FULL CUSTOM IMPLEMENTATION OF AN 8B/10B ENCODER WITH A MODIFIED CODING TABLE

B. Koteswar Rao*

Dr S. Kishore Reddy**

ABSTRACT

This paper presents a design of 8B/10B encoder with a modified coding table. The proposed encoder has been designed based on a reduced coding table with a modified disparity control block. After being synthesized using CMOS 0.18µm process, the proposed encoder shows the operating frequency of 343 MHz and occupies the chip area of 1886 µm2 with 189 logic gates. It consumes 2.74mW power. Compared to conventional approaches, the operating frequency is improved by 25.6% and chip area is decreased to 43%.

*Associate Professor, ECE, SBCE, Khammam, AP, India

**Associate Professor, ECE, SR Engineering College, Warangal, AP, India

I. INTRODUCTION

The 8B/10B encoder is used to generate ample data transition for facilitating a clock recovery function in the various networks. Also it provides a DC balance by trying to equalize the number of '0' and '1' in the data stream. There are extra codes called as special code group for identifying the data sequence boundary in the data recovery unit. Thus the 8B/10B encoding technique is used in various high-speed serial data transmission standards, which are Fiber channel, Gigabit Ethernet, Server Net, and Infiniband. As the demand of high-speed communication increases, the 8B/10B encoding block is also required to be operating in giga bits per second range. Therefore it is necessary to have a more efficient and high speed operating 8B/10B encoder/decoder design. Fig. 1 shows the 8B/10B encoder/decoder block in communication system.

Most popular, de facto standard, 8B/10B encoder design is based on the scheme proposed by IBM [1]. However, the logic implemented based on the conventional encoding table needs many logic depth, thus the operating speed can be limited. To overcome this speed issue, new encoding approaches were suggested. By introducing pipelining, logic optimization, and utilizing a memory tables for coding and disparity inside ASIC or FPGA were investigated.

In this paper, we present a design of 8B/10B encoder by coding table reduction. The proposed encoder has a simplified coding table using adder compared to conventional coding table and has a modified disparity control block. The paper describes the proposed encoder architecture first and shows the design results.

II. ARCHITECTURE

A. Conventional 8B/10B Encoder

Fig. 2 shows a conventional 8B/10B encoder. Encoding starts by dividing two different sub-blocks, which are 5B/6B encoder (inputs A to E) and 3/4B encoder (inputs F to H). After disparity is checked, the encoding switches set the final encoded output values. The disparity control block is to make DC balance on the data stream by equalizing the number of '0' and '1'. The 'K' input is for special character that is used for data packet boundary detection. The encoding table for the conventional encoder is given in Table I.

The encoding table in Table I is most widely accepted for realizing the encoding function. However, implementation of the original encoding table given in Table I requires many logic stages, which lead to a low operating speed. These problems need to be improved for higher speed operation system applications.

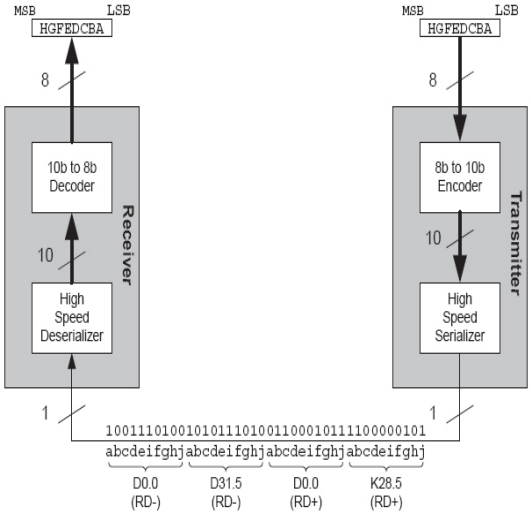

Figure1. 8B/10B encoder/decoder block in communication system.

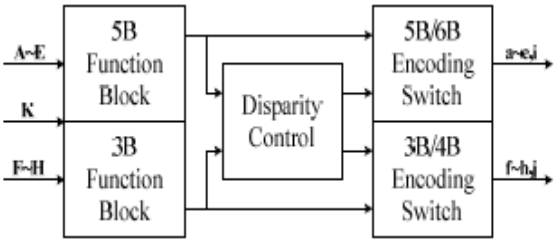

Figure2. Block diagram of conventional 8B/10B encoder.

B. Proposed 8B/10B Encoder

The proposed block diagram for a modified 8B/10B encoder is shown in Fig. 3. The blocks look similar to the conventional one, but at input side the pre encoder blocks are modified and the special character inputs are omitted. Instead, the special character case will be handled in the pre encoder blocks.

As indicated in the Table I in the original 5B/6B encoding, the encoded outputs (denoted as 'abcd') are not changed except input data 'A' through 'D' are all zero or one. From this point, by classifying the summing values of the input bit patterns, the original encoding table can be reduced. The simplified encoding table is given in Table II.

In the Table II, the output value 'x' means that no logic is required because input and output are the same. The encoding switch in conventional scheme can be replaced by DFF's and XOR gates. Complementing operation depending on encoded values is not taken. Instead the non-complemented output and disparity values are generated and processed later in following blocks.

The codes of D.24 and K.28 in the original table are not applied by the reduction algorithm. The special cases (D.24, K.28 in the original table) should be handled in the modified encoding scheme because in case of D.24 (=00011) input, the summing value gives '1' at ABCD. Therefore the outputs are not changed. But depending on previous bit sequence of 'f' and 'g', the bit length of consecutive '1' or '0' could be more than five, which is not allowed in the 8B/10B encoder. In this case it is encoded to '001100' for DC balance. In addition to that, in the case of K.28, the encoded output 'i' should be changed from '0' to '1' for representing comma value. Therefore additional logic for handling D.24 and K.28 cases is required. The additional logic, however, has a little overhead in logic size.

Table IV shows the simplified 3B/4B encoding table with the same reduction algorithm applied in the 5B/6B encoding reduction. Here, we also need to take care of the special code for preventing more than five consecutive same bit patterns. Also it is needed to convert '1110' to '0111' for the '111' input case.

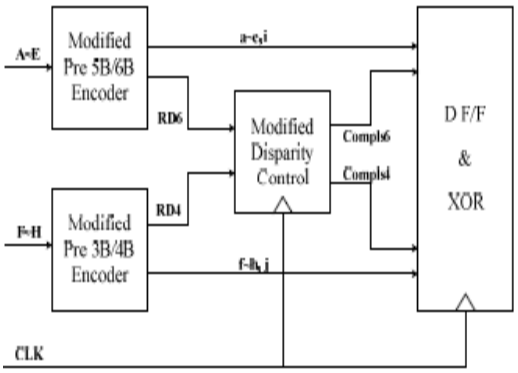

Figure 3. Block diagram of Proposed 8B/10B encoder.

TABLE I. ORIGINAL 5B/6B ENCODING TABLE

Name	ABCDE K	Bit encoding	D-1	abcdei	D0
D.0	00000 0	L04	+	011000	-
D.1	10000 0	L13·E'	+	100010	-
D.2	01000 0	L13·E'	+	010010	-
D.3	11000 0	L22·E'	x	110001	0
D.4	00100 0	L13·E'	+	001010	-
D.5	10100 0	L22·E'	x	101001	0
D.6	01100 0	L22·E'	x	011001	0
D.7	11100 0		-	111000	0
D.8	00001 0	L13·E'	+	000110	-
D.9	10001 0	L22·E'	x	100101	0
D.10	01001 0	L22·E'	x	010101	0
D.11	11001 0		x	110100	0
D.12	00110 0	L22·E'	x	001101	0
D.13	10110 0		x	101100	0
D.14	01110 0		x	011100	0
D.15	11110 0	L40	+	101000	-
D.16	00001 0	L04, L04·E	-	011011	+
D.17	10001 0	L13· D'·E	x	100011	0
D.18	01001 0	L13· D'·E	x	010011	0
D.19	11001 0		x	110010	0
D.20	00101 0	L13· D'·E	x	001011	0
D.21	10101 0		x	101010	0
D.22	01101 0		x	011010	0
D/K.23	11101 x		-	111010	+
D.24	00011 0	L13·D·E	+	001100	-
D.25	10011 0		x	100110	0
D.26	01011 0		x	010110	0
D/K.27	11011 x		-	110110	+
D.28	00111 0		x	001110	0
K.28	00111 1	L22·K	-	001111	+
D/K.29	10111 x		-	101110	+
D/K.30	01111 x		-	011110	+
D.31	11111 0	L40, L40·E	-	101011	+

TABLE II. MODIFIED 5B/6B ENCODING TABLE

E	Adder Result (for 4 input (A~D))	Encode Output						Next RD
		a	*b*	*c*	*d*	*e*	*i*	
0	0(3'b000)	x	1	1	x	0	0	-
0	1(3'b000)	x	x	x	x	1	0	-
0	2(3'b000)	x	x	x	x	0	1	0
0	3(3'b000)	x	x	x	x	0	0	0
0	4(3'b000)	x	0	x	0	0	0	-
1	0(3'b000)	x	1	1	x	1	1	+
1	1(3'b000)	x	x	x	x	1	1	0
1	2(3'b000)	x	x	x	x	1	0	0
1	3(3'b000)	x	x	x	x	1	0	+
1	4(3'b000)	x	0	x	0	1	1	+

(Except for D.24 case => abd=x, c=1, e,i == 00, NextRD = - K.28 case => i=1, NextRD = + others same D.28)

C. Modified Disparity Control Block

Table V shows an original disparity coding tables. In conventional encoding table, the complemented output and current disparity value are generated after checking the previous disparity value. In the modified scheme, however, the encoded output and current disparity value are generated without considering the previous disparity value. By checking the current running disparity and the outputs from 5B/6B and 3B/4B encoding block with current disparity, the next disparity value and the execution of complementing the final output are determined in the modified disparity block. The block diagram of the modified disparity control is shown in Fig. 4. Running disparity outputs of CurRD6 and CurRD4 are coming from the modified Pre 5B/6B encoder and Pre 3B/4B encoder, respectively.

Depending on previous and current disparity outputs, the disparity control determines whether to complement the pre encoded 5B/6B and 3B/4B outputs. The whole operation of the disparity control is summarized in Table VI.

PreRD and CurRD in Table VI stand for previous and current running disparities of encoded output. And when 'compls' is equal to '1', then complementing the encoded output is executed. The Next RD is next running disparity used as Pre RD after one clock cycle in the 5B/6B disparity check block. In order to decide the complement the pre encoded outputs, the previous 4B encoded outputs are used as shown in Fig. 4.

For K28.3 code, the disparity value in the previous encoding stage is minus (-) and the disparity value of current encoding stage becomes plus (+). Then the sign of 'compls4' is '0' and generate the encoded data without complementing action. As a result the encoded result can have a run length of 6. This is not allowed in the 8B10B encoding scheme. In order to solve the problem, the logic has to set the 'Compls4' for K28.3 and K28.2 code and the 'Compls6' for D7.7.P7 code to be '1' after checking the disparity.

III. PERFORMANCE ANALYSIS

The proposed encoder has been design using Verilog and simulated using the VCS. Synthesis and P&R (Place and Route) are performed by the Synopsys Design Compiler and the Astro, respectively, using the CMOS 0.18um technology library. Fig. 6 shows the layout of the proposed algorithm.

Fig. 5 shows a simulated output for the proposed encoder. A PRBS (Pseudo Random Bit Sequence) input patterns are used to verify the error-free encoding operation.

As an example, the expected encoded output '0110010010' (2th byte in output_10b signal) is generated for the D6.4 code (='01100001': first byte in input_8b signal). And after decoding this value by a decoder, original input value is exactly recovered.

For the performance comparison, we also designed a Encoder based on the original IBM's encoding table. In Table VII, the performance comparison to original IBM design is given. It shows that the cell area has been increased to 2.1%, but the operating frequency is improved by 25.6% compared to conventions. The 8B/10B encoder proposed by Xilinx [5] has been also compared. The operating frequency is improved by 3.9%, the cell area has been decreased to 43% compared to the Xilinx approach.

The operating frequency improvement has been achieved by the modified disparity algorithm comparing the current disparity rather than previous disparity value in conventions. The Xilinx's encoder used a state-machine based operation to increase the operating speed. In terms of power consumption, the proposed scheme shows about 40% power reduction compared to IBM's. The Xilinx's encoder shows the lower power consumption but it needs a latency of five clock cycles. As a result, the proposed 8B/10B encoder shows the overall performance improvement in the operating frequency, chip area, and power consumption.

IV. CONCLUSION

This paper presents a design of 8B/10B encoder with a modified coding table. The proposed encoder has been designed based on a reduced coding table with a modified disparity control block. After being synthesized using CMOS 0.18μm process, the proposed encoder shows the operating frequency of 343 MHz and occupies the chip area of 1886 μm2. It consumes 2.74mW power. Compared to conventional approaches, the operating frequency is improved by 25.6% and chip area is decreased to 43%.

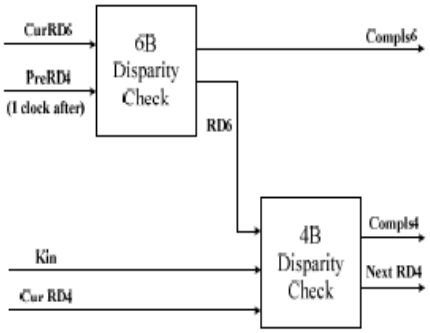

TABLE III. ORIGINAL 3B/4B ENCODING TABLE

Name	FGH K	Bit Encoding	D-1	fghj	D0
D/K.x.0	000 x	F' ·G' ·H'	+	0100	-
D.x.1	100 0	(F±G) ·H'	x	1001	0
D.x.2	010 0	(F±G) ·H'	x	0101	0
D/K.x.3	110 x		-	1100	0
D/K.x.4	001 x		+	0010	-
D.x.5	101 0		x	1010	0
D.x.6	011 0		x	0110	0
D.x.P7	111 0		-	1110	+
D/K.y.A7	111 x	F ·G ·H·(S+K)	-	0111	+
K.28.1	100 1	(F±G)·H'	+	1001	0
K.28.2	010 1	(F±G)·H'	+	0101	0
K.28.5	101 1		+	1010	0
K.28.6	011 1		+	0110	0

TABLE IV. MODIFIED 3B/4B ENCODING TABLE

(S+K)	H	Adder Result (input F,G)	Encode Output				Next RD
			f	*g*	*h*	*j*	
x	0	0(2'b00)	x	1	x	0	-
x	0	1(2'b01)	x	x	x	1	0
x	0	2(2'b10)	x	x	x	0	0
x	1	0(2'b00)	x	x	x	0	-
x	1	1(2'b01)	x	x	x	0	0
x	1	2(2'b10)	x	x	x	0	+
x	1	2(2'b10)	0	x	x	1	+

(Except for D/K.y.A7 case => f=0,j=1)

TABLE V. ORIGINAL 5B/6B, 3B/4B DISPARITY CODING TABLE

5B/6B disparity coding table			
Name	*Disparity*	*Name*	*Disparity*
D.0	L22' ·L31' ·E'	D.17	
D.1	L22' ·L31' ·E'	D.18	
D.2	L22' ·L31' ·E'	D.19	
D.3		D.20	
D.4	L22' ·L31' ·E'	D.21	
D.5		D.22	
D.6		D/K.23	L22' ·L13' ·E
D.7	L31·D' ·E'	D.24	L13·D·E
D.8	L22' ·L31' ·E'	D.25	
D.9		D.26	
D.10		D/K.27	L22' ·L13' ·E
D.11		D.28	
D.12		K.28	K
D.13		D/K.29	L22' ·L13' ·E
D.14		D/K.30	L22' ·L13' ·E
D.15	L22' ·L31' ·E'	D.31	L22' ·L13' ·E
D.16	L22' ·L13' ·E		
3B/4B disparity coding table			
Name	*Disparity*	*Name*	*Disparity*
D/K.x.0	F' ·G'	D.x.P7	F·G, F·G·H
D.x.1		D/K.y.A7	F·G, F·G·H
D.x.2		K.28.1	(F≠G)·K
D/K.x.3	F·G	K.28.2	(F≠G)·K
D/K.x.4	F' ·G'	K.28.5	(F≠G)·K
D.x.5		K.28.6	(F≠G)·K
D.x.6			

TABLE VI. MODIFIED DISPARITY AND COMPLMENT DECISION TABLE.

PreRD6(4)	CurRD6(4)	Compls6(4)	NextRD6(4)
-	-	1	+
-	0	0	-
-	+	0	+
0	-	0	-
0	0	0	0
0	+	0	+
+	-	0	-
+	0	0	+
+	+	1	-

(Except for K28.3, K28.2 case => force compls4='1'
D.7.7.P7 case => force compls6='1')

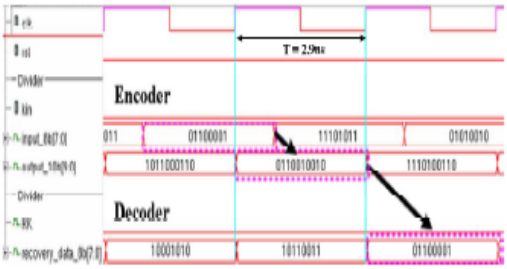

Figure 5. 8B/10B Encoder/Decoder Simulation Result.

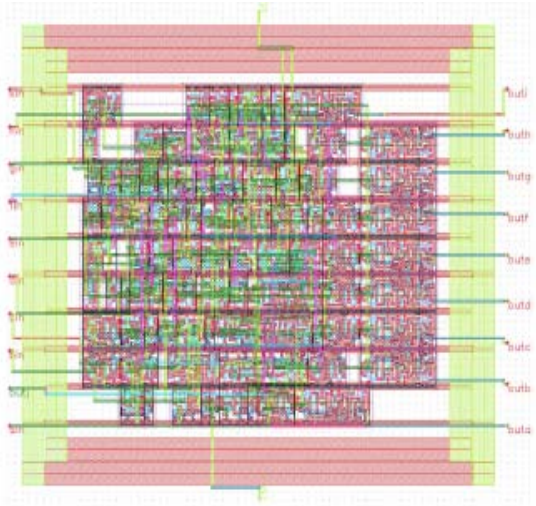

Figure 6. 8B/10B Encoder layout.

TABLE VII. PERFORMANCE SUMMARY

	This work	IBM	Xilinx
Maximum Frequency (MHz)	343	273	330
Latency (Clocks)	0	0	5
Cell Area (μm^2)	1886	1847	3308
Total # of gates	189	185	331
Power Consumption (mW)	2.74	3.83	1.06

ACKNOWLEDGMENT

This work was supported by "System IC 2010" project of Korea Ministry of Knowledge Economy. Hardware and software tools were supported by IT-SOC and the IDEC program.

REFERENCES

[1] A.X.Widmer, "A DC-balanced, Partitioned-Block, 8B/10B transmission Code", IBM J. Res. Develop, vol. 27, pp.440-451, September, 1983.

[2] H. Lee, S. Park, "High speed 8B/10B encoder/decoder design by Logic Reduction", 2003 SoC Design Conf., pp.910-913, Nov, 2003.

[3] Actel, "Implementing an 8b/10b Encoder/Decoder for Gigabit Ethernet", Application Note, Oct, 1998.

[4] Lattice, "8b/10b Encoder/Decoder" Reference Design, Nov, 2002.

[5] Xilinx, Logic core, 8B/10B Encoder v5.0, May, 2004.