} + X^3{}_{(t)} + U_{(16i)+1} + U_{(16i)+3} + U_{(16i)+4} + U_{(16i)+7} + U_{(16i)+10}$ (11.k)

$Y_{(16i)+11} = X^1{}_{(t)} + X^2{}_{(t)} + X^3{}_{(t)} + X^4{}_{(t)} + U_{(16i)} + U_{(16i)+2} + U_{(16i)+4} + U_{(16i)+5} + U_{(16i)+8} + U_{(16i)+11}$
(11.l)

$Y_{(16i)+12} = X^1{}_{(t)} + X^2{}_{(t)} + X^4{}_{(t)} + U_{(16i)} + U_{(16i)+1} + U_{(16i)+3} + U_{(16i)+5} + U_{(16i)+6} + U_{(16i)+9} + U_{(16i)+12}$
(11.m)

$Y_{(16i)+13} = X^1{}_{(t)} + X^4{}_{(t)} + U_{(16i)} + U_{(16i)+1} + U_{(16i)+2} + U_{(16i)+4} + U_{(16i)+6} + U_{(16i)+7} + U_{(16i)+10} + U_{(16i)+13}$ (11.n)

$Y_{(16i)+14} = X^4{}_{(t)} + U_{(16i)} + U_{(16i)+1} + U_{(16i)+2} + U_{(16i)+3} + U_{(16i)+5} + U_{(16i)+7} + U_{(16i)+8} + U_{(16i)+11} + U_{(16i)+14}$ (11.o)

$Y_{(16i)+15} = X^3{}_{(t)} + U_{(16i)+1} + U_{(16i)+2} + U_{(16i)+3} + U_{(16i)+4} + U_{(16i)+6} + U_{(16i)+8} + U_{(16i)+9} + U_{(16i)+12} + U_{(16i)+15}$ (11.p)

4. Implementation and Results

To implement the 16 RSC model given by equations (10) and (11), two phases were presented. The first is by using MATLAB to validate the integrity of the method. The second is by using Quartus II to measure the design parameters.

The circuit in Figure 2 is designed and implemented using VHDL Language under Quarts 11.1 Software. Figure 3 shows the general RSC that process one input (In1), and generates one output (out1) in each cycle.
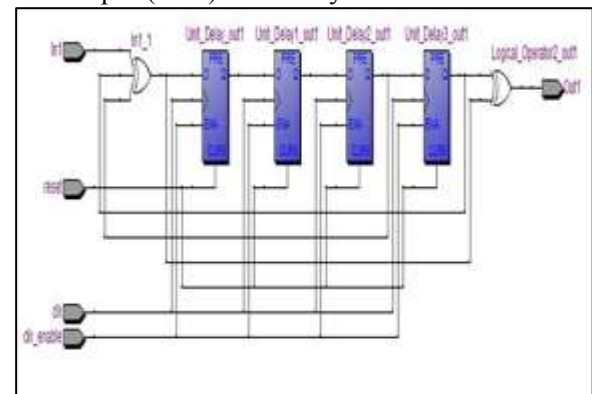

Figure 3: Serial RSC Circuit

The original turbo encoder circuit that was shown in Figure 1 is built using VHDL language, and its circuit is shown in Figure 4.In which one input is entered to the first RSC to produce the systematic output and the first parity output, and the same

input but interleaved is entered to the second RSC to produce the second parity output. The input signal is taken over and produces the output only when the Cleanable signal is high (logic one) and when the CLK signal is a positive edge. The reset signal is used to reset all the circuit when its signal is active low (logic 0). These signals are beneficial in the encoder circuit and in any digital circuit to control it and prevent the output signal from changing its state when the input signal is changed suddenly.
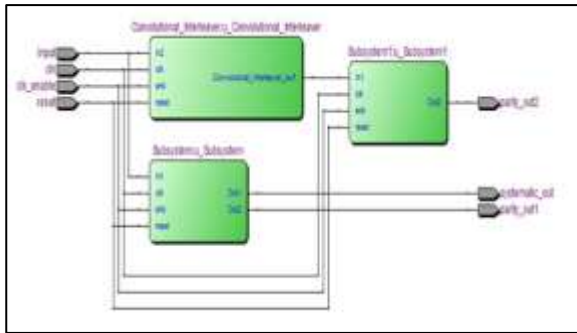

Figure 4: The original Turbo encoder circuit

Equations (10) and (11) are used to implement the parallel RSC circuit that enters 16 input bits and generates 16 output bits in one clock cycle. The parallel design of the 16 RSC circuit is shown in Figure 5. The proposed circuit used 20 registers, 41 logic elements.
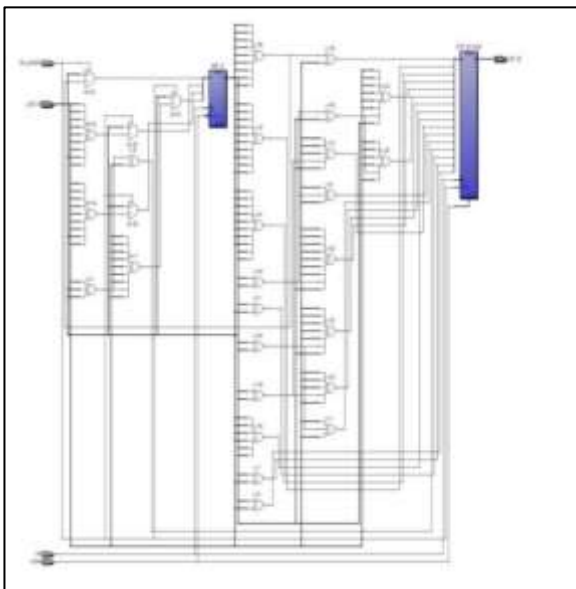

Figure 5: Parallel 16 bit RSC

To check that this circuit works correctly, the circuit was tested using the ModelSim program. The same data is entered to both the proposed parallel circuit and the original circuit. The same results have been achieved, but in a different

clock cycle, wherein the parallel circuit only one clock cycle was needed, but in the serial circuit, 16 clock cycles were needed to generate the 16 outputs. The period of the clock cycle was chosen as 100ps. The MATLAB software is also used in order to compare the parallel proposed circuit with the serial circuit it was noticed that the simulation results were similar.

The simulation results are shown in Figures 6 and 7 of the original serial circuit and the parallel circuit. The same input bits are used in both circuits (serial circuit and the parallel circuit), the serial circuit takes 6400 ps to complete the processing operation while the parallel circuit takes only 500 ps to process 64 bits.
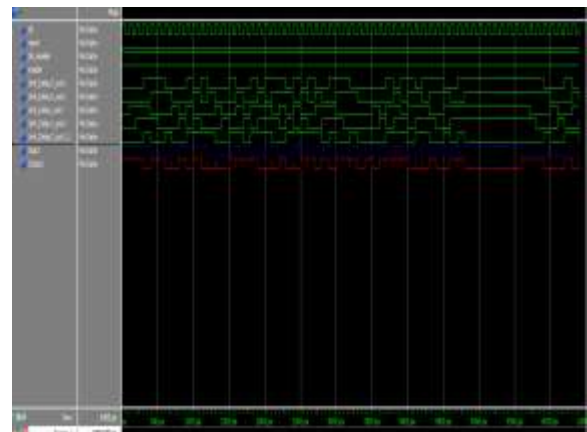

Figure 6.The serial circuit simulation result

Figure 7.The parallel circuit simulation result

5. Conclusion

This work describes the Turbo encoder and new architecture that can provide high throughput that is needed for a wide range of communication systems. This new architecture is based on the state-space representation that is used to increase the speed of the circuit by using multiple parallel inputs. The proposed algorithm reduces the number of clock cycles of the encoder circuit by processing 16 inputs in one cycle instead of processing one input in one cycle. The circuit algorithm based on an 8-bit paralyzation

algorithm given by [6]. When applying the new algorithm on the Terasic DE2 FPGA board, it used 20 registers, 41 logic elements, which is less than 1% of the Altera Cyclone II chip space. The maximum frequency that the time quist analysis gives was 582.07 MHz, so the maximum throughput is 9312 Mbit/sec. It could be concluded that this turbo encoder architecture has a very high throughput that can be used in the new communication technology.

References

[1] H. Robert and M. Zaragoza, "The art of error correcting," A John Wiley & Sons, 2002.

[2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," IEEE transaction on communications, vol. 44, no. 10, pp. 1261–1271, 1996.

[3] T. K. Moon, "Error Correction Coding: Mathematical Methods and Algorithms," A John Wiley & Sons, 2005.

[4] H. Chuang, K. Tseng, and W. Fang, "A High-Throughput Radix-4 Log-MAP Decoder With Low Complexity LLR Architecture," IEEE Journal of Solid-State Circuits, vol. 1, no.1, pp.231–234, 2009.

[5] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE," IEEE Journal of Solid-State Circuits, vol. 46, no. 1, pp. 8–17, 2011.

[6] A. A. Purwita, A. Setio, and T. Adiono, "Optimized 8-Level Turbo Encoder Algorithm and VLSI Architecture for LTE," International Conference on Electrical Engineering and Informatics, July 2011.

[7] M. Ayinala and K. K. Parhi, "High-Speed Parallel Architectures for Linear feedback Shift Registers," IEEE Transaction on signal processing, vol. 59, no. 9, pp. 4459–4469, 2011.

[8] M. Ayinala and K. K. Parhi, "Efficient parallel VLSI architecture for linear feedback shift registers," Signal Process. Syst. (SIPS), 2010 IEEE Work, pp. 52–57, 2010.

[9] Q. Al-doori and O. Alani, "A Multi Polynomial CRC Circuit for LTE-Advanced Communication Standard," 7th Computer Science and Electronic Engineering Conference (CEEC), pp. 19–23, 2015.

[10] J. H. Derby, "High-Speed CRC Computation Using Stat-Space Transformations," Global Telecommun. Conf. (GLOBECOM'01) .IEEE, vol. 1, pp. 166–170, 2001.

[11] M. Grymel and S. B. Furber, "A novel programmable parallel CRC circuit," IEEE Trans. Very Large Scale Integr. Syst., vol. 19, no. 10, pp. 1898–1902, 2011.

[12] D. J. C. MacKay, "Information theory, inference, and learning algorithms," Cambridge University, Version 7.2 (fourth printing) March 28, 2005.

[13] "Multiplexing and channel coding," 3GPP TS 36.212 version 8.4.0, September 2008.

[14] S. Kumar and H. Dalal, "Performance comparision of turbo codes and modified turbo codes," International Journal of Science, Eng and Technology Research (IJSETR), vol. 3, no. 5, pp. 1520–1523, 2014.

[15] A. Imran, "Software implementation and performance of UMTS turbo decoder," MSc. thesis, Tampere Univ of Technology, 2012.

[16] C. Benkeser, A. Burg, T. Cupaiuolo, and Q. Huang, "Design and optimization of an HSDPA turbo decoder ASIC," IEEE Journal of Solid-State Circuits ,vol. 44, no. 1, pp. 98–106, 2009.

[17] H.Ljunger,"Turbo decoder with early stopping criteria," MSc. thesis, Lund Univ 2016.

[18] M.Ayinala and K. K. Parhi, "Efficient parallel architecture for linear feedback shift registers," IEEE Transaction on signal processing, vol. 62, no. 11, pp. 1068–1072, 2015.