An 8b/10b Encoding Serializer/Deserializer (SerDes) Circuit for High Speed Communication Applications Using a DC Balanced, Partitioned-Block, 8b/10b Transmission Code

Mohammad Maadi
Middle East Technical University, Department of Electrical and Electronics Engineering, 06531, Ankara, Turkey
Email: mohammad.maadi@metu.edu.tr

***Abstract*—In this paper, an 8b/10b encoding serializer/deserializer (SerDes) circuit using a DC-balanced, partitioned block, 8b/10b transmission code was presented. The information format of this transmission code consists of packets which are variable in length and can be suitable for high speed applications. 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. Except serializer circuit which was designed using Cadence in 0.6 μm CMOS technology, all of the blocks were designed in Verilog and VerilogXL in XILINX. Finally all of the blocks were combined together to have an integrated system.**

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

I. Introduction

For transferring huge amounts of bits in parallel mode, the combination of serializer/deserializer (SerDes) circuits are commonly used to compensate for the inability in high speed limited I/O systems. A typical serializer circuit converts the parallel n-bit data into a faster 1-bit signal. Encoding SerDes circuits can be used in some popular applications like DVI/HDMI, Serial ATA, USB 3.0, Gigabit Ethernet, PCI Express and some interface electronics [1]-[4].

For high speed communication networks and computer links, a suitable transmission code is needed. A free DC code or a code with a constant DC component [5] has many advantages for electromagnetic wire links and fiber optic [6].

In this paper an 8b/10b encoding serializer/deserializer (SerDes) circuit has been designed for high speed communication applications.

II. System Architecture

According to Fig. 1, our system consists of two major parts; serializer and deserializer. Serializer circuit is supposed to receive 8-bit signal at 20MHz clock frequency and encode it into 10-bit code at the same clock. This 10-bit encoded data is sent to 10-bit parallel to serial block and the block delivers a 1-bit serialized signal at frequencies in the range of 200-800MHz's. On the other hand the 1-bit serialized signal is sent to 10-bit serial to parallel block in deserializer part and then data is decoded using 10b-8b decoder and finally the same 8-bit signal which was sent to serializer, is received at the output of deserializer block.

A. Design Procedure

Except 10-bit parallel to serial converter, the complete structure (shown in Fig. 1) has been designed using Verilog (XILINX 10.1 and CADENCE). The 10-bit parallel to serial converter has been designed in Cadence from schematic to layout extraction and post-layout simulation. Since we want to integrate the Verilog codes and the blocks of the layout, the Verilog codes were implemented in Cadence using VerilogXL as well. All of needed clock frequencies and the adjusting of synchronization between these clocks (serializer and deserializer) have been implemented separately by different Verilog codes.

In designing of 10-bit parallel to serial converter circuit, we tried to consider many critical performance parameters such as die area, power dissipation, latency and throughput.

III. 8b/10b Encoder and 10b/8b Decoder

A. 8b/10b Encoder Program

As we mentioned before, except high speed serializer circuit, rest blocks have been designed using VERILOG code in Xilinx software. Since we need a coding format which is suitable for high speed communication applications, a DC balanced code with equal number of one and zeros, partitioned-block and 8b/10b transmission code has been used [6].

Before writing any program in VERILOG, the input and output ports should be defined clearly. Eleven input and output pins were defined for input and output ports separately. The coding program was written in two ways.

Manuscript received December 18, 2013; revised May 15, 2014.

doi: 10.12720/ijeee.3.2.144-148

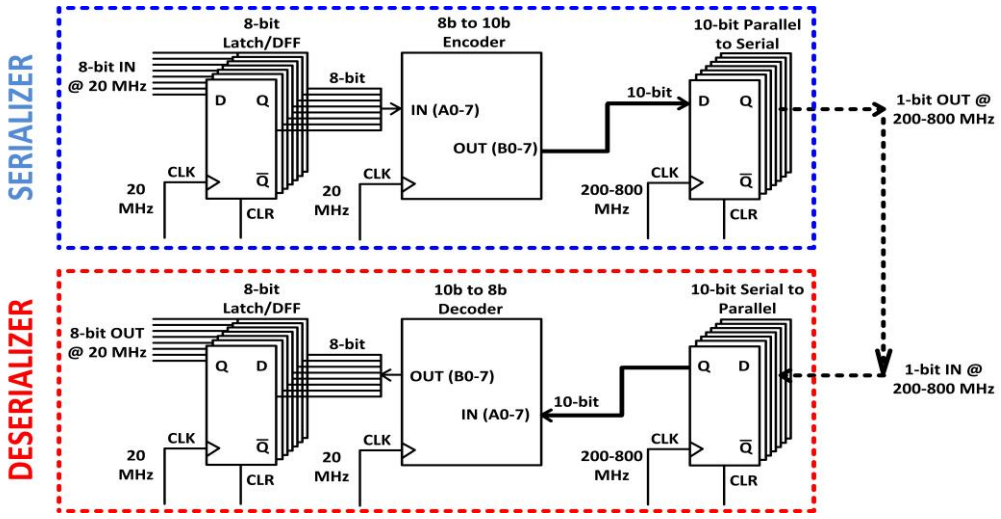

Figure 1. Simplified system architecture.

In first one, the data-in and data-out are defined with separated pins which can be connected to other circuits one by one. In second method the data-in and data-out are defined by 8b and 10b buses, respectively.

According to encoding tables [6], encoder receives 8-bit input code and converts them to 10b. Each eight bit consists of two parts; first five bits which should be coded to six bits and second three bits which must be coded to four bits. In final part, these coded information should be united as an integrated 10b.

Some logic treatments can be found between input and output codes [6]. In main body of the written program, all the possible input codes were defined. If we look at the input codes [6], we found that first four bits of five bits are common in some of them, so they can be defined in fewer lines in the program. For example, in 00000 and 00001, the first four bits are zero.

L_{xy} can help us to define the codes using the number of their ones and zeros in first four bits, where the x is the number of ones and y is the number of zeros in first four input bits. As a sample, the complete first four definitions are given in Table I.

B. Encoder Program Simulation Results in Xilinx

By following conversion rules of the codes, the VERILOG and VERILOGXL codes were written in Xilinx and CADENCE, respectively. Since the design rules of the conversion method are complex, we need to write the program precisely and the conversion rules should be classified in various packages.

For program simulation in Xilinx, we need to write the test bench of the program. Since the encoder clock frequency is 20MHz, the clock period of the program is defined 50ns. After 25ns, state of the clock is reversed. In written test bench, some extra lines have been added to show the results both in data and pulse modes. The program was simulated in 1 μs.

Fig. 2 shows the 8b/10b encoding simulation results during 800ns. As Fig. 2 shows, the encoder gets the 8-bit input data from datain [8:0] port and gives the coded data from dataout [9:0] port. For example, whenever the encoder receives 00010110, it generates 1101010110 at its output. Table II shows the complete results of the designed 8b/10b encoder in the period of 1 μs.

TABLE I. THE COMPLETE FIRST FOUR DEFINITIONS OF CODES.

Definition in VERILOG L_{xy}	CODES A–B–C–D ain–bin–cin-din	Number of 1's	Number of 0's
L22	0011 1100 1010 0101 0110 1001	2	2
40	1111	4	0
L04	0000	0	4
L13	1000 0100 0010 0001	1	3
L31	0111 1011 1101 1110	3	1

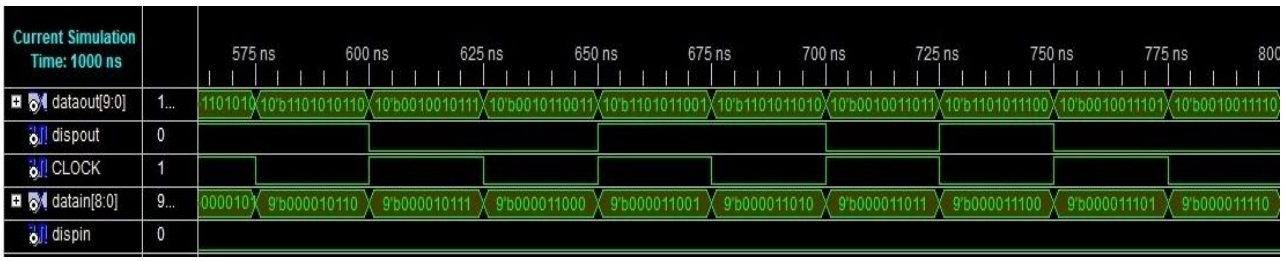

Figure 2. The 8b/10b encoding simulation results during 800 ns.

TABLE II. THE COMPLETE RESULTS OF DESIGNED 8B/10B ENCODER IN 1 μS.

Time	CLOCK	datain	dispin	dataout	dispout
0	0	000000000	0	0010111001	0
25	1	101000011	0	1010100011	1
50	1	000000001	0	0010101110	0
75	0	000000010	0	0010101101	0
100	1	000000011	0	1101100011	1
125	0	000000100	0	0010101011	0
150	1	000000101	0	1101100101	1
175	0	000000110	0	1101100110	1
200	1	000000111	0	1101000111	1
225	0	000001000	0	0010100111	0
250	1	000001001	0	1101101001	1
275	0	000001010	0	1101101010	1
300	1	000001011	0	1101001011	1
325	0	000001100	0	1101101100	1
350	1	000001101	0	1101001101	1
375	0	000001110	0	1101001110	1
400	1	000001111	0	0010111010	0
425	0	000010000	0	0010110110	0
450	1	000010001	0	1101110001	1
475	0	000010010	0	1101110010	1
500	1	000010011	0	1101010011	1
525	0	000010100	0	1101110100	1
550	1	000010101	0	1101010101	1
575	0	000010110	0	1101010110	1
600	1	000010111	0	0010010111	0
625	0	000011000	0	0010110011	0
650	1	000011001	0	1101011001	1
675	0	000011010	0	1101011010	1
700	1	000011011	0	0010011011	0
725	0	000011100	0	1101011100	1
750	1	000011101	0	0010011101	0
775	0	000011110	0	0010011110	0
800	1	000011111	0	0010110101	0
825	0	000100000	0	1001111001	1
850	1	000100001	0	1001101110	1
875	0	000100010	0	1001101101	1
900	1	000100011	0	1001100011	0
925	0	000100100	0	1001101011	1
950	1	000100101	0	1001100101	0
975	0	000100110	0	1001100110	0
1000	1	000100111	0	1001000111	0

C. Decoder Program Simulation Results in Xilinx

Since the decoder clock frequency is 20MHz, the clock period of the program is defined 50ns. After 25ns, state of the clock is reversed. In written test bench, some extra lines have been added to show the results both in data and pulse modes. The program was simulated in 1 μs.

Fig. 3 shows the 10b/8b decoding simulation results. As Fig. 3 shows, the decoder gets the 10-bit input data from datain [9:0] port and gives the coded data from dataout [8:0] port. Table III shows the complete results of the designed 10b/8b decoder in the period of 1 μs.

TABLE III. THE COMPLETE RESULTS OF DESIGNED 10B/8B DECODER IN 1 μS.

Time	CLOCK	datain	dispin	dataout	dispout
0	0	0000000000	0	101011111	z
25	1	0010010101	0	000010101	z
50	1	0000000001	0	101011110	z
75	0	0000000010	0	101011101	z
100	1	0000000011	0	101011100	z
125	0	0000000100	0	000111011	z
150	1	0000000101	0	000101111	z
175	0	0000000110	0	000100000	z
200	1	0000000111	0	000100111	z
225	0	0000001000	0	000110111	z
250	1	0000001001	0	000110000	z
275	0	0000001010	0	000111111	z
300	1	0000001011	0	000101011	z
325	0	0000001100	0	000111000	z
350	1	0000001101	0	000101101	z
375	0	0000001110	0	000101110	z
400	1	0000001111	0	000101111	z
425	0	0000010000	0	000110000	z
450	1	0000010001	0	000100001	z
475	0	0000010010	0	000100010	z
500	1	0000010011	0	000110011	z
525	0	0000010100	0	000100100	z
550	1	0000010101	0	000110101	z
575	0	0000010110	0	000110110	z
600	1	0000010111	0	100110111	z
625	0	0000011000	0	000101000	z
650	1	0000011001	0	000111001	z
675	0	0000011010	0	000111010	z
700	1	0000011011	0	100111011	z
725	0	0000011100	0	000111100	z
750	1	0000011101	0	100111101	z
775	0	0000011110	0	100111110	z
800	1	0000011111	0	000111111	z
825	0	0000100000	0	000100000	z
850	1	0000100001	0	000111110	z
875	0	0000100010	0	000111101	z
900	1	0000100011	0	000100011	z
925	0	0000100100	0	000111011	z
950	1	0000100101	0	000100101	z
975	0	0000100110	0	000100110	z
1000	1	0000100111	0	000101000	z

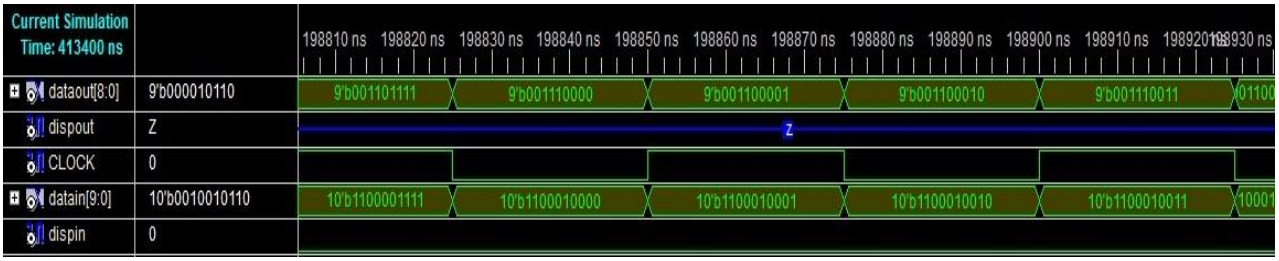

Figure 3. The 10b/8b decoding simulation results.

After testing the blocks and getting the correct results using VERILOG codes in Xilinx, some modifications were applied so that the written codes are compatible with Cadence VERILOGXL program. Except high speed serializer circuit, all the rest blocks were designed using VERILOGXL and their symbols were generated so that they can be used in our schematics in Cadence.

IV. 8B/10B Encoder and 10B/8B Decoder

Our designed 10-bit serializer is supposed to receive an 8-bit signal at 20MHz, encode it into 10-bit code at the same clock frequency, and deliver a serialized 1-bit signal at 200-800 MHz's. Fig. 4 shows the designed serializer which is able to work in DDR mode.

Designed Serializer circuit consists of two main rows. In each row there are five DFF's (High speed) and some combinational logics. In fact, each row is a shift register with parallel load. Each row receives five bits data in parallel mode and delivers them to output via a 2X1 multiplexer.

The period of the clock is 5ns and the circuit should be able to serialize 10b data in arbitrary frequency. In fact, the speed is the most important challenge in this design which should be solved using some essential factors like designing of proper logic gates, exact timing and etc. If the timing problems have not been solved correctly, some parts of the serialized data will be lost.

According to Fig. 4, the two rows of the serializer circuit work in different clock states. When one bit data is loaded to one of them, the second one becomes ready to send data and vice versa. The clocks of two lines are separated by one NOT gate so that DFF's of the two rows can work in opposite cycles. For each row, one single line with the name of $\overline{WRITE}$ / *SHIFT* has been considered. When this line is zero, the data is written to related row and when this line is one, the data is shifted to next stage. Two rows are connected via a multiplexer. When the signal of this multiplexer is zero, it transfers the data of first line and when the selection port of this multiplexer is one, it transfers the second line's data.

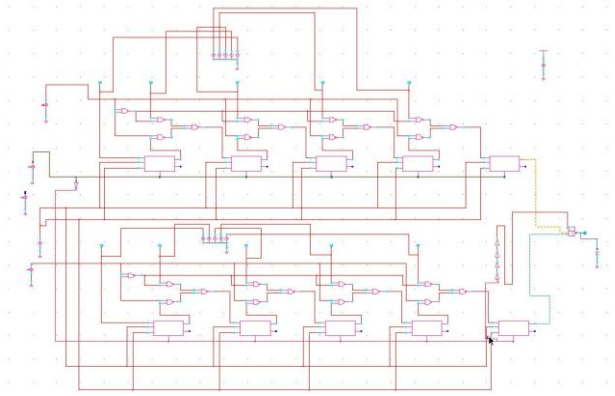

Figure 4. The 10b/8b decoding simulation results.

A. 10-bit Serializer Simulation Results

As shown in Fig. 5, 1011001100 code has been loaded to serializer input lines in parallel mode and we received this code at its output in serial mode.

The entire process of our designed blocks has been checked by sending many codes and fortunately we could receive the codes without any lost bit.

Fig. 6 shows the layout of the designed circuit in 0.6 µm CMOS technology in Cadence. All the steps such as DRC, layout extraction and post layout simulation have been implemented successfully. Table IV summarizes the results of the designed circuit.

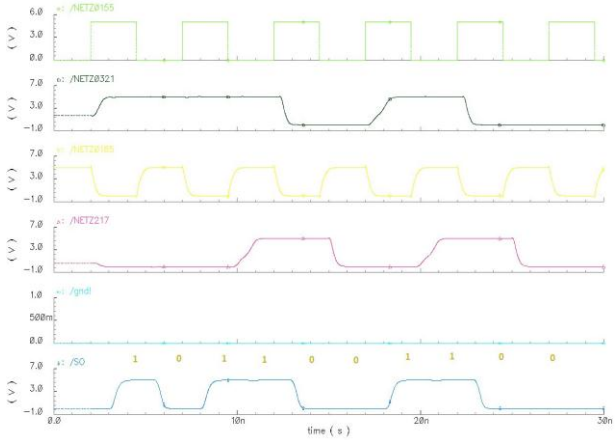

Figure 5. Receiving of 1011001100 data in serial mode.

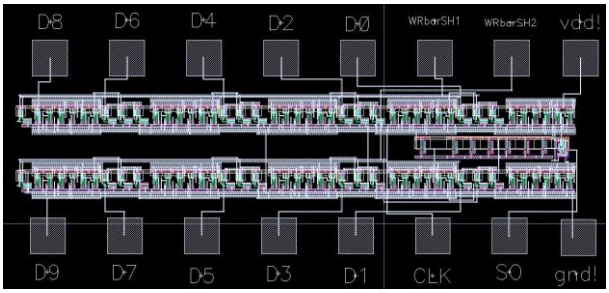

Figure 6. Layout of the serializer circuit.

TABLE IV. Designed Circuit Specifications.

Specification	Value
Power Consumption	<5mw
Die Area	0.15mm^2
CMOS Technology	0.6µm
VDD	5V

V. Conclusion

The designed 8b/10 encoding SerDes circuit consists of two main parts. First part was designed in VERILOG and the second part including a 10b parallel load serializer, was implemented in CADENCE. All of the designed blocks were simulated and combined with other blocks. Timing problems were solved using some digital techniques. Finally whenever an 8-bit data is sent to serializer block, after encoding it into 10-bit code, the deserializer decodes the data into 8-bit so that we are able to get the same information at the output without any lost bit.

References

[1] M. Maadi and B. Bayram, "Custom integrated circuit design for ultrasonic therapeutic CMUT array," *Microsystem Technologies*, pp. 1-17, Mar. 2014.

[2] J. H. Sinsky, M. Duelk, and A. Adamiecki, "High-Speed electrical backplane transmission using duobinary signaling," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 1, pp. 152-160, Jan. 2005.

[3] M. Sharad, V. S. P. Rao, and P. Mandal, "A New double data rate (DDR) dual-mode duobinary transmitter architecture," in *Proc. 24th International Conference on VLSI Design (VLSI Design)*, 2011, pp. 12-17.

[4] J. Cao, M. Green, *et al.*, "OC-192 Transmitter and Receiver in Standard 0.18-µm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, Dec. 2002.
[5] Y. Takasaki, M. Tanaka, N. Maeda, K. Yamashita, and K. Nagano, "Optical pulse formats for fiber optic digital communications," *IEEE Transactions on Communications*, vol. 24, no. 4, pp. 404-413, Apr. 1976.
[6] 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.

Mohammad Maadi was born in Macoo, Iran. He received the B.S. degree in 2007 from IAU and the M.S. degree in 2013 from Middle East Technical University; both degrees were in electrical and electronics engineering. From 2007 to 2010, he worked as an electronics engineer and project manager in some private companies of Iran. He could get the membership of the Iranian Inventors Association after registering his B.S. project, "Intelligent Color Recognizer and Analyzer System", as an invention in the General Department of Industrial Ownerships of Iran in 2008. From 2011 to 2013, he got TUBITAK scholarship as a Research Assistant in the Department of Electrical and Electronics Engineering at Middle East Technical University. During his M.S., he mainly focused on integrated circuit design for flip-chip bonded capacitive micromachined ultrasonic transducers (CMUTs). His research interests include integrated circuit design for ultrasound 3D imaging and therapeutic CMUT arrays, design of analog, digital and mixed-signal integrated circuits and design of micro electromechanical systems (MEMS) for medical applications.