Chapter 5

Design of High Speed Re-configurable Co-processor for Interleaver and De-interleaver Operations

5.1 Introduction

Interleaving is not an error-correcting code it is just a permutation procedure whose main goal is to separate neighbouring bits or blocks of bits. This process is essential for the performance of certain codes, like RS and turbo , so interleaving is not used alone but as an auxiliary component to other codes (as in the CD example above).

In general, interleaving uses a simple deterministic reordering procedure, like that illustration in the example below, which is employed to break up error bursts.

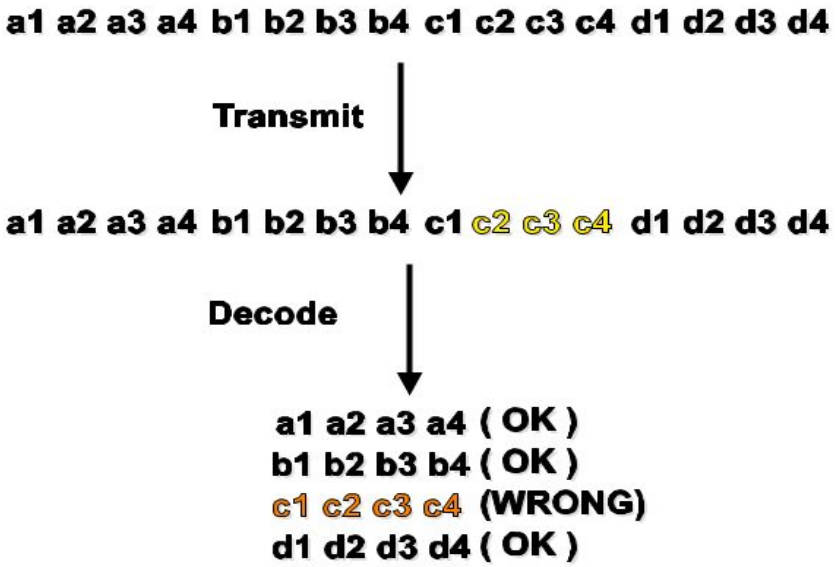

Figure 5.1 interleaving used for spreading error bursts (a) system without interleaving.

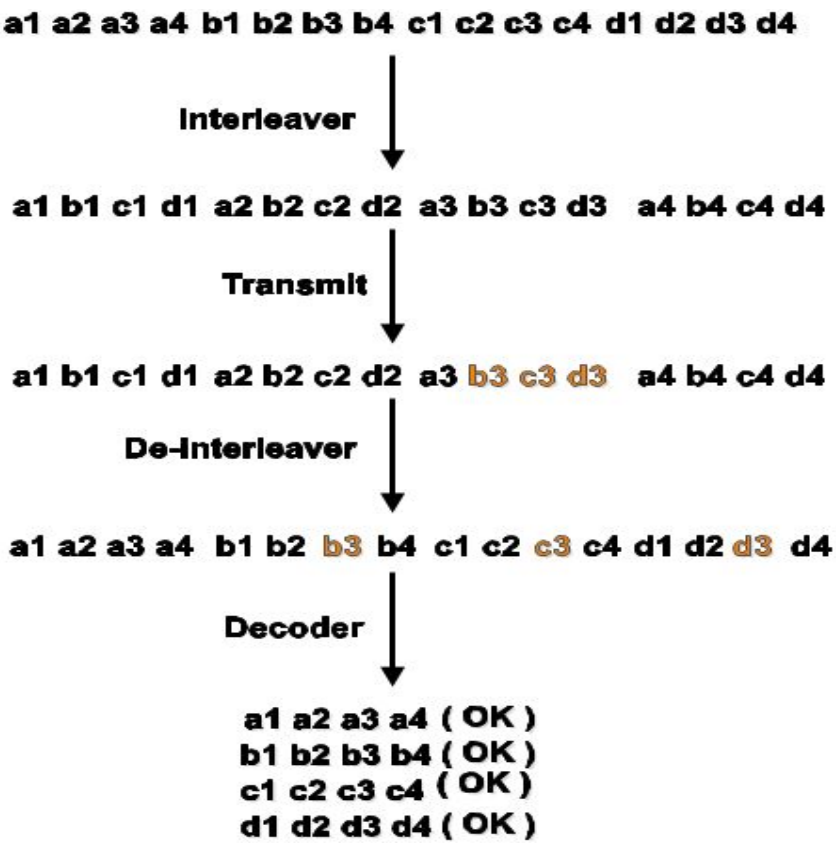

Figure 5.1 interleaving used for spreading error bursts (b) system with interleaving.

Consider an encoder that produces 4-bit codewords, which are transmitted over a communication channel to a corresponding decoder that is capable of correcting one error per codeword. This is illustrated in figure 5.1 (a), where a sequence of four 4-bit codewords is shown. Say that a 3-bit error burst occurs during transmission, corrupting the last three bits of codeword c (that is, c2c3c4).Due to the decoder's limitation, this codeword cannot be corrected. Consider now the situation depicted in figure 5.1 (b), where the same codewords are transmitted and the same error burst occurs. However, before transmission, interleaving is applied to the codewords. The consequence of this procedure is that the errors are spread by the de-interleaver, resulting in just one error per codeword, which can then be properly corrected by the decoder.

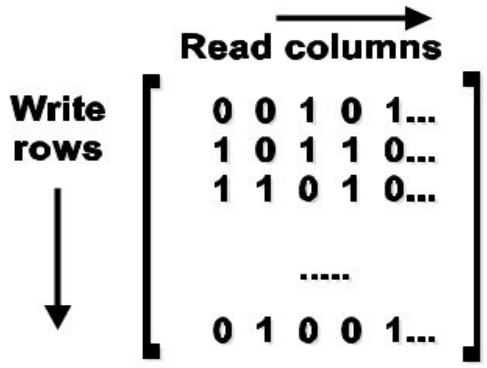

Figure 5.2 (a) Block

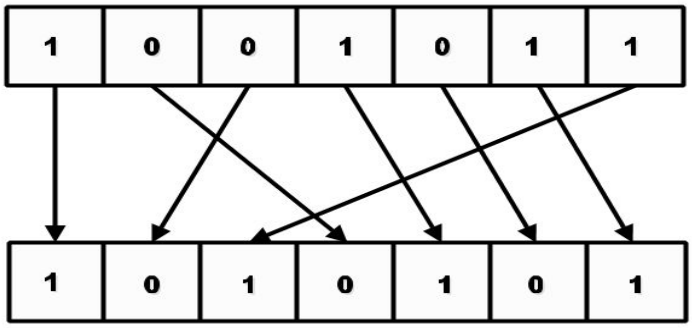

Figure 5.2 (b) pseudo-random interleavers

Two other types of interleavers are depicted in figure 5.2, .in figure 5.2 (a), a block interleaver is shown consisting of a two-dimensional memory array to which data are written vertically but is read from horizontally. In figure 5.2 (b), a pseudo-random interleaver is presented, which consists of rearranging the bits in a pseudo-random order. Note that the bits are not modified in either case, but they are just repositioned

Data interleaving is sometimes confused with data scrambling. The first difference between them is that the latter is a randomization procedure (performed pseudo-randomly), while the former can be deterministic (Figure 5.1). The second fundamental difference is that interleaving does not modify the data contents (it only modifies the bit positions).

5.2 A Communication System and Interleaver – First Encounter

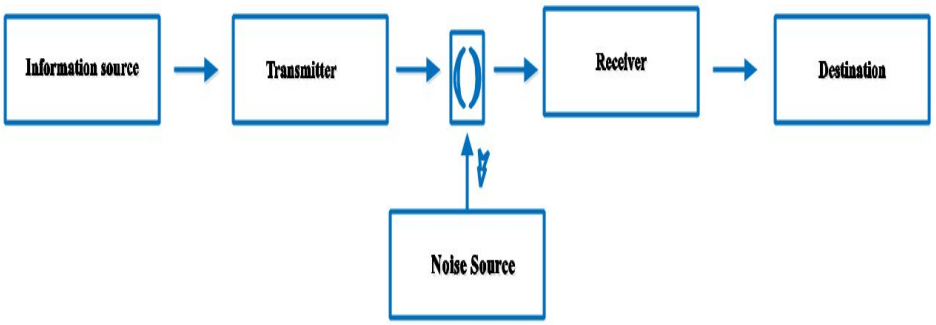

Figure.5.3. Basic communication system model

Figure 5.3 shows the basic communication system. It consists of an information source (sequence of messages or producing a message), transmitter it 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) . 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).

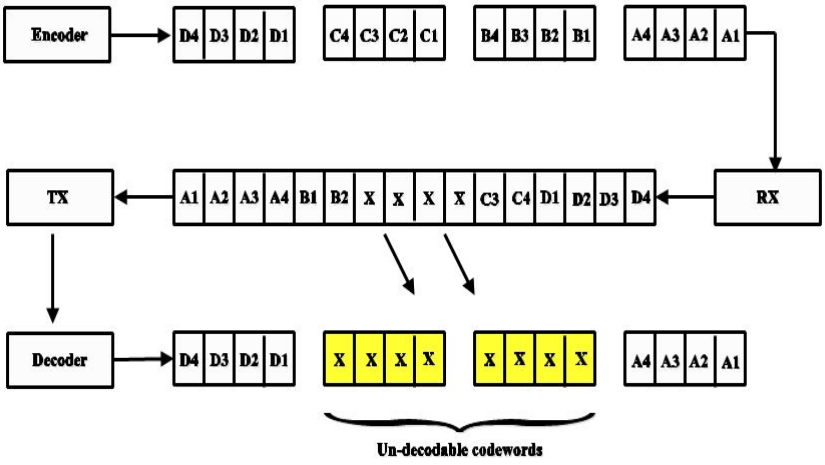

Figure.5.4. A simplified transmission and reception over a noisy channel

Figure 5.4 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 [34].

5.3 Block diagram description

Based on different implementation features, the interleavers are divided in two main types named as convolutional interleaver and block interleave. 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 x C, where C is total number of column and R is the total number of row.

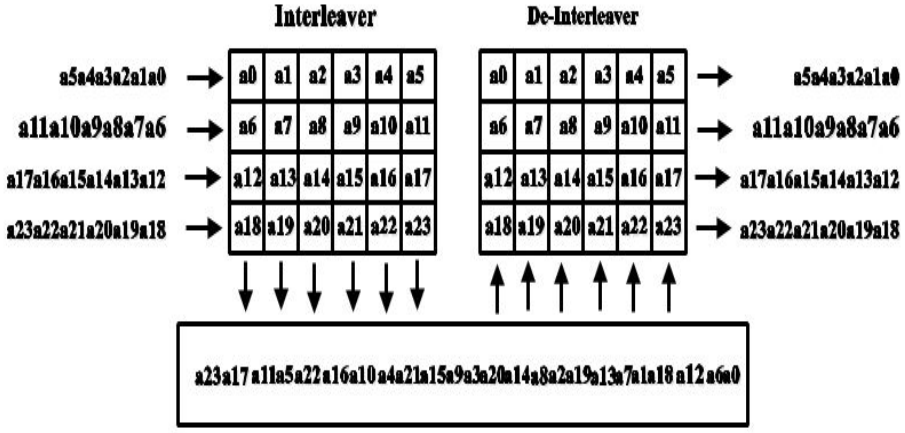

Figure.5.5. Simple block interleaving without permutations

Figure 5.5 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 [35]. 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.

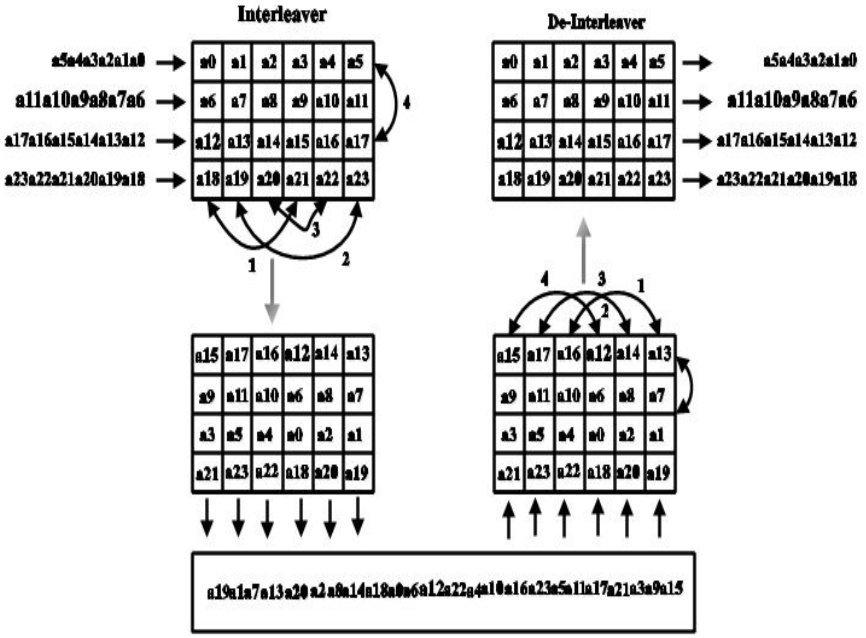

Figure.5.6. Block interleaving with row and column permutations.

Figure.5.6.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 [36]. A de-interleaver process can restore the order of original data on the receiver side.

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 5.7. 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.

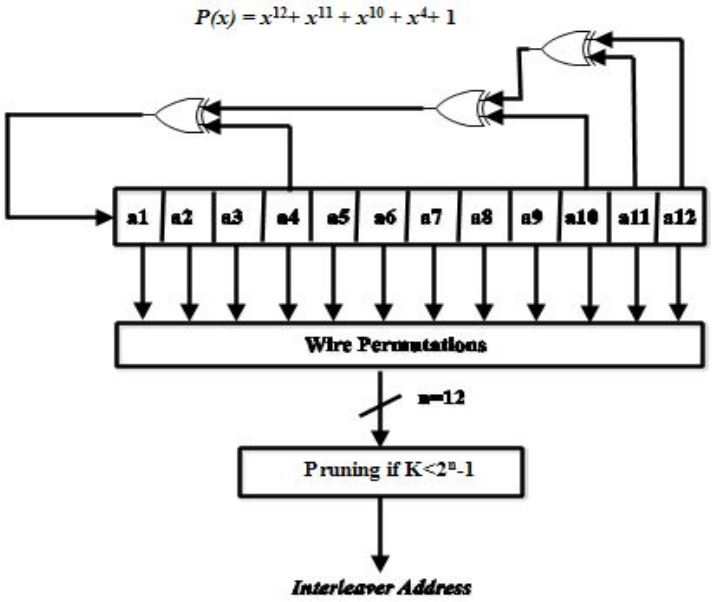

Figure 5.7 A maximal length LFSR with n=12

Figure 5.7 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 [37]. The starting seed value defines the staring addresses sequence. If complete cycle is greater than the block size of the primitive polynomial, in such case a pruning device has to be in-corporate to reject the invalid indices. Based on different construction properties and methodologies, the interleaver module can further be divided in different sub-categories.

5.4 Convolutional-Interleaver

Figure 5.8 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 .ever 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 ith ($1< i \leq X-1$) 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.

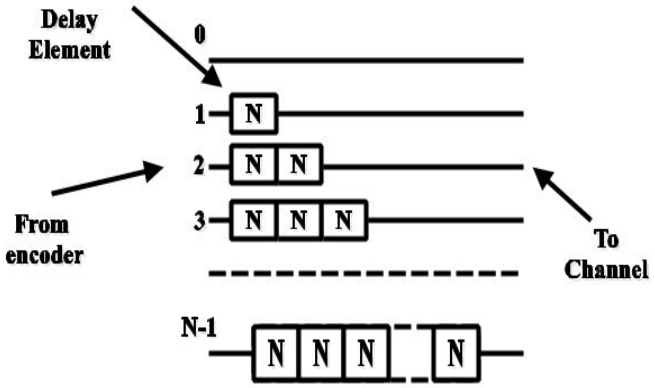

Figure.5.8. Convolutional-Interleaver

5.5 Hardware Description of Co-processor

In our experimentation, Xilinx Spartan-3 (device XC3S 400) with 400K gate count FPGA is used [38]. 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 as16 bit (embedded) shift register (SRL16). DAB application requires a convolutional interleaver of array size [39] of 17 X 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 [40].

5.6 Proposed Model of Convolutional Interleaver

In our proposed work interleaver and de-interleaver Co-processor 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 40000 FPGA is used [41].it has total 900 configurable logic blocks (CLBs) arranged in 48x28 matrix. Each configurable logic blocks has four slices and digital audio broadcast application requires a convolution interleaver of array size [42]

A novel architecture of Re-configurable co-processor for interleaver and de-interleaver of an 8 bit convolutional interleaver with N=1 is shown in Figure 5.9.

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 5.9. 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 Re-configurable 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.

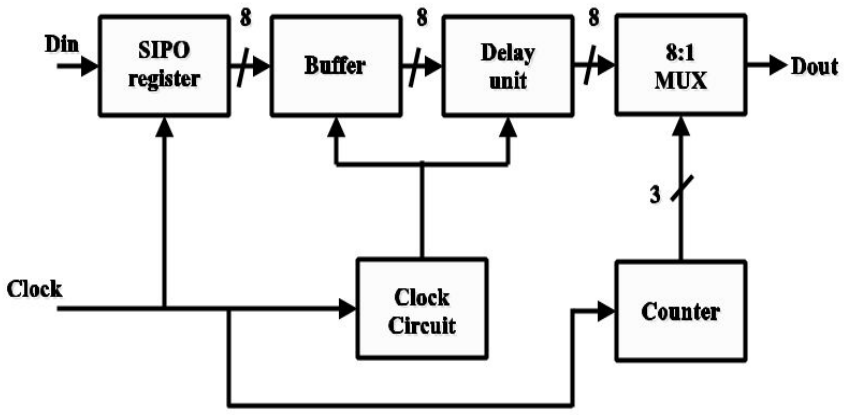

Figure.5.9. Block diagram of proposed 8 bit convolutional interleaver.

Figure 5.10 shows the 8 bit input signal (111111112) 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).

5.7 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 5.1 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 802.11N 2014 standards (data rate = 16Mbps).

Table 5.1 Performance Comparisons For Interleaver And De-Interleaver

Interleaver word length	1 bit delay units required	SC140	Proposed system	Slice saving in %
8 bit	8 X 7 = 56	56 ÷ 2 = 28	14	50.00 %
32 bit	32 X 31 = 992	992 ÷ 2 = 496	92	81.45 %

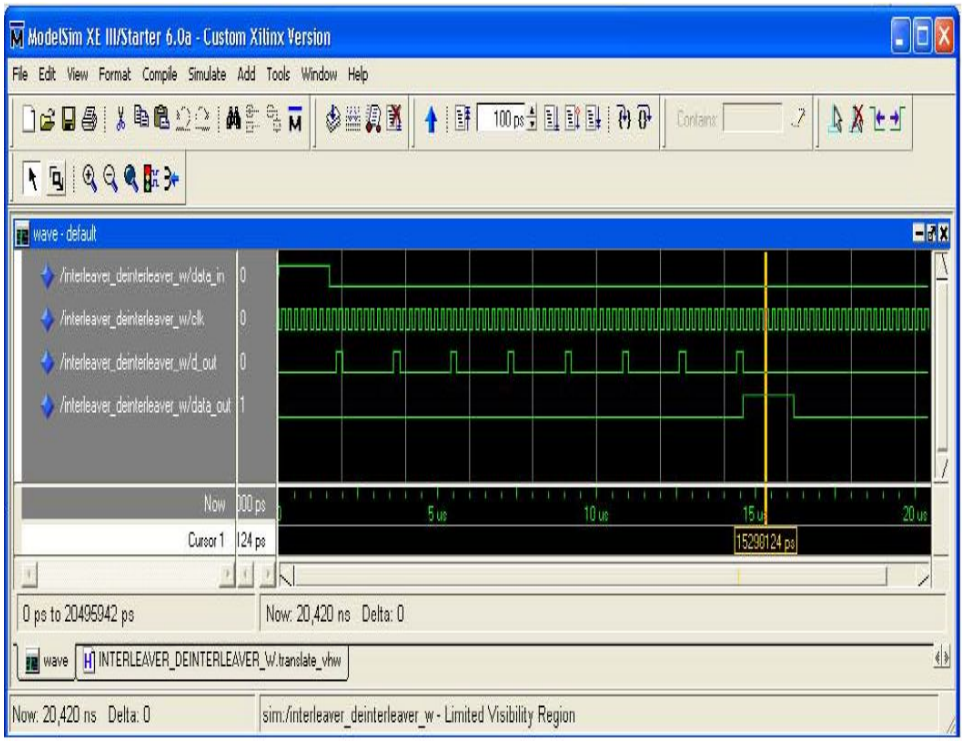

Figure 5.10. Simulation result with input code word = 111111112