Indian J.Sci.Res. 17(2): 287-290, 2018 ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

MODELLING OF EFFICIENT INTERLEAVING IN A MIMO-OFDM COMMUNICATION SYSTEM

[1] H.Anita, [2]Priyanka, [3]G. Jahnavi,
[1,2,3] Department of Electronics and Communication Engineering, Mahaveer Institute of Science & Technology, Hyderabad

Abstract-This work is based a memory-efficient and faster interleaver implementation technique for MIMO-OFDM communication systems on FPGA. The IEEE 802.16 standard is used as a reference for simulation, implementation, and analysis. This is the method for inter leaver design on FPGA and its memory utilization. This paper work concentrate on efficient interleaver design for IEEE802.16 system implemented on FPGA. The work focuses on the interleaver design. Our goal is to achieve minimum memory usage, faster interleaving, and increased speed of the overall system.

Keywords: MIMO-OFDM, IEEE802.16, FEC, Interleaving.

I. Introduction

The IEEE 802.16 defines the standard for broad band wire less access covering the physical layer and medium access specifications for wireless metropolitan area networks (WMAN) . The IEEE 802.16 Air Interface Standard is a technology that is playing a key role in fixed broad band wireless MAN . The forward error correction(FEC) mechanism in the standard plays a very important role in its performance . A number of techniques are being used to achieve highly effective error-control coding such as Turbo codes and concatenated codes . However , interleaving also plays a major role in the FEC mechanism. The aim of interleaving is to reorder the incoming data and make the adjacent bits non-adjacent by a factor,to cope with the burst errors occurring during the transmission of data over the channel. Memory utilization and frequent memory accesses time area crucial part of Interleaver design, targeting less memory utilization and reduced memory access inorder to reduce the power dissipation of the overall system.This paper is organized as follows. Section II presents an overview of proposed system.Section III presents the design of the whole MIMO-OFDM transmitter and puts an emphasis on a innovative design of encoder and puncturing.Section IV discuss the simulation results and analysis of coding and puncturing techniques. SectionV concludes the paper.

II. System Description

Block Diagram

The basic OFDM communication system physical layer is shown in Figure 2. The forward error correction (FEC) blocks include convolutional encoding, puncturing, and interleaving. A modification of the system described in Figure 1 is to use two separate data streams to enhance the data rate and possibly increase the number of antennas by using spatial as well as transmit diversity. However, In this analysis, only spatial diversity isused by having two parallel data streams that make up a 2×2 MIMO-OFDM communication system.

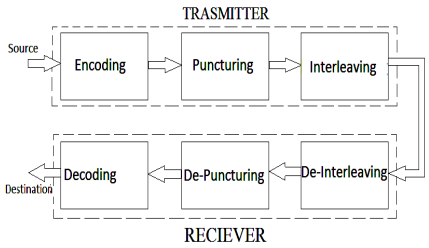


Figure 1:OFDM System

Convolutional Encoding

In telecommunication a convolution code is a type of error-correcting code in which each m - bit information symbol (each m- bit string) to be encoded is transformed into an n-bit symbol,where m/n is the code *rate* ($n \geq m$).The transformation is a function of the last k information symbols, where k is the constraint length of the code.

Convolutional codes are used extensively in numerous applications inorder to achieve reliable data transfer, including digital video, radio , mobile communication , and satellite communication . These codes are often implemented in concatenation with a hard-decision code, particularly Reed Solomon. Prior to turbo codes, such constructions were the most efficient, coming closest to the Shannon limit. To convolutionally encode data , start with k memory registers , each holding 1 input bit . Unless other wise specified , all memory registers start with a value of 0. The encoder has n modulo-2 adders (a modulo 2 adder can be implemented with a single Boolean XOR gate, where the logicis: 0+0=0 , 0+1=1 , 1+0=1 , 1+1=0) ,and n generator polynomials— one for each adder (see Figure below).An input bit m_1 is fed into the left most

[1]Corresponding Author

register.

Using the generator polynomials and the existing values in there main in g registers, the encoder outputs *n* bits. Now bit shift all register values to the right (*m*1 moves to *m*0, *m*0 moves to *m*-1) and wait for the next input bit . If there are no remaining input bits,the encoder continues output until all registers have returned to the zero state.

The Figure2 .shown above is a rate 1/3 (*m*/*n*) encoder with constraint length(*k*) of 3. Generator polynomials are *G*1=(1,1,1) , *G*2=(0,1,1) , and *G*3=(1,0,1). Therefore , output bits are calculated (modulo 2) as follows:

- *n*1=*m*1+*m*0+*m-1*
- *n*2=*m*0+*m*-1
- *n*3=*m*1+*m*-1.

Puncturing

In coding theory, puncturing is the process of removing some of the parity bits after encoding with an error-correction code . This has the same effect as encoding with an error-correction code with a high error rate,or less redundancy . However , with puncturing the same decoder can be used regard less of how many bits have been punctured , thus puncturing considerably increases the flexibility of the system without significantly increasing its complexity. Puncturing is often used with the Viterbi algorithm in coding systems.

A punctured code is obtained by periodically deleting encoded symbols from ordinary encoded symbols this process is known as puncturing process . If a rate of '1/n' parent encoder is punctured by deleting some of the _np' encoded bits to _p' information bits then _p' is called puncturing period.After puncturing symbol from the encoded sequence corresponding to same amount of information is reduced therefore the rate of encoder is increased by puncturing process.

Representation of punctured codes.

For rate 1/n parent encoder the puncturing patern can be represented as _nxp' matrix p whose elements are 1's and 0's with 1 indicating inclusion and 0 indicating deletion.

Example: given encoder generator polynomial.

Figure 2: Rate 1/3 non-recursive ,non-systematic convolutional encoder with constraint length 3.It indicated that with in two encoded blocks , the first bit of second encoded block is deleted.

Interleaving

Interleaving is process or methodology to make a system more efficient, fast and reliable by arranging data in a noncontiguous manner. There are many uses for interleaving at the system level , including:

Storage:

As hard disks and other storage devices are used to store user and the systemdata, there is always a need to arrange the stored data in an appropriate way.

Error Correction:

Errors in data communication and memory can be corrected through interleaving.

Multi-Dimensional Data Structures.

Interleaving is frequently used in digital communication and storage systems to improve the performance of forward error correcting codes.Many communication channels are not memoryless : errors typically occur in bursts rather than independently.If the number of errors within a code word exceeds th eerror-correcting code's capability, it fails to recover the original code word. Interleaving a melio rates this problem by shuffling source symbols across several code words , there by creating a more uniform distribution of errors.

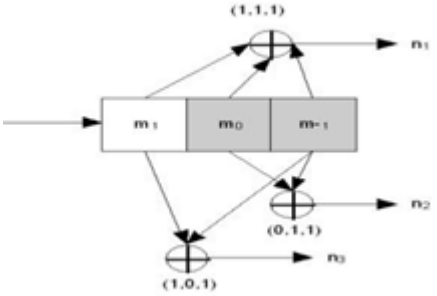


Figure 2

Interleaver designs include:

- Rectangular (or uniform) interleaver (similar to the method using skip factors described above)
- Convolutional interleavers
- Random interleavers (where the interleaver is a known random permutation)
- S-random interleaver (where the interleaver is a known random permutation with the constraint that no input symbols within distance S appear within a distance of S in the output).
- Another possible construction is a contention-free quadratic permutation polynomial (QPP) . It is used for example in the 3 GPP Long Term Evolution mobile telecommunication standard.

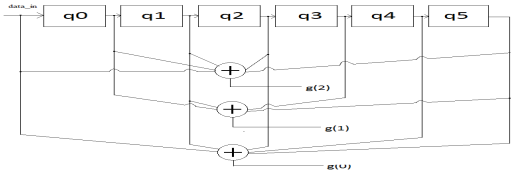
In multi-carrier communication systems , additional interleaving across carriers may be employed to mitigate the effects of prohibitive noise on a single or few specific carriers (e.g.,frequency-selective fading in OFDM transmission).

III .Design Of Convolutional Encoder And Puncturing

1/3 Coderate Convolution Encoding

To convolutionally encode data , start with *6* memory registers , each holding 1 input bit . All memory registers start with a value of 0 . The encoder has 3 modulo-2 adders (a modulo 2 adder can be implemented with a single Boolean XOR gate , where the logic is : 0+0=0 , 0+1= 1 , 1+0=1 , 1+1=0) , and *3* generator polynomials—one for each adder (see Figure below).

An input bit is fed into the left most register . Using the generator polynomials and the existing values in the remaining registers, the encoder outputs *3* bits . Now bits hit all register values to the right (*q0* moves to *q*1 , *q*1 moves to *q2,q2* moves to *q3,q3* moves to *q*4, *q4* moves to *q*5) and wait for the next input bit.If there are no remaining input bits,the encoder continues output until all registers have returned to the zero states.The Figure3. Shown above is a rate 1/3 (*m*/*n*) encoder with constraint length (*k*) of 3. Generator polynomials are *G*1=(1,1,1,0,1,1), *G*2=(1,1,0,1,0,1) , and *G*3=(1,1,0,1,0,1).



Therefore,output bits are calculated (modulo 2) as follows:

G[0]=are*g[0]*=(data_in+q[1]+q[2]+q[4]+q[5]

G[1]=(data_in+q[1]+q[3]+q[5]).

G[2] =(data_in+q[1]+q[2]+q[5]

Puncturing.

The elements of the puncturing array are zeros and ones , corresponding to keeping or deleting the encoded bits respectively.

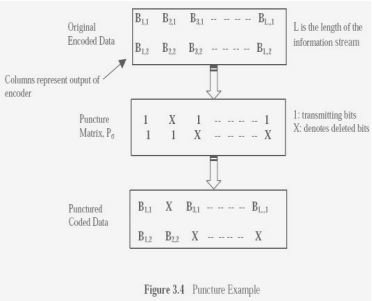


Figure 3

The puncturing device deletes symbols from the code sequence according to the puncturing matrix.

IV. Simulation Results And Analysis.

Clock block divides input clock by 3 and generator output clock which is slower than the main input.

The schematic for convolution encoding consists of inputs : clock , data_in , enable and reset and outputs are code_out which is of 3 bits and en_out.

From the above results we can see that all the results are triggered at positive edge of clock pulse.And whenreset=‘1‘,data_in=‘1,enable=‘1‘ all the outputs are zeros.When reset=‘0‘,data_in=‘0‘,enable=‘1‘and All outputs are shown in above figure.

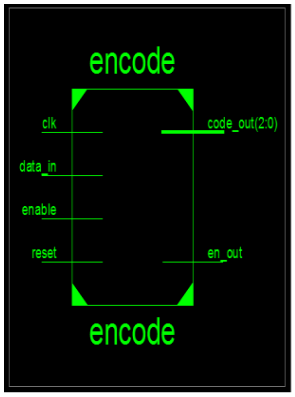


Figure4 : schematic for convolution encoding.

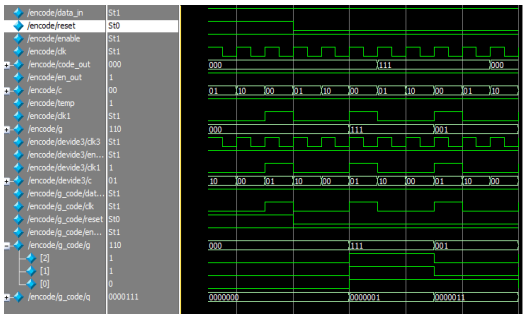


Figure5: Results for convolution encoding.

Convolutional Encoding

The encoder takes data in as its input and gives out code_out,which is encoded data. The encoder consists of two subblocks namely Generator polynomial block and clock divider block.Generator polynomial block takes single bit data_in as the input and generates polynomial g[0] , g[1] and g[1]. When count is “001”then code_out={g[0],g[0],g[0]},When count is “011” then code_out={g[1],g[1],g[1]} ,When count is “101” then code_out={g[2],g[2],g[2]}.And thus one bit input data is encoded as 3 bit output.

Puncturing:

Puncturing technique is based on the puncturing matrix ,it consists of 4 bit data_in and 4bit data_out. Initially puncturing matrix is defined which is of 14 bits .

Depending on the value of each bit in the punctured matrix input is transferred to output.That is if bit of puncturing matrix is 1 then input is transferred to output and if bit is zero then input is not transferred to output.

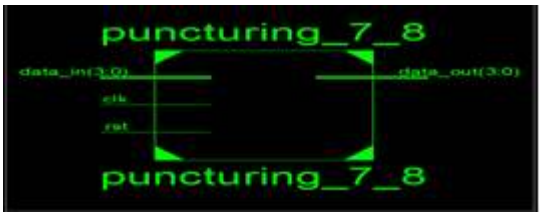


Figure 6:schematic for puncturing

The schematic for puncturing consists of inputs data_in,clk and reset and output data_out which is of 4bits.

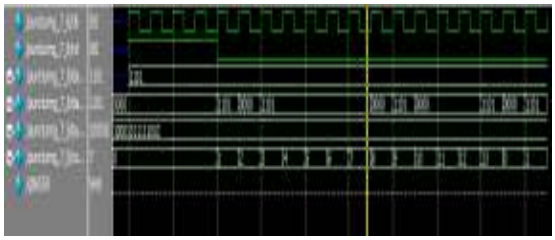


Figure7: Results for puncturing code. From the above result it can be seen that,When the reset=‘0‘then all the outputs are zero. When reset=‘1‘ and if_bit ‘in the punctured matrix is_1‘ then ouput is same as input and if bit in the punctured matrix is_0 ‘then output is zero.

V.Conclusion

In this paper,it is provided an efficient way to design the IEEE 802.16 transmitter for FPGA . A special design method is used to implement the interleaver with minimum memory requirement and initial latency. This approach can also be used to design other high-speed communication systems or to improve their speed . The proposed optimizations could be utilized in real applications since they only require to replace the current interleaving parameters and do not involve any hardware alteration.Here the convolution encoding and puncturing techniques havebeen coded and stimulated. Furthermore interleaving , deinterleaving , depuncturing and convolution decoding are to be accomplish in the next phaseofwork.

References

[1]IEEE Standard for Local and Metropolitan Area Networks–Part16: Air Interface for Broadband Wireless Access Systems , IEEE Std. 802.16-2009,

[2] Fernandon H. Gregorio―OFDM phycoding an dinterleaving.‖ May 2007

[3] Y.-N.Chang,―A low-cost dual-mode deinterleaver design , IEEE Transactions on Consumer Electronics , May.2008 .

[4] X.Yin and J.Liu, ― Design and implementation of an improved 3G turbocodes interleaver for 3GPP system , in 9th International Conference on Electronic Measurements and Instruments (ICEMI), Beijing , China,Aug2009

[5]N.Crisan,L.C.Cremene,E.Puschita,andT.Palade,―Spectral Efficiency Improvement for the under-11GHz Broadband Wireless Access, in InternationalConferenceonTelecommunications,(ICT), Jun.2008,pp.

[6] S.-J.ParkandJ.-H.Jeon,―Interleaver optimization of convolutional turbo code for 802.16 systems ,IEEECommunicationLetters,May.2009.

[7] Z.Iqbal and S.Nooshabadi , Effects of Channel Coding and Interleaving in MIMO-OFDM Systems ,in 54th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS),Seoul,Korea,Aug.2011.

[8] C.Berrou,Y.Saouter, C. Douillard, S. Kerouedan,andM.Jezequel,―Designing good permutations for turbo codes: toward a single model,inProc.IEEEICC,Paris, France, June2004.

[9] S.Haene,D.Perels,andW.Fichtner,"System-Level Characterization of a Real-Time4x4MIMO-OFDM Transceiver on FPGA ,"in European SignalProcessingConference,Poznan,Poland,Sep. 2007.

[10]S.Haene,A.Burg,D.Perels,P.Luethi,N.Felber,andW. Fichtner,"FPGA Implementation of Viterbi Decoders for MIMO-BICM ," in Conference Record of theThirty-Ninth Asilomar Conference on Signals , Systems and Computers,Oct.28-Nov.01 2005.