Implementation of Turbo Code Based Xilinx System Generator

Mahmood F. Mosleh[1(✉)], Mais F. Abid[1], and Mohammed Al-Sadoon[2]

[1] Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
drmahmoodfarhan@gmail.com, maisfalah70@gmail.com
[2] School of Engineering and Informatics, Bradford University, Bradford, UK
M.A.G.Al-Sadoon@bradford.ac.uk

Abstract. The turbo code (TC) is one of the most type of Forward error correction (FEC) code that used in Third Generation Partnership Project (3GPP) which standardization works by the Long Term Evolution (LTE). In this paper an integrated system based TC by LTE standard is implemented. A Simulink model is designed using Xilinx System Generator (XSG) applied by MATLAB version R2012a, ISE design suite version 14.5 and applied by Xilinx Spartan 6 xc6sIx45t-fgg484 board using FPGA clock period 10 ns. The system is tested in two steps, in the first one, the Spartan kit is connected within loop of transmitter and receiver. In the second step the Spartan kit is connected to execute the whole system and display the output signal using real scope device. The results confirm that the proposed system is decoded the original signal without any errors.

Keywords: Turbo code · Xilinx system generator · Spartan

1 Introduction

The FEC is active tools that used to increase the reliability of data transmitted in digital communication. Iterative Decoder (ID) is one of attractive channel code which is used in modern communication. TC is the better one of FEC, it is an efficient tool which approaches the limit of Shannon through the use of a convolution codes with a largest constraints length or a block code with the largest blocks length, by the use of iterative and recursive coder TC beat to this limitation [1].

TC used in 3GPP which standardization works by LTE. One of the major task of 3GPP LTE is to increase the speed of radio access in mobile communications. The nature iterative of TC raised their complexity compared to other decoding algorithms, there is two main types of ID algorithms, Soft-Output-Viterbi Algorithm (SOVA) and Maximum A posteriori Probability (MAP) Algorithm which required intricate decoding operations through many of the iterations cycles. So, for FPGA implementation of turbo codes, decrease the complication of the ID while Maintain the performance of bit error rate (BER) is an important in the consideration of the design [2].

For several hardware operations the XSG supplies a group of Simulink blocks that can be implement on different Xilinx FPGAs kits, it is possible to use these blocks to

V. Sucasas et al. (Eds.): BROADNETS 2018, LNICST 263, pp. 317–324, 2019.
https://doi.org/10.1007/978-3-030-05195-2_31

simulate the operation of the hardware system by using XSG Simulink environment. Most Digital signal processing (DSP) needs floating point format for representing data, although it is easy to build this on many of the computers that executing the software of high level such as Simulink, but this is very difficult in the world of the hardware because of the complexity of the floating point arithmetic implementation. That's XSG is used fixed point format to represent all its values in the system.

This paper is divided into two main sections. XSG simulation results where the proposed system is implemented using Simulink MATLAB program and System Generator results with hardware tests are represented where the proposed system is implemented using XSG tools and downloaded to Xilinx Spartan 6 kit for real time by using hardware Co-simulation. Hardware co-simulation merges the ability of MATLAB simulation with a hardware implementation to confirm the ability of the system. The design uses the MATLAB version R2012a and ISE design suite version 14.5.

A contribution of this paper is to implement an integrated communication system included a TC with an encoder and decoder in the transmitter and receiver respectively and to use a co-simulation program interface with MATLAB in order to implement a Simulink model in to the Hdl code without need to write the program in VHDL specially with high complexity circuit like MAP decoder.

Many researchers have been implemented Turbo Decoder (TD) in FPGA blocks in various method. In [3], the authors discussed the implementation of MAP TD with Software Radio Modem. In [4], the authors discussed the efficient power implementation of the Log MAP TD. In [5], the authors discussed the speed improvement and the TD implementation on Central Processing Unit (CPU) based on software defined radio (SDR). The implementation of 3GPP and 3GPP2 turbo encoder on FPGA Xilinx Virtex- IV is reported in [6]. In [7], the authors discussed the implementation of the 3GPP TD on Graphic Processing Unit (GPU).

2 Turbo Code for LTE

2.1 Encoder

TC are a type of the high performance FEC codes, and that was the first operation codes to approach closely to the capacity of the channel. Turbo codes are used in 3GPP LTE mobile communication. In this research, it has been used a turbo encoder according to LTE standard which consists of a two parallel convolution code with two 8-state constituent encoders and one convolutional interleaver as shown in Fig. 1. The task of the convolution interleaver is to take the block of the M-bit data and perform a permutation of this block. The performance of TC depends on the structure of the interleaver and the permutation sequences [8]. The basic rate of Turbo coding according to LTE specifications is 1/3. A block of M-bit data is encoded into a code word with 3 M + 12 data bits, where the 12 tail bits are used for the trellis termination. When begin to encode the input bits of the information, the first value of the shift registers of the 8-state constituent encoders shall be all zero. The convolution encoder may be represented as the follows [9]:

$$G_0 = G1 = 1 + D + D2 + D3 + D6 \tag{1}$$

Then convolution encoder basically multiplies the generated polynomials by the input bit, as the follows

$$A1(k) = G_0 * U(k) = abc\ldots g \tag{2}$$

$$A2(k) = G_1 * U(k) = ABC\ldots G \tag{3}$$

Then interleaving the outputs from the convolution encoder

$$E(k) = aAbBcC\ldots g \tag{4}$$

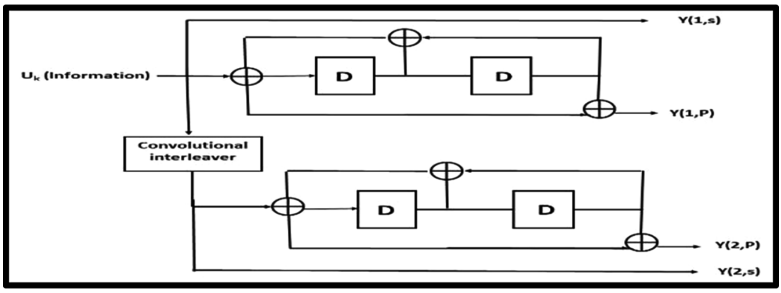

Fig. 1. Structure of the 1/3 CR Turbo encoder in LTE

2.2 The Iterative Decoder

In this research, the algorithm of ID for TD based MAP decoder is shown in Fig. 2. The ID structure is composed of two MAP decoders and interleaver which installed between these two MAP decoders to make permutations to the sequence of the input. The decoding is an iterative operation which exchanges the extrinsic information also called (L-posteriori) between MAP decoders. MAP decoder consisted of forward metric α, backward metric β and extrinsic information L-posteriori. The L-posteriori is calculated by dividing all values for positive states with corresponding values for negative states, and then take the log for such result [10, 11]:

$$log \frac{\sum_{b^+} \alpha_{t-1}(s').\beta(s).Z_{et}(s', s)}{\sum_{b^-} \alpha_{t-1}(s').\beta(s).Z_{et}(s', s)} \tag{5}$$

Where Z is the branch metric, s' is the previous state and s is the current state.

The α and β are forward metric and backward metric respectively. Each one are calculated using the following equations

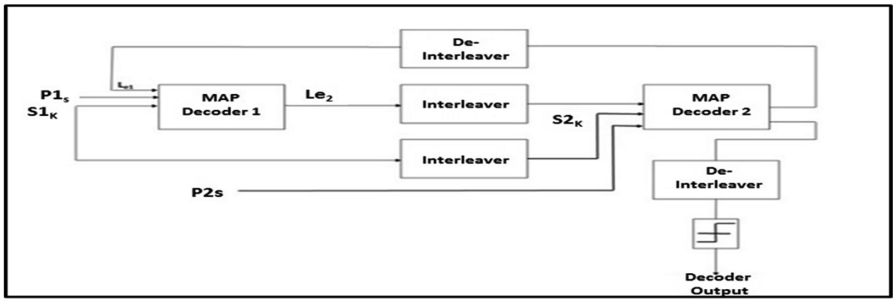

Fig. 2. The ID block diagram

$$\alpha_t(s) = \frac{\sum_{s'} \alpha_{t-1}(s') Z_t(s', s)}{\sum_s \sum_{s'} \alpha_{t-1}(s') Z_t(s', s)} \tag{6}$$

$$\beta_t(s) = \frac{\sum_{s'} \beta_{t-1}(s') Z_t(s', s)}{\sum_s \sum_{s'} \beta_{t-1}(s') Z_t(s', s)} \tag{7}$$

The results of Eq. 1 will be two states either positive or negative one which indicate the decoded bit is either 1 or 0 respectively

3 System Model

The system model used in this research is show in Fig. 3. In this section it will explain the function of each block of such system as follows

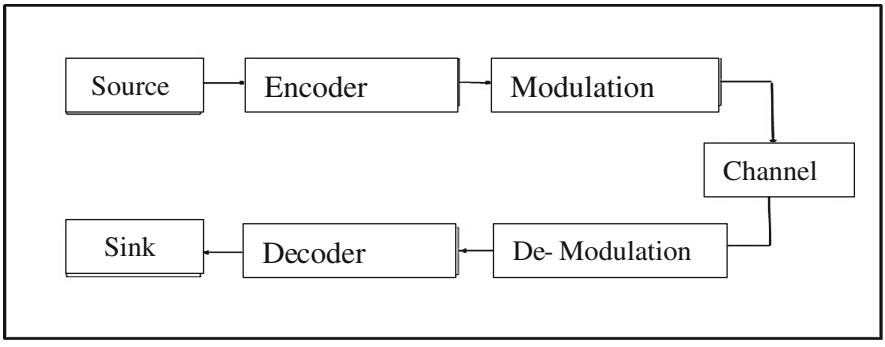

Fig. 3. System model

The source is generating a bit streams. The probability of 1's is 0.5. This stream bit is fed to encoder to apply the encoding process. The encoder block will encode the input data with a CR of 1/3 so that each input bit will be represented by 3 bits. First one is the same input bit called systematic bit, the second and third is the adaptive parity bit. Then the encoder output will be multiplexing the 3 parallel bits into one series row.

The Modulation block will map the input bits into one symbol. The number of bits in each symbol depended on the level of modulation of BPSK which maps one bit in each symbol. The channel used is AWGN type for the purpose of experimentally testing. The parameters of this channel are Eb/No, number of bits per symbol, power of the input signal and the period of symbol. The rest blocks is form the receiver end which apply the inverse function of the corresponding transmitted end.

4 Simulation

The implementation of proposed TD in XSG is show in Fig. 4. The simulation is done by using MATLAB Simulink model and XSG tools. The outputs from the implemented and simulated design have been checked to make sure that the implemented circuit work well as the simulated design and there is not much difference between the MATLAB and ISE designs. The design has used MATLAB version R2012a and ISE design suite version 14.5.

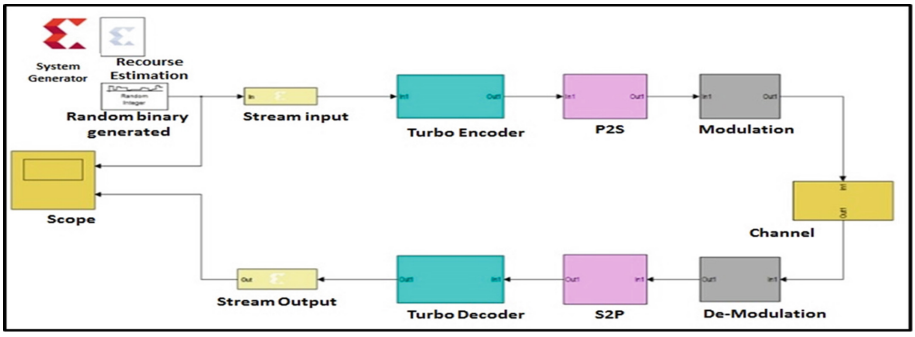

Fig. 4. The proposed system

5 Results

Two cases of hardware proposed Tc implementation has been done using Xilinx spartan6 board.

5.1 The Simulation of SG Result

In this case the input bit stream has been generated using random binary signal generated block as show in Fig. 4. Also, to confirm the result other source using LFSR has been used to generate the signal with 8 numbers of bits, 15 feedback polynomial and 3F initial value in hexadecimal as shown in (Fig. 5), for the above two signals the generated bit file is loaded into Xilinx Spartan 6 xc6sIx45t-fgg484 board using FPGA clock period 10 ns with Simulink system period of 1/3 s. The output signal for random binary signal generated is displaying using the scope of Fig. 4. Figure 6 the transmitted and received signal of random binary signal generated. It is show that the decoder is deconstructing the transmitted signal without error. For LFSR the signal is displayed using scope of Fig. 5. Also, the transmitted and received signal show in Fig. 7 confirm that the decoder can be decoded any signal regardless of source type.

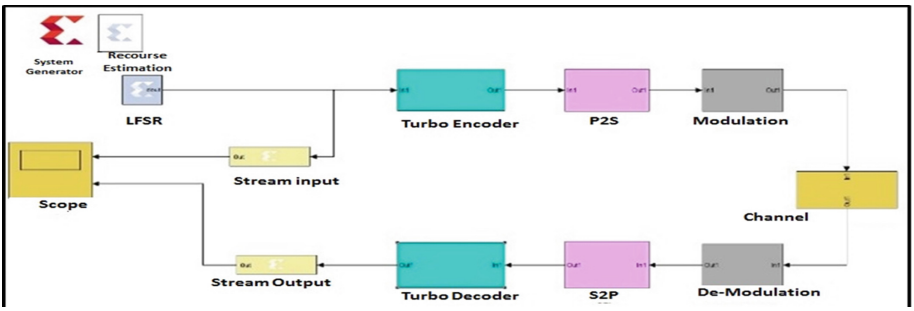

Fig. 5. The proposed system

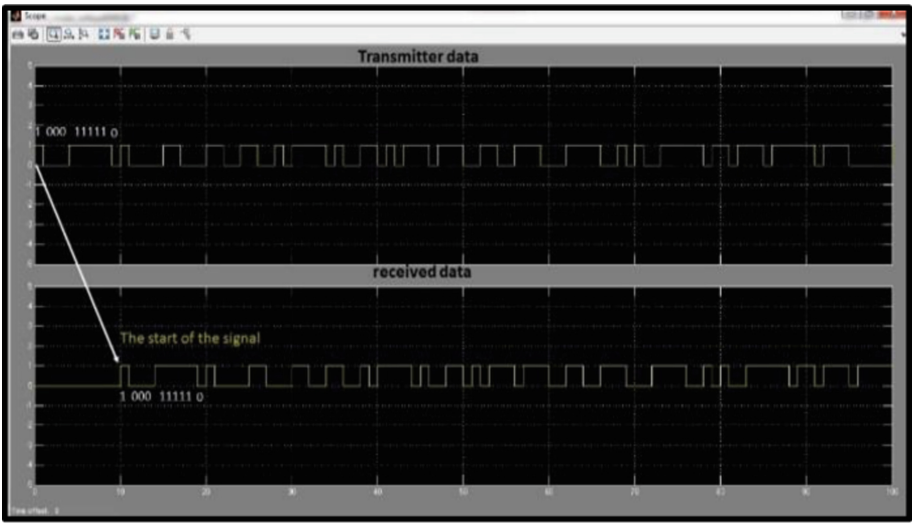

Fig. 6. The XSG results with random integer input block

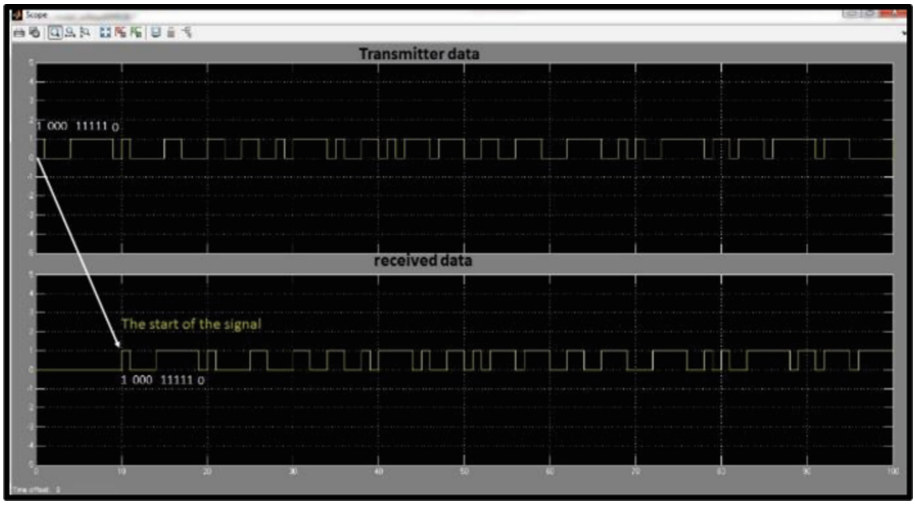

Fig. 7. The XSG results with LFSR input block

5.2 The FPGA Implementation

For the purpose of FPGA implementation it has been used Spartan 6 xc6sIx45t-fgg484 board used for hardware processing. The Spartan 6 kit is connected to the PC through Parallel/USB Programming cable for Joint Test Action Group (JTAG) configuration/ communication. Figure 8 illustrates the hardware implementation which consists of two scopes, one is virtual in PC screen to display the transmitted signal and the second is real device to shows the decode signal. As it is clear in Fig. 8 the system is decode the signal without any error, note that the signal in real scope is a part of virtual scope due to different in scale.

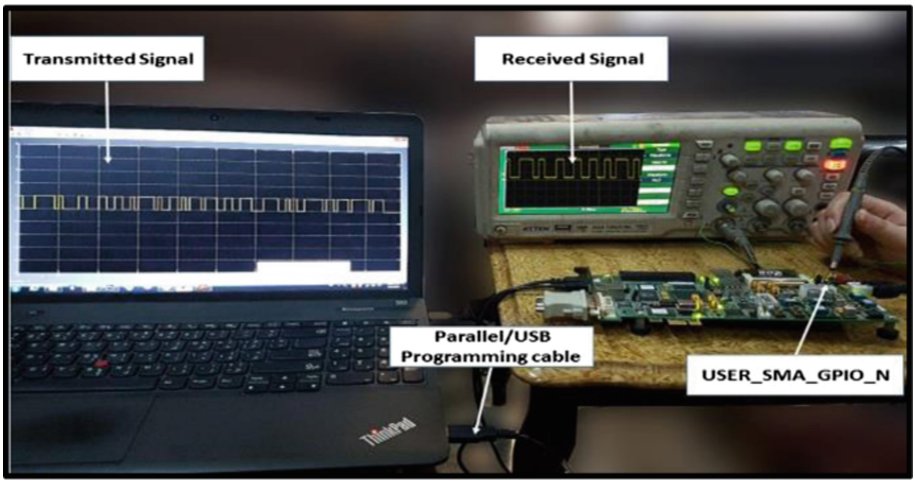

Fig. 8. The XSG results with LFSR input block

6 Conclusion

In this paper it has been building an integrated system consists of an encoder and ID using in addition to modulation and demodulation with AWGN channel. The system is implemented in two cases, the first on is to connect the Spartan 6 in the loop of the transmitter and the receiver and shows the results in two virtual scopes. The second case connect the Spartan to execute the whole system and shows the decoded signal in real device scope. The results confirm that the system is applied in real time and reconstruct the signal without any error.

References

1. Spanos, A., et al.: Reduced complexity rate-matching/de-matching architecture for the LTE turbo code. In: 2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 411–414. IEEE (2014)
2. Taskaldiran, M., Morling, R.C., Kale, I.: The modified Max-Log-MAP turbo decoding algorithm by extrinsic information scaling for wireless applications. In: Powell, S., Shim, J. (eds.) Wireless Technology. Lecture Notes in Electrical Engineering, vol. 44, pp. 203–213. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-71787-6_13
3. Lin, S., Costello, D.J.: Error Control Coding. Prentice Hall, Englewood Cliffs (2004)
4. Kang, B., et al.: Power-efficient implementation of turbo decoder in SDR system. In: IEEE International SOC Conference, Proceedings, pp. 119–122. IEEE (2004)
5. Huang, L., et al.: A high speed turbo decoder implementation for CPU-based SDR system. In: Proceeding of IET International Conference on Communication Technology and Application, pp. 19–23 (2011)
6. Tripathi, S., Mathur, R., Arya, J.: Unified 3GPP and 3GPP2 turbo encoder FPGA implementation using run-time partial reconfiguration. In: Wireless Telecommunications Symposium (WTS), pp. 1–8. IEEE (2010)
7. Yoge, D.R.N., Chandrachoodan, N.: GPU implementation of a programmable turbo decoder for software defined radio applications. In: 2012 25th International Conference on VLSI Design (VLSID), pp. 149–154. IEEE (2012)
8. Raut, R.D., Kulat, K.D.: Int. J. Comput. Appl. (0975-8887) **1**(24) (2010)
9. Sadjadpour, H.R., Sloane, N.J.A., Salehi, M., Nebe, G.: Interleaver design for turbo codes. IEEE J. Sel. Areas Commun. **19**(5), 831–837 (2001)
10. Yoo, I., Kim, B., Park, I.-C.: Reverse rate matching for low-power LTE-advanced turbo decoders. IEEE Trans. Circuits Syst. I Regul. Pap. **62**(12), 2920–2928 (2015)
11. Nithya, B., Pandiaraj, P., Thenkumari, K.: Development of error correction mechanism based on RCIC turbo codes. LTE Network, Elysium Journal (2015)