Lookup Table Based 8B/10B Coding used in High Speed Serial Communication Application

[1]Allan Jose, Amala Avarachan
Department of Electronics and Communication Engineering
Amal Jyothi College of Engineering, Kanjirapally

***Abstract* — In this paper, a method for implementation of the DC balanced 8B/10B serializer and deserializer which employ ROM based lookup table is proposed. Serializer and deserializer blocks were designed separately. The serializer circuit gets the 8 bit data in parallel mode and delivers the 10 bit coded serialized data to deserializer and deserializer decodes the information and delivers the 8 bit parallel information. This method can be used by other high speed serial buses such as USB, PCI Express, IEEE 1394b, Serial ATA, SAS, Fiber Channel, SSA, Gigabit Ethernet, InfiniBand, XAUI, Serial RapidIO, DVI and HDMI that are done using encoding SerDes circuit. VHDL coding of the entire module are done in Xilinx ISE project navigator and simulated in Modelsim. Moreover, the proposed method has very low complexity and fast to execute with minimum logic and also easy to implement. Finally all of the blocks were combined together to have an integrated system and the whole circuit is implemented in FPGA Spartan-6 of Xilinx family.**

***Index Terms*—Serializer, deserializer, encoding, decoding**

I. INTRODUCTION

In day-today life, with the combination of serializer and deserializer, a large amount of parallel data can be transmitted through serial communication. Thus it helps to compensate for the inability in high speed limited I/O systems. For high speed communication network and computer links an appropriate transmission code is required. A free DC code or a code with a constant DC component has many advantages for electromagnetic wire links and fiber optics, which is the best suitable one.

In the year 1983, Al Widmer and Peter Franaszek introduced 8B/10B encoding in the IBM Journal of Research and Development. Along with the error detection technique, it helps to improve the transmission characteristics. In order to accomplish DC balance and bounded disparity, a line code that is 8B/10B code which maps 8 bit symbols to 10 bit symbols. It also helps to provide enough state changes which allow reasonable clock recovery. From this it is understandable that the difference between the counts of ones and zeros in a string of at least 20 bits is no more than two, and also that there are not more than five ones or zeros in a row. Thus the demand for the lower bandwidth limit of the channel which is necessary to transfer the signal gets reduced [1].

As the name suggests, the eight bits of data get transmitted as a 10 bit entity called a symbol or character. The lower five bits of data are encoded into a 6 bit group which is called the 5b/6b encoding; similarly the top three bits are encoded into a 4 bit group called the 3b/4b encoding. These code groups are concatenated together to form the 10 bit symbol which is transmitted on the transmission channel. The data symbols are often referred to as D.x.y where x ranges over 0–31 and y over 0–7. The 8b/10b encoding also define up to 12 special symbols or control characters that can be sent in place of a data symbol, which are often used to indicate start-of-frame, end-of-frame, link idle, skip and similar link level conditions. At least one of them is a comma symbol which needs to be used for defining the alignment of the 10 bit symbols. They are referred to as K.x.y [1].

The entire circuit is designed in VHDL by using Xilinx ISE project navigator and simulation is obtained using Modelsim.

II. SYSTEM ARCHITECTURE

The system consists of two main parts Serializer and Deserializer as shown in figure 1. An 8 bit signal is received by serializer circuit and encodes it into 10 bit data by using 8B/10B encoder. This 10 bit encoded data is sent to 10 bit parallel to serial block and the block delivers a 1 bit serialized signal. On the other hand, this 1 bit serial signal is sent to the deserializer circuit which consists of serial to parallel block and delivers 10 bit parallel signal. This 10 bit signal is then decoded using 10B/8B Decoder. Finally the same 8 bit signal which was sent to the serializer is received at the output of deserializer block. This has been used for high speed communication purposes.

A. Serializer

A serializer mainly consists of 8B/10 encoder and 10 bit parallel to serial converter. This block converts the 8 bit data into 10 bit symbols and this 10 bit is converted into serial form with the help of parallel to serial converter to transmit through the channel. This logic uses a clock which helps to serially clock the packets out.

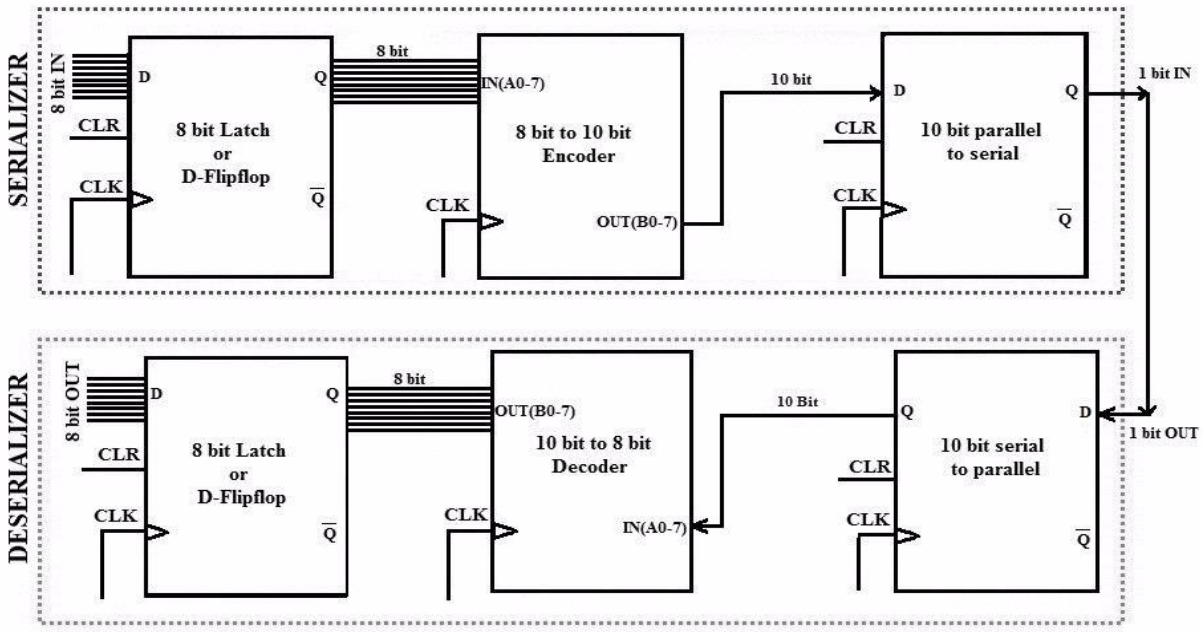

Fig. 1. Simplified System Architecture

i. 8B/10B Encoder

The encoder block helps to encode each byte of input data into 10 bits symbol. As per the design encoder module consists of input data, clock, data D/K as the inputs and the acquired outputs are 10 bit output symbol and running disparity. The incoming data is either a data or a control signals and each byte is encoded to 10 bits symbol. Figure 2 shows the input output signals of encoder block.

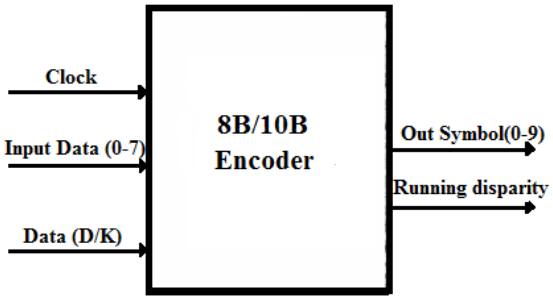

Fig. 2. I/0 Signals of Encoder Block

The entire block diagram of an encoder is shown in figure 3. A 1024 x 11 bit ROM is used to store the entire coding table of encoding. This means that it will take only 2 kilobyte of memory to store the entire code words. Both 5B/6B and 3B/4B tables are programmed in one memory in a form of one 8B/10B coding. The output consists of 11 bits and the MSB bit will be the running disparity of the next data. The remaining 10 bit will be the encoded output as per the input. The 10 bit address for the memory location is obtained by 8 bit input data, 1 bit data D/K and 1 bit previous running disparity one.

Each code word is used as the input address of the ROM and this location consist of the encoded 10 bit and the running disparity. This will help in increasing the speed of 8B/10B encoding.

The VHDL coding for the 8B/10B encoder was designed in Xilinx ISE project navigator and simulated in Modelsim. The 8B/10B encoder circuit was tested in FPGA Spartan-6 of Xilinx family.

Fig. 3. Block Diagram of Encoder

ii. Parallel To Serial Converter

This module is used to convert the 10 bit encoded parallel data into 1 bit serial data. It has a parallel clock input, a set of data input lines and input data latches. It may use an internal and external phase-locked loop to multiply the incoming parallel clock up to the serial frequency. The simplest form it has a shift register that receives parallel data once per parallel clock, and shift it out at the higher serial clock rate.

B. Deserializer

A deserializer mainly consists of 10B/8B decoder and 10 bit serial to parallel converter. The deserializer uses the reference clock to monitor the recovered clock from the bit stream. A serial to parallel converter helps to convert the serial signal into 10 bit parallel signal.

10B/8B decoder is a vital part in deserializer circuit. It receives the 10 bit signal from the output of serial to parallel converter. This is then decoded into 8 bit signal which is the same signal that is sent to the serializer circuit.

i. Serial To Parallel Converter

This module is used to generate 10 bit data from the incoming serial data. It has a receive clock output, a set of data output lines and output data latches.

ii. 10B/8B Decoder

The decoder part receives a 10 bits symbol which was transmitted by the encoder. As per the design decoder mainly consists of clock and input symbol as inputs and the obtained outputs are running disparity and output symbol. The byte clock used in decoder module is same as that of which is used in encoder module. After decoding, the output of this module has a 1 byte decoded data. Figure 4 displays the input output signals of decoder block.

Decoding of 268 unique transmitted characters should be taken place at the decode circuit .That is 256 byte values for data characters and 12 special control characters. Decoder tracks running disparity to verify that the disparity sequence of the received symbols is valid.

Choice of a LUT-based implementation, which uses FPGA slices, or a block-memory based implementation that uses a dedicated on-chip block memory block.

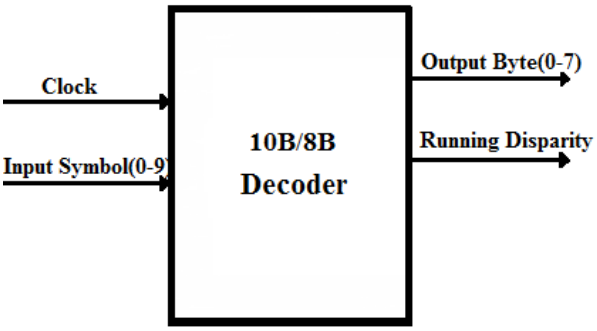

Fig. 4. I/O Signals of Decoder Block

Figure 5 illustrates the implementation of decoder in this design. It is necessary to save all the code words and related data in the memory block of the decoder. That is the decoder needs to have the 8B/10B table as well. For this purpose a 2 kilobyte ROM is used. In this design the incoming symbols are used to address the memory. In the ROM, a 1024 x 9 bit is used to store the entire coding. With the 10 bit address as the input which is getting from the encoder, the output can be directly obtained by reading the memory location. More than the decoded data and the Data/Control signal, the decoder error can be obtained if an invalid encoded output is obtained as the address of decoder. The running disparity error happens if the current running disparity is not matched with the expected running disparity and indicates that the running disparity is changed through the line. The symbol error happens when an invalid symbol is received.

The decoder is fully synchronous to its appropriate clock. All Decoder input ports have their setup time referenced to the rising edge of the CLK input. All Decoder outputs are also synchronous to their respective clock input.

The VHDL coding for the 8B/10B decode was designed in Xilinx ISE project navigator and simulated in Modelsim. The 8B/10B decoder circuit was tested in FPGA Xilinx Spartan-6.

Fig. 5. Block Diagram of Decoder

III. OPTIMIZATION

With the use of 8b/10b encoding, it is necessary to acquire high speed buses which is used to obtain higher speeds, that is about 5 gigabit per second. For this purpose a good idea of design and implementation is required. This must be improved, so it is necessary to apply the best devices and techniques which support the high frequency of bus clock. In addition to look-up table techniques, Spartan-6 which has the best speed of routing between logic blocks has been used. It is a FPGA of Xilinx Company which facilitates to achieve the above goal. Look up table will be time consuming if it is not used in a typically way. If each codeword is used as an address space and the memory space where the address points to involve the decoded word or vice versa, acquisition time could be so remarkable.

As compared to the IBM circuit which was developed by Al Widmer and Peter Franaszek the timing provided by the look up table method is impressive because only the reading process from the memory is taking place. In the IBM circuit for the implementation of encoder and decoder with emitter coupled logic required a total of 380 logic gate with a gate delay of one-tenth of a byte-clock interval. The logic space required for the FPGA implementation of look up table is less as compared to that of IBM circuit. The power dissipation and chip area for both the circuit is comparatively same.

From these results, the look up table which provide good timing with minimum logic space and complexity in the design has been obtained. Today the memories are becoming so fast, thus the above goal is being achieved.

IV. SIMULATION REULTS

Figure 6 and Figure 7 displays the RTL schematic of encoder and decoder that are synthesized using Xilinx ISE Project navigator.

In the following figure 8 the simulation result of an 8B/10B encoder is shown. The 8 bit inputs are ai, bi, ci, di, ei, fi, gi and hi where the input is provided as 11111111. The output bits are ao, bo, co, do, eo, io, fo, go, ho and jo. The output is obtained according to the encoding table and the output obtained is 1010110001. Ki is the control signal it can be set 0 or 1 according to the data and control symbols. A clock is to be provided which is supplied by using sbyteclk. Reset can be provided though reset in the simulation.

Figure 9 and 10 shows the simulation result of parallel to serial converter and serial to parallel converter respectively. Clock, enable and reset are given as clk, en and rst. Parallel data is given through din and the serial data can be obtained through dout. Serial input's the serial data and parallel data is obtained in parallel form.

In figure 11 the simulation result of a 10B/8B decoder is shown. The input 10 bits are ai, bi, ci, di, ei, ii fi, gi, hi and ji and the input provided as 101011000. The output will be 8 bit which are ao, bo, co, do, eo, fo, go and ho. The output of the decoder will be the same signal that is fed into an encoder which is 11111111. Thus the same input signal is received at the decoder and the desired output is obtained for the decoder circuit.

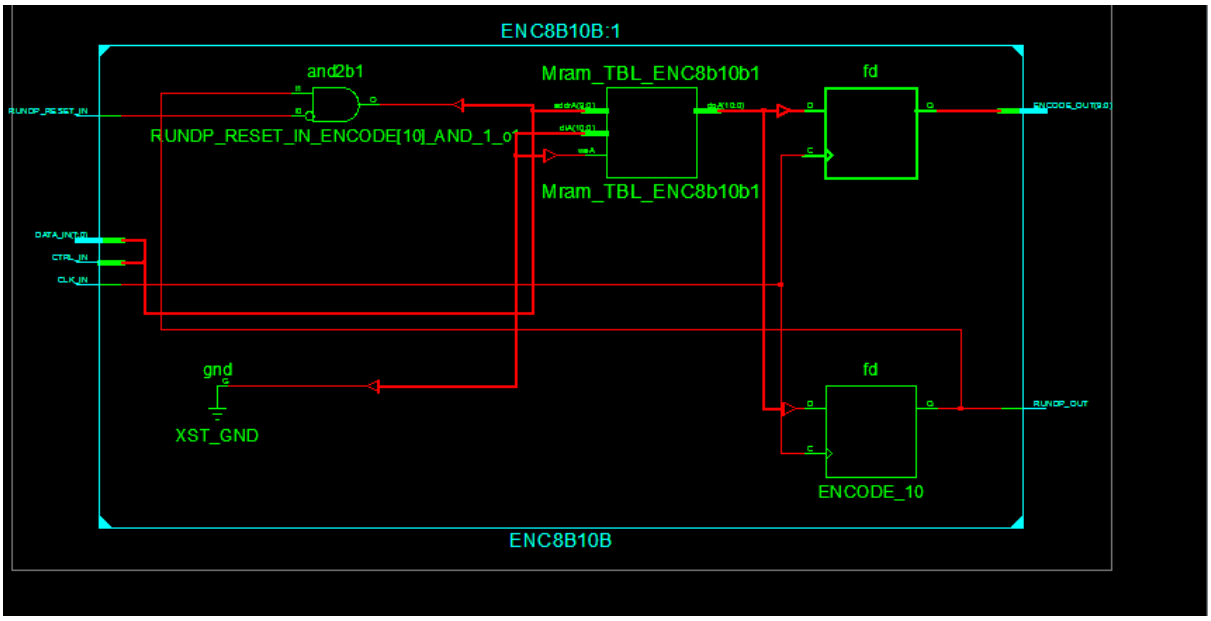

Fig. 6. RTL Schematic of Encoder

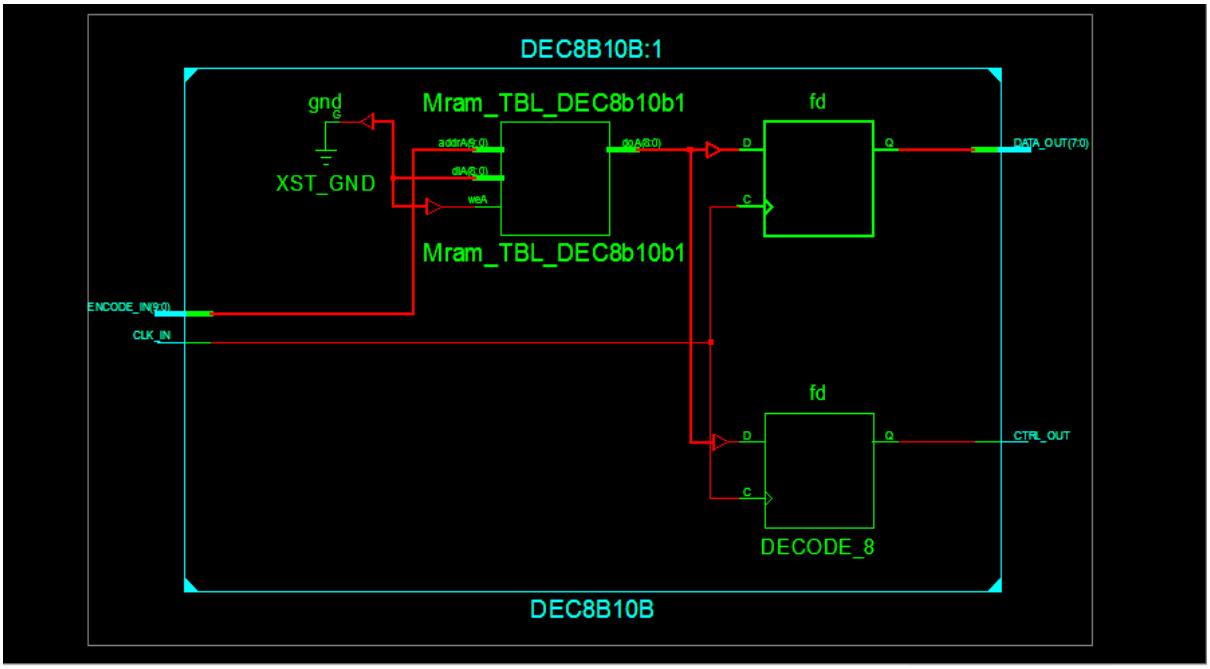

Fig. 7. RTL Schematic of Decoder

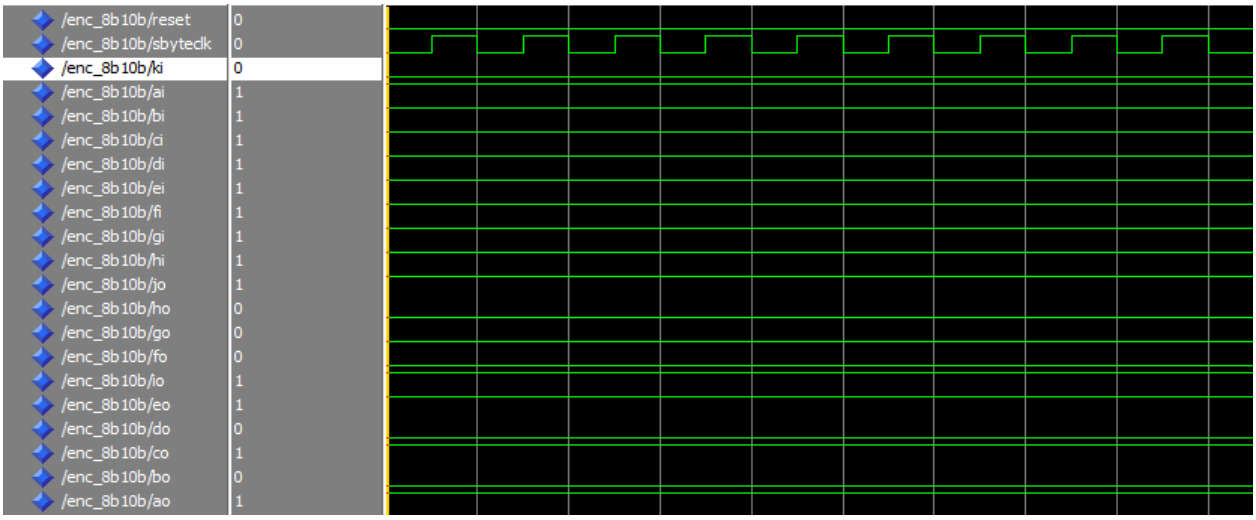

Fig. 8. 8B/10B Encoder Simulation Results

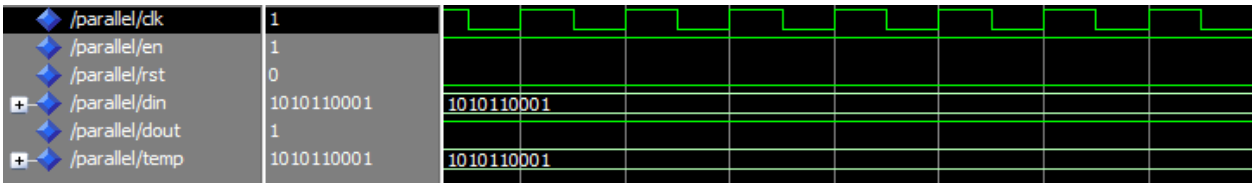

Fig. 9. Parallel to Serial Converter Simulation Results

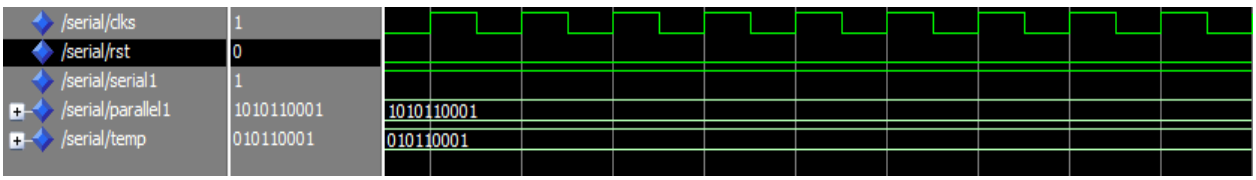

Fig. 10. Serial to Parallel Converter Simulation Results

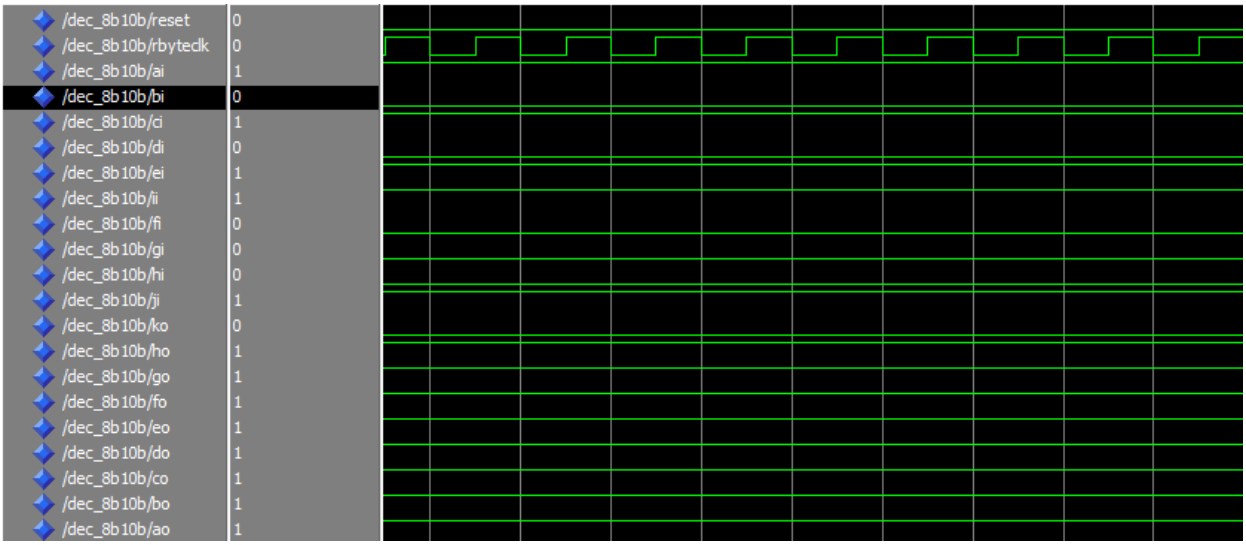

Fig. 11. 10B/8B Decoder Simulation Results

IV. CONCLUSION

In this paper, serializer and deserializer circuit was presented. The combined look-up table technique and Xilinx Spartan-6 technology to implement 8b/10b coding for high speed serial communication application make this encoding more powerful. Moreover, the proposed method has very low complexity and fast to execute with minimum logic and also easy to implement.

ACKNOWLEDGMENT

First of all we sincerely thank God the Almighty who is the most efficient and merciful for giving us knowledge and courage to complete the project successfully.

We derive immense pleasure in expressing our sincere thanks to the Principal Rev. Dr. Jose Kannampuzha, for his permission and infrastructural facilities for the successful completion of our project.

We extend our sincere gratitude to Prof. K G Satheeshkumar, Head of the department for his encouragements and motivation during our project.

We express our heartfelt gratefulness to Mr. Binu C Pillai, Assistant Professor, Department of ECE, project guide, for his valuable guidance and suggestion during the project.

REFERENCES

[1] A.X.Widmer and P.A. Franaszek, "*A DC-balanced, partitioned block, 8B/10B transmission code*, "IBM Journal of Research and Development, vol.27, no.5, pp. 440-451, Sep 1983.

[2] Mohammad Maadi," *An 8b/10b Encoding Serializer/Deserializer (SerDes) Circuit for High Speed Communication Applications Using a DC Balanced, Partitioned Block 8b/10b Transmission Code*", International Journal of Electronics and Electrical Engineering Vol. 3, No. 2, April, 2015.

[3] Universal Serial Bus 3.0 Specification Revision 0.9, July 30, 2008

[4] Ravi Budruk, Don Anderson & Tom Shanely*," PCI Express System Architecture–Mindshare Inc*", pp 419-434, 2004.