Design and verification of 8b/10b encoder

SANJEEVREDDI JAMAKHANDI[1], PRADEEP SV[2], DR SIVA YELLAMPALLI[3]

[1]*PG Scholar VLSI Design and Embedded System, VTU, VTU Extension Centre, UTL technologies LTD, Bangalore*
[2]*Assistant Professor, VLSI Design and Embedded System, VTU Extension Centre, UTL technologies LTD, Bangalore*
[3]*Principal, VTU Extension Centre, UTL technologies LTD, Bangalore*

---***---

Abstract - *this paper highlights the requirement of many serial data transmission standards utilizes the 8b/10b encoding to ensure sufficient data transitions for clock recovery. The design is implemented using Verilog in Cadence platform. The system is verified using system Verilog coverage driven verification with the 100% code coverage and functional coverage.*

Key Words: **8b/10b encoder, code coverage, functional coverage.**

1. INTRODUCTION

8B/10B encoder is a block that is frequently used in communication systems. Hence development of IP cores for 8B/10B encoder is important. Verification is an exclusive process of checking the function of this code. With the increase in complexity of designs, the functional verification have increased sharply in recent years mainly pushed by the major EDA companies. Today the verification engineers have outnumbered the design engineers of the most complex designs. Studies revolved that 74% of all respective pins of ICs are due to functional errors. Next verification has become the bottle neck in a projects time to profit goal [1]. The coverage carried out to demonstrate the efficiency of performance of the RTL code of 8B/10B encoder.

This paper is organized as follows. Section 2 reviews the structure and coding scheme of 8B/10B encoder, Section 3 explains the functional verification and section 4 proposes the simulated results of the proposed work, while conclusions are wrapped up in Section 5.

2. 8B/10B ENCODER

A byte-oriented transmission code converts an 8 bit symbol to a 10 bit symbol [2].The converted 10 bit symbol should be such that it contains equal numbers of „1‟s and „0‟s. The application of scheme is such that at one time not more than five 0‟s or 1‟s are ever transmitted. The difference in the number of 1‟s and 0‟s is called disparity and can accept values of 0, -1 or +1 in the form of encoded 10-bit symbols. In order to maintain an overall DC balanced stream the disparity of one 10-bit output code is feed back to the encoder in order to compensate for non-zero disparity ifany. The 8B/10B encoder have its application in PCI express, Serial ATA, USB 3.0, Fiber Channel, SSA and many more.

2.1 Structure Of 8B/10B Code

Special characters are included in the transmission codes known as D characters and K characters. In each byte of the transmission code the parity is monitored and accordingly D and K characters are related to positive or negative parity. Here the encoder selects parity for each code word for maintaining balance running parity. The block diagram [3] of the encoder is shown in Fig 1.

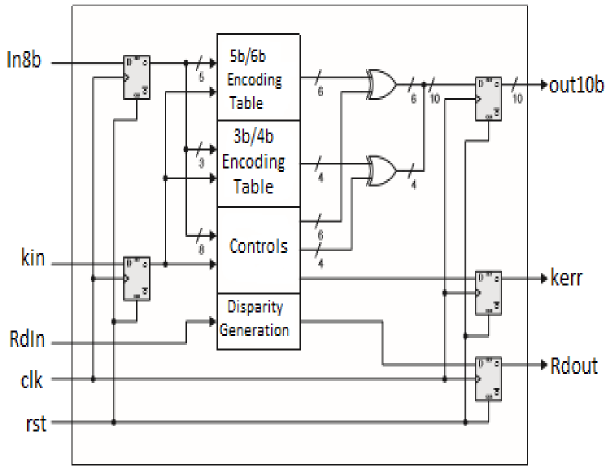


Fig 1: 8B/10B Encoding block diagram

The encoder, the Rdout output should be connected back to the RdIn input. Except clk, rst and RdIn, the other inputs are pipelined to generate the outputs. However, since the running disparity of the data currently being transmitted is required for the encoding of the following data, the RdIn-to-Rdout loopback path can only contain one register level and cannot be pipelined [3].

2.2 8B/10B Encoding scheme

As the scheme name suggest, eight bits of data are transmitted as a 10-bit entity called a symbol or character. The low 5-bits of data are encoded into a 6-bit group (the 5b/6b) and top 3-bits are encoded into a 4-bit group (the 3b/4b). These code groups are concatenated together to for the 10-bit symbol. Every 10-bit encoded data group has one

of the three possibities to help in limiting the number the number of consecutive “1s”or “0s” in any 2-code words [4].
_Five “1s” and five “0s”
_Four “1s” and six “0s”
_Six “1s” and four “1s”

The column heading “Name” in the Table 1, gives the inputs ABCDE as the 32 decimal equivalents with A and E as low and high order bit. The K line must be held at 0 for regular data (D.x).Apart from this a few code points name K.x or D/K.x and have a 1 or x in column K can be a part of special characters that can be identifiable as other than data. The line K must be 1 for encoding special characters. The Table 2 follows the conventions and notations of Table 1 and converts the bits FGH into the digits fghj [2].

TABLE 1. 5B/6B Encoding

Name	ABCDE	Kin	abcdei (RdIn +ve)	abcdei (RdIn -ve)
D.0	00000	0	100111	011000
D.1	10000	0	011101	100010
D.2	01000	0	101101	010010
D.3	11000	0	110001	110001
D.4	00100	0	110101	001010
D.5	10100	0	101001	101001
D.6	01100	0	011001	011001
D.7	11100	0	111000	000111
D.8	00010	0	111001	000110
D.9	10010	0	100101	100101
D.10	01010	0	010101	010101
D.11	11010	0	110100	110100
D.12	00110	0	001101	001101
D.13	10110	0	101100	101100
D.14	01110	0	011100	011100
D.15	11110	0	010111	101000
D.16	00001	0	011011	100100
D.17	10001	0	100011	100011
D.18	01001	0	010011	010011
D.19	11001	0	110010	110010
D.20	00101	0	001011	001011
D.21	10101	0	101010	101010
D.22	01101	0	011010	011010
D/K.23	11101	X	111010	000101
D.24	00011	0	110011	001100
D.25	10011	0	100110	100110
D.26	01011	0	010110	010110
D/K.27	11011	X	010110	101001
D.28	00111	0	001110	001110
K.28	00111	1	001111	110000
D/K.29	10111	X	101110	010001
D/K.30	01111	X	011110	100001
D.31	11111	0	101011	010100

TABLE 2. 3B/4B Encoding

Name	FGH	Kin	fghj (RdIn -ve)	Fghj (RdIn +ve)
D/K.x.0	000	X	1011	0100
D.x.1	100	0		1001
D.x.2	010	0		0101
D/K.x.3	110	X	1100	0011
D/K.x.4	001	X	1101	0010
D.x.5	101	0		1010
D.x.6	011	0		0110
D.x.P7	111	0	1110	0001
D/K.y.A7	111	X	0111	1000
K.28.1	100	1	0110	1001
K.28.2	010	1	1010	0101
K.28.5	101	1	0101	1010
K.28.6	011	1	1001	0110

2.2 Special characters

The extra code points apart from the 256 needed for encoding a byte of data are called special characters. They are used to mark the start and end of packets, for establishment of byte synchronization, and sometimes for communicating control operations such as RESET, SHUT-OFF IDLE, ABORT, and link diagnostics. In Table 3 the set of twelve special characters depicted can be generated by the coding rules given in Table 1 and 2. They all follow with the general coding constraints of a maximum digital sum variation of 6 and a maximum run length of 5 [2].

TABLE 3. Special Characters

Name	HGF EDCBA	Kin	abcdei fghj(RdIn - ve)	abcdei fghj(RdIn +ve)
K28.0	000 11100	1	001111 1000	110000 1011
K28.1	001 11100	1	001111 1001	110000 0110
K28.2	010 11100	1	001111 0101	110000 1010
K28.3	011 11100	1	001111 0011	110000 1100
K28.4	100 11100	1	001111 0010	110000 1101
K28.5	101 11100	1	001111 1010	110000 0101
K28.6	110 11100	1	001111 0110	110000 1001
K28.7	111 11100	1	001111 1000	110000 0111
K23.7	111 10111	1	111010 1000	000101 0111
K27.7	111 11011	1	110110 1000	001001 0111
K29.7	111 11101	1	101110 1000	010001 0111
K30.7	111 11110	1	011110 1000	100001 0111

2.4 Running disparity

Disparity refers to the inequality between the number of ones and zeros within a 10-bit symbol and is used to help maintain DC balance on the link. A symbol with more ones has a positive disparity and more zeros has a negative disparity. When a symbol has an equal number of ones and zeros, it’s said to have a neutral disparity [5].

3. Functional verification

Functional verification is necessary in today’s aspects and main thing it improve the quality and re usability. The design is passing all the tests under verification to check the correctness of the design as per specifications. This verification environment can be reusable and easy tracking of verification progress happens by coverage metrics. Fig.2 show the verification environment composed with several components. The generator component generates random input vectors, using system Verilog HVL (Hardware verification language). The randomness is important to achieve a high distribution over the huge space of the available input stimuli.

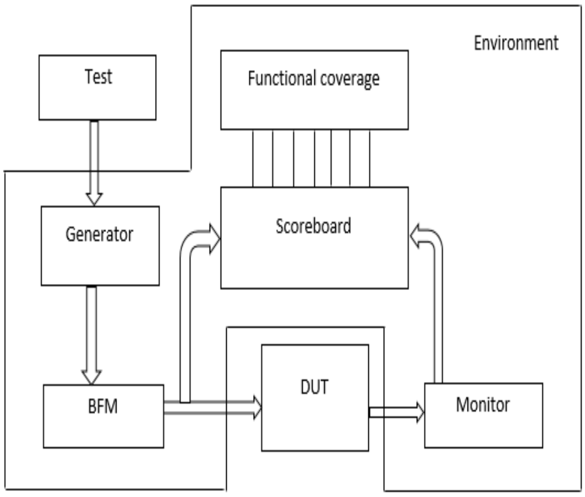


Fig 2: Verification Environment

The bus functional model (BFM) for a device interacts with the DUT by both driving and sampling the DUT signals. Bus functional models are easy to use and provide good performance. Monitor does not drive any signals, it monitor the DUT outputs. The monitor converts the state of the design and its outputs to a transaction abstraction level so it can be stored in a ' score-boards' database to be checked later on.

Scoreboard is referred as storage structure; it stores the generated input vectors till the monitor module sends the output of the DUT. Then scoreboard compare with the output of the DUT. Functional coverage gives the report of how much we have verified and also it shows how many possible scenarios are possible and how many are covered.

Using this reusable verification environment, designed interface is verified and different coverage matrices are created. These coverage matrices include functional coverage, statement coverage, branch coverage and code coverage details obtained from the test bench. Different crossed bins are used to check randomized data which ran in different scenarios.

Verification environment shown in Fig.2 is designed as a reconfigurable VIP (Verification Intellectual Property). It meets the requirement and specification of 8b/10b encoder. This VIP is designed using Mentor Graphic QuestaSim 10.

4. SIMULATION AND VERIFICATION OF THE DESIGN.

The simulation results for the 8B/10B encoder has been presented in Fig.3 and Fig.4.

In Fig 3 shows that simulation result of control characteristics with kin=1.when kin is 1 if any input data out of 12 special characteristics it show the Kerr signal high.

In Fig 4 show the simulation results of data characteristics when Kin=0.

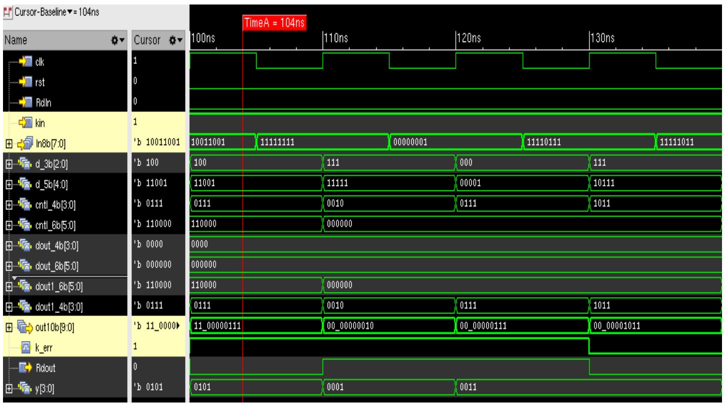


Fig 3: Simulation result of DUT with control characteristics

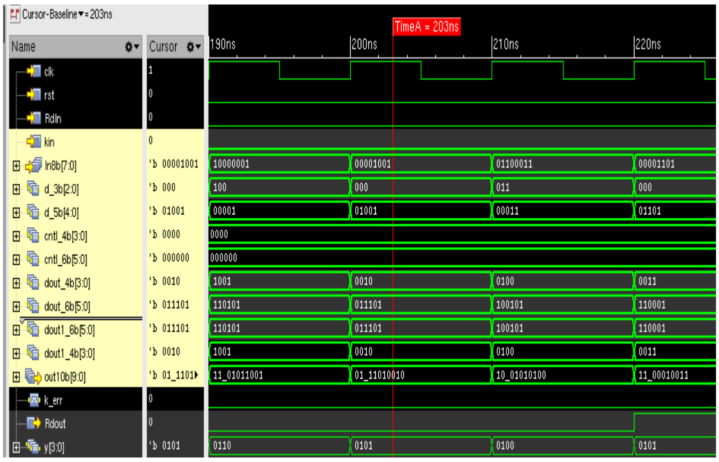


Fig 4: Simulation result of DUT with data characteristics

Simulation and verification results for the 8B/10B encoder is as shown in Fig 5. In fig we can see the design working for many randomly generated inputs in which both character type and running disparity are covered at same clock. Functional coverage of the design is obtained. Fig 6 shows the functional coverage of the design.

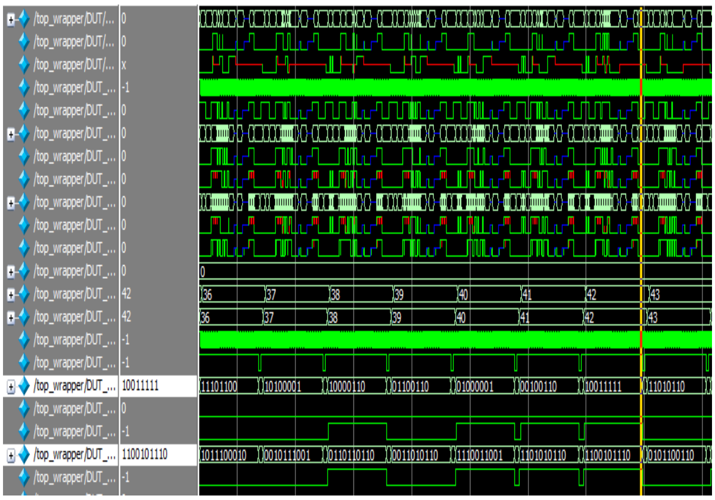
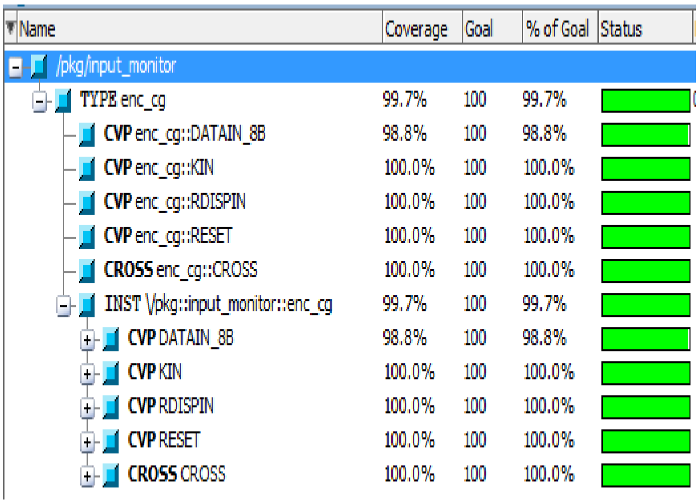


Fig 5: Simulation and verification of the design



Name	Coverage	Goal	% of Goal	Status
/pkg/input_monitor				
TYPE enc_cg	99.7%	100	99.7%	🟩
CVP enc_cg::DATAIN_8B	98.8%	100	98.8%	🟩
CVP enc_cg::KIN	100.0%	100	100.0%	🟩
CVP enc_cg::RDISPIN	100.0%	100	100.0%	🟩
CVP enc_cg::RESET	100.0%	100	100.0%	🟩
CROSS enc_cg::CROSS	100.0%	100	100.0%	🟩
INST \/pkg::input_monitor::enc_cg	99.7%	100	99.7%	🟩
CVP DATAIN_8B	98.8%	100	98.8%	🟩
CVP KIN	100.0%	100	100.0%	🟩
CVP RDISPIN	100.0%	100	100.0%	🟩
CVP RESET	100.0%	100	100.0%	🟩
CROSS CROSS	100.0%	100	100.0%	🟩

Fig 6: Functional coverage of the design.

5. CONCLUSION

The paper presents the functional and code coverage details of an 8B/10B encoder RTL code. The IP is verified for functional and code coverage using Questasim 10. During the process of code coverage it is found that though the encoder gave the expected output, still its overall code coverage is not 100%, indicating that all paths of design are not used to their strength. So it is clear that higher order complete verification has to be taken up to reduce the errors in the design and increase the efficiency of the 8B/10B encoder IP which will it robust in the design of communication system on chip.

ACKNOWLEDGEMENT

The work presented here was supported by VTU Extension Centre, UTL Technologies Ltd., Bengaluru. The author would like to thank Mr. Pradeep S V from VTU Extension Centre, UTL Technologies Limited, Bengaluru, for their technical support and guidance. I would like to express deep sense gratitude to Mr.Ashish Kothari, CEO, iMspired solutions Pvt Ltd, Bengaluru, who considered me like a friend and made me feel at ease in times of difficulty during my project work.

REFERENCES

[1] M. E. Radu, S. M. Sexton, “Integrating extensive functional verification into digital design education,” IEEE Trans. On Education, Vol. 51, No. 3, August 2008.

[2] A. X. Widmer, P. A. Franaszek, “A DC-balanced, partitioned-block 8b/10b transmission code,”IBM Journal of Reasearch and Development, vol. 27, no. 5, pp. 440,

[3] Latticesemi.com, ”8b/10b encoder/decoder,” January - 2015.

[4] M.S.Suma and S.S.Rekha, “16B/20B CODEC Development and its ASIC Implementation.” in Proc. International Conference on Future Computer and Communication, pp.622-626, April,2009.

[5] Mike jackson and ravi budruk,”PCI Express Technology”, comprehensive guide to generation 1.X,2.X,3.0.

[6] Cadence Designs Systems and Mentor Graphics Inc., “Open Verification Methodology User Guide” Version 2.0, Sept.2008 available from http://www.ovmworld.org

[7] Bergeron,“Writing Testbenches Using System Verilog”Springer, ISBN-10: 0-387-29221-7, Business Media, 2006.

BIOGRAPHIES

Sanjeevreddi Jamakhandi is currently pursuing his M.Tech in VLSI design and embedded systems from VTU Extension Centre, UTL Technologies Limited, Bengaluru. Doing project in VTU Extension Centre, UTL Technologies Limited, Bengaluru. He obtained his bachelor’s degree in Electronics and Communication engineering from VTU University, Gulbarga. His interest includes, digital circuits design and verification.