Design and Implementation of an area efficient interleaver for MIMO-OFDM systems:


4th Sem , M.Tech (Digital electronics), SSIT, Tumkur, Karnataka
2 Lecturer, Dept. of E&C, SSIT, Tumkur, Karnataka
3 Dean & HOD, Dept. of E&C, SSIT, Tumkur, Karnataka

ABSTRACT- This work is based on 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 and analysis. This is the method for interleaver design on FPGA and its memory utilization. This project work concentrate on efficient interleaver design for IEEE 802.16 system implemented on FPGA. Our goal is to achieve minimum memory usage, faster interleaving, and increased speed of the overall system.

Key words: MIMO-OFDM, IEEE 802.16, FEC, INTERLEAVING.

II. SYSTEM DISCRITION

A. CONVOLUTIONAL ENCODING.

In telecommunication, convolution code is a type of error-correcting code in which

- Each m-bit information symbol (each m-bit stream) to be encoded is transformed into an n-bit symbol, where m/n is the code rate (n>m).
- The transformation is a function of the k information symbols.

To convolutionally encode data, start with a k memory registers, each holding one bit input. All memory registers start with a value of zero, unless otherwise specified. The encoder has n modulo-2adders (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 $n$ generator polynomials, one for each adder [3](see Figure 2 below). An input bit $m_i$ is fed into the leftmost register.

Using the generator polynomials and the existing values in the registers, the encoder output is $n$ bits. Now shift all register values to the right 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.

If a rate of ‘1/n’ parent encoder is punctured by eliminating some of the ‘np’ encoded bits to ‘p’ information bits [4]. 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 pattern can be represent 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.

$$G = [1+D^2, 1+D+D^2]$$

And punctured matrix as

$$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

It indicates that within two encoded blocks, the first bit of second encoded block is eliminated.

C. INTERLEAVING

Interleaving is a process that makes a system more efficient, fast and reliable by arranging data in a non-contiguous manner. Uses of interleaving,

- **Storage:** As hard disks and other storage devices are used to store user and the system data, there is always a need to arrange the stored data in an appropriate way.
- **Errors in data can be corrected through interleaving.**
- **Multi-Dimensional Data Structures.**

Interleaver designs include:

- Rectangular (or uniform) interleaver (similar to the method using skip factors described above)
- Convolutional Interleavers.
- Random interleavers [6](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. It is used for example in 3GPP long term evolution mobile telecommunication standard.

---

Figure 2: Rate 1/3 non-recursive, non systematic convolutional encoder with constraint length 3.

The Figure 2 shown above is a rate 1/3 ($m/n$) encoder with constraint length ($k$) of 3. Generator polynomials are $G_1 = (1, 1, 1), \ G_2 = (0, 1, 1), \ G_3 = (1, 0, 1)$. Therefore, output bits are calculated (modulo 2) as follows:

- $n_1 = m_1 + m_0 + m_{-1}
- n_2 = m_0 + m_{-1}
- n_3 = m_1 + m_{-1}$.
III . SYSTEM IMPLEMENTATION.

The IEEE 802.16 (WiMAX) system is implemented on FPGA for design emulation and verification. Convolutional encoder and puncturing blocks are implemented using shift registers and XOR gates. For QPSK, 16-QAM, and 64-QAM mapping, puncturing reduces the code rate to 3/4 as described in [1]. There is no puncturing used for BPSK mapping as the code rate is always rate 1/2. The interleaver is implemented using RAM blocks in the FPGA and employing logic cells (LC) for the state machine of the address generator for read/write operations.

Double buffering technique is used to implement the interleaver to eliminate the delay in the interleaving process. After the first block of symbols is stored in the buffer, the address generator[7] starts generating read addresses and starts reading data from the buffer. In the mean time, the second buffer is filled with incoming data and the interleaver will start reading from the second buffer after the first one is read out completely. Table I shows the buffer sizes for different interleaving schemes used in our system. The number of buffers increase with the increase in modulation symbol size, so that we can write/read data from them at the same time.

<table>
<thead>
<tr>
<th>Modulation schemes</th>
<th>BPSK</th>
<th>QPSK</th>
<th>16-QAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer size</td>
<td>384</td>
<td>384</td>
<td>384</td>
</tr>
<tr>
<td>No of buffers</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

A. Interleaver for BPSK Mapping:

For BPSK mapping, the interleaver is implemented using a single memory block of double the required size. For example, an interleaver of size 192 is implemented using a buffer of 384 bits as shown in table I. Incoming bits are first stored in the RAM until 192 bits are filled and then the read-out is enabled. A state machine generates write addresses for the RAM to write data to it. After 192 writes to the RAM, it asserts the read enable signal and starts generating read addresses for the RAM while continuing with the write process. This way the next part of the buffer is filled when the interleaver finishes reading the first 192 bits and then it starts reading the next 192 bits.

B. Interleaver for QPSK Mapping:

For QPSK mapping, the interleaver is implemented using two memory blocks of double the required size. A state machine generates a single address for the two RAMs to write data to them simultaneously. The pattern thus formed can be read out column-wise from RAM1 and RAM2 alternatively to implement the interleaver function. After 192 writes to each RAM, it asserts the read enable signals and starts generating read addresses for each RAM while continuing with the write process.

The address generator generates two read addresses successively with an increment of 6, in order to read 2 locations from one RAM during one read operation. The first column of RAM1 is read first, followed by the first column of RAM2 and this process continues until the last location is read[9] This technique enables us to write 3 bits to the buffer at the same time and read 3 bits successively to generate a 3-bit symbol for QPSK mapping.

C. Interleaver for 16-QAM Mapping

For 16-QAM mapping the interleaver is implemented using four memory blocks of double the required size. The method of read and write address generation is the same as explained before except the fact that now we have four separate RAMs that are first filled simultaneously and then the data is read out column-wise from RAM1, RAM2, RAM3, and RAM4 alternatley to implement the interleaver function. The address generator generates four read addresses successively with an increment of 3, in order to read 4 locations from one RAM during one read operation. This technique enables us to write 5 bits to the buffer at the same time[10] and read 5 bits successively to generate a 5-bit symbol for 16-QAM mapping.

IV. HARDWARE RESOURCE UTILIZATION.

A. Interleaver Memory Utilization

Table II shows the RAM resource utilization for the different types of modulation schemes.

<table>
<thead>
<tr>
<th>Mapping type</th>
<th>No. of Slices Used out of 5472</th>
<th>No. of 4 input LUTs out of 10944</th>
<th>No. of slice Flip flops out of 10944</th>
<th>Maximum Frequency (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>1025</td>
<td>1048</td>
<td>1188</td>
<td>44.592</td>
</tr>
<tr>
<td>QPSK</td>
<td>889</td>
<td>1226</td>
<td>981</td>
<td>42.592</td>
</tr>
<tr>
<td>16-QAM</td>
<td>1094</td>
<td>1323</td>
<td>1275</td>
<td>92.520</td>
</tr>
</tbody>
</table>

Each look-up table (LUT) is 32 bit and BRAM is 36 Kb, which can also be used in separate blocks of 18 Kb. As we can see, the implementation is very efficient in terms of RAM resource utilization if Distributed RAM extraction method is used during the synthesis of our design. However, if Auto RAM extraction is used, the synthesizer uses Block RAM resources to implement some of the memory for interleavers.
in order to save distributed RAM resources which results in an increase in the operating frequency of the overall system. In this implementation, most of the block RAM resources are wasted as the modulation size increases to 64-QAM.

From the table II, We can see that memory utilization is less for QPSK modulation scheme in terms of number of slices used in the look-up table [LUT] and 16-QAM is better in terms of frequency of operation.

V. CONCLUSION

In this paper, an efficient way to design the IEEE 802.16 transmitter on FPGA is presented. 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 time applications since they only require to replace the current interleaving parameters and do not involve any hardware alteration. The transmitter using different modulation schemes have been coded and stimulated and compares with respect to area, frequency and power utilizations.

REFERENCES


