Journal of Engineering Sciences
Vol11,Issue2,FEB/2020
ISSN NO:0377-9254

IMPLEMENTATION OF TURBO ENCODER IN VEHICLE SYSTEM

[#1]EADA EVANGELINE, M.Tech Student

[#2]RAMESH JITTY, Assistant Professor

Dept of ECE

JYOTHISHMATHI INSTITUTE OF TECHNOLOGY AND SCIENCE, KARIMNAGAR, TELANGANA.

Abstract

Studies design and implementation of the Turbo encoder to be an embedded module in the in-vehicle system (IVS) chip. Both serial and parallel computations for the encoding technique are studied. The two design methods are presented and analyzed. Developing the parallel computation method, it is shown that both chip size and processing time are improved.

1. Introduction

MANY advanced wireless communication standards adopted turbo codes as the channel coding scheme due to its near Shannon error-correcting performance [1]. The decoding procedure is performed in two different half iterations, where the reliability of received bits is computed in the form of extrinsic values using interleavers and soft-input–soft-output (SISO) decoders in an iterative way. On even half iterations, the decoding process is performed on the noninterleaved data and parity, whereas on odd half iterations, the interleaved data are decoded.

The extrinsic values, representing the reliability of the information bits, are sent to another half iteration by passing through the interleaver/deinterleaver unit until the acceptable error level is achieved. Recently, long-term evolution (LTE) advanced has been dominated as the next-generation wireless communication standard, which is aimed at higher peak data rates close to 3 Gb/s [2]. The turbo decoder, which is specified in LTE, reveals to be a limiting block toward this goal due to its iterative decoding nature, high latency, and significant silicon area consumption.

The decoding procedure is performed using the algorithm presented in [3]. Since the implementation of the actual maximum a posteriori (MAP) algorithm incurs very high computational complexity, typically, two modified forms of the MAP algorithm, i.e., the max-log-MAP and log-MAP algorithms [4], [5], are commonly realized instead. In these two alternative methods, the MAP core consists of log-likelihood ratio (LLR) units, as well as the core units to compute α, β, and γ, i.e., the forward, backward, and branch metrics, respectively.

In fact, the α and β units, due to their recursive computation nature, are the most challenging units to implement, occupying almost 40% of the whole MAP core area [6]. The γ unit, on the other hand, is a trivial part of the turbo decoder, consisting of few addition computations. Therefore, an area-efficient architecture for α and β metrics computation is highly desirable, which has always been a challenge in literature. In order to address this challenge, in this brief, a new relation between the α and β metrics is introduced; based on this new relation, a novel add–compare–select (ACS) unit for forward and backward computation is proposed. The proposed scheme results in, at most, an 18.1% reduction in the silicon area compared with the designs reported to date.

2. Basic design of LTE turbo decoder

During the past few years, modern 3G/4G wireless communication systems such as 3GPP (3rd Generation Partnership Project)

UMTS/HSPA+ (Universal Mobile Telecommunications System/High-Speed Packet Access Evo-lution) [2], 3GPP LTE (Long Term Evolution) and LTE-Advanced [3] have been deployed to meet the evergrowing demand for higher data rates and better quality of service. High throughput is one of the most important requirements for emerging wireless communication standards. For instance, the 3GPP UMTS standard Release 11 extends HSPA+ with several key enhancements including increased bandwidth and number of antennas.

These enhancements lead to 336Mbps peak data rate with 2 _ 2 MIMO (multiple-input multiple-output) and 40MHz bandwidth (or 4 _ 4 MIMO and 20MHz bandwidth) [2]. Recently, up to 672Mbps data rate has been proposed for the future release of 3GPP standards [4], [5]. As a 4G candidate, the 3GPP LTE-Advanced promises up to 1 Gbps data rate as its long term goal. LTE-Advanced [3] have been deployed to meet the ever-growing demand for higher data rates and better quality of service. High throughput is one of the most important requirements for emerging wireless communication standards. For instance, the 3GPP UMTS standard Release 11 extends HSPA+ with several key enhancements including increased bandwidth and number of antennas. These enhancements lead to 336Mbps peak data rate with 2 _ 2 MIMO (multiple-input multiple-output) and 40MHz bandwidth (or 4 _ 4 MIMO and 20MHz bandwidth) [2]. Recently, up to 672Mbps data rate has been proposed for the future release of 3GPP standards [4], [5].

As a 4G candidate, the 3GPP LTE-Advanced promises up to 1 Gbps data rate as its long term goal. Turbo codes are specified in many wireless communication standards such as the HSPA+ and LTE/LTE-Advanced as for-ward error-correction codes to ensure reliable communications via wireless channels, due to their outstanding error-correcting performance [6]. A turbo decoder contains two key compo-nents: soft-input softoutput (SISO) decoders and interleavers. During the decoding process, log-likelihood ratio (LLR) soft values are exchanged between component SISO decoders in an iterative way. The interleaver is a critical component for the turbo decoder to achieve good error-correcting performance, by permuting the LLRs randomly between iterations and max-imizing the effective free distance of turbo codes.

Since the introduction of turbo codes, numerous VLSI architectures and implementations have been proposed [7]–[10]. To achieve high throughput, parallel turbo decoding architectures are usually employed, in which several SISO decoders operate in parallel with each working on a segment of the received codeword [9]– [24]. The parallel turbo decoders suffer from severe memory conflict problems due

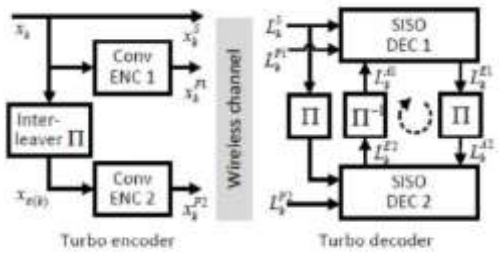

Fig 1. Block diagaram of Turbo Encoder/Decoder

to the randomness of the interleaver, which becomes a major challenge for high throughput designs and implementations .

3. proposed design of LTE turbo decoder

3.1 Introduction to IVS

The European emergency call (eCall) system is a telematics system designed to save more lives in vehicle accidents. It is a governmental mandatory system that is to be implemented by March 2018. The EU eCall system provides an immediate voice and data channel between the vehicles and an emergency center after car accidents. The data channel provides the emergency center

with the necessary data for emergency aids. The EU eCall system main parts includes the in-vehicle system (IVS), the public safety answering point (PSAP), a cellular communication channel. The IVS activates the data channel automatically when a car accident occurs. The IVS collects the minimum set of data (MSD) that includes GPS coordinates, the VIN number, and all required data for an emergency aid. It sends the MSD to the closest PSAP through a cellular channel in up to 4 seconds [1]. The PSAP sends the emergency team to the location of the accidents.

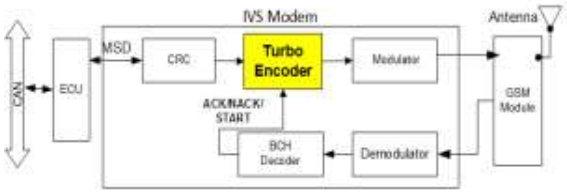

Fig. 2: The IVS block diagram

The IVS modem employs multiple modules for the MSD signal processing. The modules of the IVS are shown in Figure 2. The IVS employs a Turbo encoder as a forward error correcting (FEC) [1]. The Turbo encoder implements the digital data encoding technique in data transmissions. Turbo coding is one of the most popular and efficient coding technique to improve bit error rate (BER) in digital communications [3] [4]. The cyclic redundancy check (CRC) [5], the modulator.

.The turbo encoder technique is one of the most powerful FEC techniques in digital communication [8]. The IVS employs a Turbo encoder module with 1/3 code rate. The Turbo encoder functionalities are detailed in the third generation partnership project (3GPP) standards. The 3GPP Turbo encoder is illustrated in Figure 3 [8]. The input signal of the turbo encodes is the MSD data appended with the CRC parity bits in binary. The block length of the MSD data is 1148 bits. The output of the module is the MSD encoded data in binary. Implementing the turbo coding technique with 1/3 coding rate and thrills bits, the length of the output is 3456 bits. The thrills structure has an impact of the Turbo encoder [9].

The Turbo encoder employs a parallel concatenated convolutional code (PCCC). The PCCC uses two constituent encoders with eight states as it is shown in Figure 2. The initial status of the register are zeros. The first constituent takes the MSD bits and implements the employed convolutional technique.

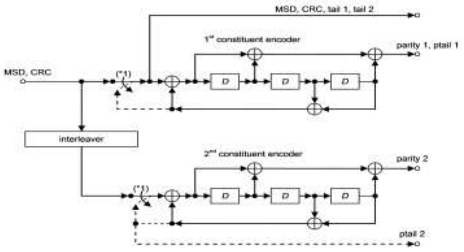

Fig. 3: The structure of the Turbo encoder

parity1 bits. The second constituent implements an identical technique of the first constituent, but it calls for the MSD bit after they are interleaved with a 3GPP designed interleaver technique [8].

The length of the input data, parity1, and parity2 are 1148 bits. There are 12 bits of the tail bits. They are driven from the shift register feedback. The tail bits are applied for end points between the encoded data blocks. The output structure of the Turbo encoder is illustrated in Figure 4.

Fig. 4: The output buffer of the Turbo encoder.

Random Errors:

Error locations are independent of each other. Error on one location will not affect the errors on other locations. Channels that introduce these types of errors are called channels without memory (since the channel has no knowledge of

error locations since the error on location does not affect the error on another location).

Burst Errors:

Errors are depended on each other. For example, in channels with deep fading characteristics, errors often occur in bursts (affecting consecutive bits). That is, error in one location has a contagious effect on other bits. In general, these errors are considered to be dependent and such channels are considered to be channels with memory.

Interleaver/ Deinterleaver :

One of the most popular ways to correct burst errors is to take a code that works well on random errors and interleave the bursts to "spread out" the errors so that they appear random to the decoder. There are two types of interleavers commonly in use today, block interleavers and convolutional interleavers.

The block interleaver is loaded row by row with L codewords, each of length n bits. These L codewords are then transmitted column by column until the interleaver is emptied. Then the interleaver is loaded again and the cycle repeats. At the receiver, the codewords are deinterleaved before they are decoded. A burst of length L bits or less will cause no more than 1 bit error in any one codeword. The random error decoder is much more likely to correct this single error than the entire burst.The parameter L is called the interleaver degree, or interleaver depth. The interleaver depth is chosen based on worst case channel conditions. It must be large enough so that the interleaved code can handle the longest error bursts expected on the channel. The main drawback of block interleavers is the delay introduced with each row-by-row fill of the interleaver.

Convolutional interleavers eliminate the problem except for the delay associated with the initial fill. Convolutional interleavers also reduce memory requirements over block interleavers by about one-half [1]. The big disadvantage of either type of interleaver is the interleaver delay introduced by this initial fill. The delay is a function of the interleaver depth and the data rate and for some channels it can be several seconds long. This long delay may be unacceptable for some applications. On voice circuits (as in GSM), for example, interleaver delays confuse the unfamiliar listener by introducing long pauses between speaker transitions. Even short delays of less than one second are sufficient to disrupt normal conversation. Another disadvantage of interleavers is that a smart jammer can choose the appropriate time to jam to cause maximum damage. This problem is overcome by randomizing the order in which the interleaver is emptied.

In practice, interleaving is one of the best burst-error correcting techniques. In theory, it is the worst way to handle burst errors. Why? From a strict probabilistic sense, we are converting "good" errors into "bad" errors. Burst errors have structure and that structure can be exploited. Interleavers "randomize" the errors and destroy the structure. Theory differs from reality,however. Interleaving may be the only technique available to handle burst errors successfully.

For example, Viterbi [2] showed that, for a channel impaired by a pulse jammer, exploiting the burst structure is not enough. Interleaving is still required. This does not mean that we should be careless about our choice of code and take up the slack with long interleavers. Codes designed to correct burst errors can achieve the same performance with much shorter interleavers. Until the coding theorists discover a better way, interleaving will be an essential error control coding technique for bursty channels.

4. CONCLUSION

Turbo encoder module is designed and implemented to be an embedded module in the IVS modem. Xilinx tools and Verilog HDL are employed to design and simulate the module.

Both serial and parallel computation techniques are studied for the encoding process. Comparing with the serial computation technique, the parallel computation encoding, improves the processing time by 58% and logic utilization by 73%. The processing time enhancement can be seen in both simulation and analyzing the chip processing.

REFERENCES

[1] "eCall data transfer; in-band modem solution; general description," 3GPP, Tech. Rep. TS26.267.

[2] Eroupean Commission, "eCall: Time saved = lives saved," Press Release, Brussels, August 21, 2015. website: http : //ec.europa.eu/digital − agenda/en/ecall − time − saved − lives − saved.

[3] A. Saleemet at. "Four-Dimensional Trellis Coded Modulation for Flexible Optical Communications," IEEE Journal of Lightwave Technology., vol. 35, no. 2, pp. 151-158, Nov. 2017.

[4] 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, Jan. 2011.

[5] M. Nader and J. Liu, "Design and implementation of CRC module of eCall in-vehicle system on FPGA," SAE Technical 2015-01-2844, 2015, doi:10.4271/2015-01-2844.

[6] M. Nader and J. Liu. "Developing modulator and demodulator for the EU eCall in-vehicle system in FPGAs" in IEEE, 2016 International Conference on Computing, Networking and Communications (ICNC), Hawaii, USA, Feb. 15-18, 2016, pp. 1-5

[7] M. Nader and J. Liu."FPGA Design and Implementation of Demodulator/Decoder Module for the EU eCall In-Vehicle System" in International Conference on Embedded Systems and Applications (ESA'15), Las Vegas, USA, July 27-30, 2015, pp. 3-9.

[8] "Technical Specification Group Radio Access Network; Multiplexing and channel coding (FDD)," 3GPP, Tech. Rep. TS22.212.

[9] D. Viktor, K. Michal, and D. Milan"Impact of trellis termination on performance of turbo codes," in ELEKTRO, 2016, pp.48-51.

[10] B. Muralikrishna, G.L. Madhumati, H. Khan, K.G. Deepika, "Reconfigurable system-on-chip design using FPGA," in 2nd International Conference on Devices, Circuits and Systems (ICDCS), 2014, pp.1-6.