DESIGN OF HIGH SPEED RECONFIGURABLE COPROCESSOR FOR INTERLEAVER AND DE-INTERLEAVER OPERATIONS

Mallikarjunaswamy S\textsuperscript{1}, Dr. Nataraj K.R\textsuperscript{2}, Balachandra P\textsuperscript{3}, Sharmila N.\textsuperscript{4}

\textsuperscript{1}Research Scholar Jain University, \\
\textsuperscript{2,3}SJB Institute of Technology, Bangalore-60, \\
\textsuperscript{4}R.N.S.Institute of Technology, Bangalore-98

ABSTRACT

In this paper we present the high speed reconfigurable co-processor for interleaver and de-interleaver system. Today’s communications systems have to perform in the midst of extreme signal, complex depth and diversity. With help of reconfigurable co-processor we can ensure the communications systems are ready for next generation operations. The communication system commonly uses an interleaver and de-interleaver technique for security purpose to overcome burst error such as correlated channel noise and authentication. The proposed reconfigurable co-processor for interleaver and de-interleaver is more accurate, reliable and flexible. The interleaver and de-interleaver coprocessor has been implemented using VHDL and the synthesis has been done by model-sim. The critical path is of about 0.16 nm technology and gate count of about 34,000. The Performance comparisons shows the Proposed co-processor can reduce the clock cycles for interleaving (8-bit=50%, 32-bit=81.45%), 67% de-interleaver for the IEEE 802.11a standard (12 Mbps data rate). and compared with existing DSP’s. From the results, the proposed co-processor show good performance compared to existing DSP chip in terms of number clock’s per cycle.

Keywords: Interleaver, De-interleaver, Binary symmetric channel (BSC) and length shift register sequence (LFSR)

INTRODUCTION

Digital communications are disposed to random correlated channel noise such as fading or burst errors. Interleaving is most commonly used for error correction in stream of codes to make
them operative in a burst noise location. The interleaver spreads out end-to-end data over numerous blocks of data. Any burst noise occurring will be reflected on the receiver side decoder due to the interleaver as self-regulating random symbol error which is more controllable than burst errors. Interleavers can be classified as either pseudo random or periodic. Block interleaving is an example for periodic interleaving, periodic interleaver instructs the data in a repeating sequence of bytes. This interleaver accepts the symbols in form of blocks and necessary operations perform on data. Periodic interleaving is most commonly invoked as it is easily accomplished in hardware. Pseudo random interleaver rearranges the data in a pseudo random sequence.

Interleaving and De-interleaving is a technique most commonly used in communication systems to overcome burst error such as correlated channel noise and authentication. The interleaver rearranges input data such that sequential data are spread out. At the receiver end, the interleaved data is organized back into the original data sequence by the de-interleaver. As a result of interleaving, correlated noise introduced in the transmission channel appears to be statistically independent at the receiver and thus allows better error correction. The interleaver arranged in the form of two-dimensional array, the data is read in along its row wise. Once the array is full, same data is read out along its column wise, thus changing the order of the data. (The interleaver process is denoted by the Greek letter \( \pi \) and de-interleaver process is denoted by \( \pi^{-1} \).) The interleaver process is illustrated in Figure 4, the original data order can be restored by a corresponding de-interleaver process.

The interleaver process is illustrated in Figure 1. The original data order can be restored by a corresponding de-interleaver process [1]. The de-interleaver arranged in the form of two-dimensional array. The data is read in along its column wise and read out by row wise. This interleaver may be present between the outer encoder and inner encoder at transmitter side there is two component codes uses a concatenated code. The de-interleaver between the inner decoder and outer decoder at receiver side, as shown in Figure 4. Due to this process the changing of some order of original data in order to interleaver arranged in the form of two-dimensional array, the data is read in along its row wise. Once the array is full, same data is read out along its column wise, thus changing the order of the data. (The interleaver process is denoted by the Greek letter \( \pi \) and de-interleaver process is denoted by \( \pi^{-1} \).)
codes uses a concatenated code. The de-interleaver between the inner decoder and outer decoder at receiver side, as shown in Figure 1, due to this process the changing of some order of original data in order to make extraction of the original signal difficult to the un-authorization person this process shown in Figure 2.

![Figure 2. Operations of Interleaver and De-Interleaver.](image)

A COMMUNICATION SYSTEM AND INTERLEAVER – FIRST ENCOUNTER

![Figure 3. Basic communication system model](image)

Figure 3 shows the basic communication system. It consists of an information source (sequence of messages or producing a message), transmitter (have a capability to operate on the message to maintain security, modulation process and convert it to a suitable signal for operations), a channel (a medium like wire or air etc.), a receiver (basically inverse of transmitter operations), and the destination (a machine or a person)[2]. The noise source can be defined as fading or burst errors noise (switch noise, lightening), channel distortion or white noise. In this paper deals only on burst noise. The different modulation takes place at transmitter side. For binary transmission we are using channel model termed as binary symmetric channel (BSC).
Figure 4 shows a basic binary symmetric channel (BSC) with transition probability $p$. It can be calculated by knowing the value of signal properties, amount of noise expected (i.e. probability distribution of noise) and the quantization thresholds of the modulator. With the encoding process, at receiver side the message bits are encoded in to code word $c$ with a probability $P(r|c)$ of the received vector $r$ [4].

**BLOCK DIAGRAM DESCRIPTION**

Based on different implementation features, the interleavers are divided in two main types named as convolutional interleaver and block interleaver. There are some other classifications based on their properties, which will also be discussed briefly in later sections. At transmitter side of the interleaver, the information coming from the encoder is in the form of blocks. The permutations are applied and then block of data is passed to two dimension array for further modulations. The normal implementation of a block interleaver is performed by Row-Column matrix of size $R \times C$, where $R$ is total number of rows and $C$ is the total number of columns.

Figure 5 shows the basic block diagram and performance of interleaving. In this illustration there is no inter-row or inter-column permutations are used, thus the interleaver is operated on some systematic order. The row-column block of interleaver gives the advantage that it isolates the burst errors of size $R$ by a distance $C$ [5]. If some situation a periodic single error sequence with period $R$ affects the transmission, at receiver side a de-interleaver produces a burst of errors with size $C$. In this situation further permutations need and it increase the implementation complexity.

![Figure 4: A simplified transmission and reception over a noisy channel](image)

![Figure 5: Simple block interleaving without permutations](image)
Figure 6. Block interleaving with row and column permutations

Figure 6 shows an illustration of an interleaver block with different row and column permutations is given in. The permutations are applied with certain format after the information has been written to the R x C row column matrix [6]. At receiver side the original data order can be restored by a corresponding de-interleaver process.

Other than row-column matrix construction, a block interleaver can also be constructed by using other methods. An example of generating the block interleaver is by defining a primitive polynomial to generate a maximal length shift register sequence called maximal length LFSR as shown in Figure 7. A maximal length LFSR generates all the addresses by cycling through all the possible states except the state where all bits are zero. The start of addressing sequence is defined by introducing a starting seed value. When block size is less than the complete cycle of the primitive polynomial, a pruning device has to be incorporated to discard the invalid indices. Based on different construction methodologies and properties, the block interleavers can further be divided in different sub-categories which will be described in a later section.

Figure 7 A maximal length LFSR with n=12
Figure 7 shows an interleaver module used for primitive polynomial to be generated by using maximal length shift register sequence is called maximal length LFSR. This module generates all physical addresses by cycling through the entire probable states apart from the state where all bits are zero [7][8]. The starting seed value defines the starting addresses sequence. If complete cycle is greater than the block size of the primitive polynomial, in such case a pruning device has to be incorporated to reject the invalid indices. Based on different construction properties and methodologies, the interleaver module can further be divided in different sub-categories.

CONVOLUTIONAL INTERLEAVER

Figure 8 shows a convolutional interleaver. It consists of X rows of shift registers, with different delay in each row. In general each consecutive row and columns has a delay which is N codes duration higher than the previous row. The encoder sends a code word symbols into the array of shift register, one code symbol to each row, every time each new code word symbols the commutator switches to a new shift registers and right most columns code word symbol come out to the channel. The \( i^{th} \) \((1 < i < X - 1)\) shift register has a length of \((i - 1) N\) levels where \(N = M/X\) and the last row require \(M - 1\) number of delay elements. The convolution de-interleaver executes the inverse operation of the interleaver and structure and delay elements are also different. Zeroth row of the interleaver becomes an \(X - 1\) row in the de-interleaver, 1st row of the pervious becomes \(N - 2\) row of later and so on.

![Convolutional Interleaver](image)

HARDWARE DESCRIPTION OF COPROCESSOR

In our experimentation, Xilinx Spartan-3 (device XC3S 400) with 400K gate count FPGA is used [9]. It has total 896 numbers of configurable logic blocks (CLBs) arranged in 32 X 28 matrix fashion. Each CLB has four slices and two of them are named as SLICEM and rest two as SLICEL. Each SLICEM can be used as 16 bit (embedded) shift register (SRL16). DAB application requires a convolutional interleaver of array size [6]; of \(17 \times j \) \((j = 0, 1, ..., 11) = 1122\) numbers of delay elements. Numbers of SRL 16 required to implement the interleaver is 77 which is only 4.3% of available SRL16. Because of our efficient FPGA implementation technique, sufficient FPGA resources are made available for implementing other circuitry of the transmitter / receiver [9].
PROPOSED MODEL OF CONVOLUTIONAL INTERLEVER

In our proposed work interleaver and de-interleaver coprocessor has been implemented by using VHDL and synthesis has been done by model-sim. The critical path of about 0.16 nm technology and gate count of about 34,000 FPGA is used [10]. It has total 900 configurable logic blocks (CLBs) arranged in 48x28 matrix. Each configurable logic block has four slices and digital audio broadcast application requires a convolution interleaver of array size [11].

A novel architecture of reconfigurable co-processor for interleaver and de-interleaver of an 8 bit convolutional interleaver with N=1 is shown in figure 9. The code word symbols (Xin) received in serial from an encoder is converted by the serial input parallel output (SIPO) register converts the code word symbols (Xin) received from an encoder into an 8 bit parallel code word. The SIPO output changes its value with respect to clock which is not necessary at the input of the delay input. The delay unit receives a word from the buffer unit after every 8 clock cycles. A buffer unit sends a code word to the delay unit. The delay unit is comprised of eight rows and is requiring a structure as narrated in figure 9. Each 8 bit code word of the code symbols is applied to the respective rows of the delay unit. In delay unit the code word is scrambled and the scrambled code word then applied to the 8:1 multiplexer (MUX) which converts it into stream of serial data (Xout). The SIPO register in an interleaver circuit is driven by the clock signal, a three bit counter and a clock circuit. The clock circuit divides the system clock frequency by 8 which in turn used to drive the delay and buffer unit. The 3-bit counter generates a select input for the 8:1 MUX. The proposed reconfigurable co-processor for de-interleaver is exactly similar to figure 9 but inverse operation. The scrambled code word is converted into its original code word at the de-interleaver output side.

Figure.9. Block diagram of proposed 8 bit convolutional interleaver.

Figure 10 shows the 8 bit input signal (11111111)2 is applied at data_in input of the interleaver at receiver side. The interleaver generates a scrambling output data at data_out. This scrambled output is applied as a input to the De-interleaver which rearranges them in such a way that the original code word is generated at output side(data_out).

CRITICAL ANALYSIS OF MODEL-SIM IMPLEMENTATION RESULTS

The novel architecture model of Convolutional interleaver de-interleaver pairs (both 8 bit, 32 bit and 64-bit) has been implemented by using VHDL and synthesis has been done by model-sim Xilinx Spartan-3 (Device: XC3S400) FPGA platform. Table I shows the comparison between proposed co-processor for interleave, de-interleaver and Star-Core SC140 the Proposed co-processor
can reduce the clock cycles for interleaver (8-bit=50%, 32-bit =81.45%) and 67 % de-interleaver for the IEEE 802.11a standard (12 Mbps data rate).

**Table I Performance Comparison for Interleaver and De-Interleaver**

<table>
<thead>
<tr>
<th>Interleaver word length</th>
<th>1 bit delay units required</th>
<th>SC140</th>
<th>Proposed system</th>
<th>Slice saving in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bit</td>
<td>8 X 7 = 56</td>
<td>56 + 2 = 28</td>
<td>14</td>
<td>50.00 %</td>
</tr>
<tr>
<td>32 bit</td>
<td>32 X 31 = 992</td>
<td>992 ÷ 2 = 496</td>
<td>92</td>
<td>81.45 %</td>
</tr>
</tbody>
</table>

**Figure 10.** Simulation result with input code word = 111111112

**CONCLUSIONS**

In this work, we have proved the design and the implementation of high speed reconfigurable Co-processor for interleaver and De-interleaver. The reconfigurable coprocessor that can support various communication standards and algorithms. Interleaver and De-interleaver influential conventional and scramble technique. Main objective of this paper is to improve the security purpose, overcome burst error such as correlated channel noise and authentication, to increases the redundant by using co-processor and perform many parallel operations without any overhead and performance of reconfigurable co-processor is compared with existing DSP (SC140). The interleaver and de-interleaver coprocessor has been implemented by using VHDL and synthesis has been done.
by model-sim. The critical path of about 0.16 nm technology and gate count of about 34,000. The performance comparisons show the proposed co-processor can reduce the clock cycles for interleaving (8-bit = 50%, 32-bit = 81.45%), 67% de-interleaver for the IEEE 802.11a standard (12 Mbps data rate) and compared with existing DSP’s. From the results, the proposed co-processor shows good performance compared to existing DSP chip in terms of number clock’s per cycle.

REFERENCES