An 8B/10B Encoder With 2GHz Operating Frequency

Hamed Ashrafi
Microelectronics Research Center
Urmia University, Urmia - Iran
st_h.ashrafi@urmia.ac.ir

Morteza Mousazadeh
Microelectronics Research Center
Urmia University, Urmia – Iran
m.mousazadeh@urmia.ac.ir

Kh.Hadidi
Microelectronics Research Center
Urmia University, Urmia – Iran
kh.hadidi@urmia.ac.ir

***Abstract*— This paper presents a design of 8B/10B encoder with a new architecture, reduced coding table and improved disparity and run length control block which make it suitable for high-speed data transmission. The original logic is broken into two sub logic and designed in two stages based on parallel pipelined structure. In the first stage the original 3B/4B and 5B/6B coding tables are reduced and reformed to implement their sub-blocks based on parallel processing technique. In the second stage, the disparity and run length control blocks are designed separately to work in parallel. These notifications result the minimum timing delay at each stage. The encoder is realized in CMOS .18um process. It achieves the operating frequency over 2GHz and the power consumption is 9.38 mw with 1.8v supply voltage.**

Keywords— ***8B/10B encoder, High-speed data transmission, Disparity and Run length, DC-balanced***

I. INTRODUCTION

Todays, due to rapid development in the field of electronics and communication, the high-speed data transmission is achievable in long-haul communication and computer networks. The optical fiber technology, provides condition for data stream speed, higher than 20 Gb/s due to its high transmission speed and big capacity.

In a serial data transmitting system, the number of following ones or zeros in transmitted data, doesn't have any specified number. The consecutive ones or zeros in data transmission line, make a fluctuated voltage, which severely infect the signal integration and the EMI features of system [1]. To resolve these type of problems, we need a code which is free of dc or has constant dc component regardless of data pattern [2]. The most common and standard technique is the 8B/10B encoding and decoding, proposed by IBM in early 1980s [2] and extensively, considered in high-speed data transmission. This method has two important properties, first, guarantees the equal number of ones and zeros transmitted in short timeframes, which means, the dc value of the transmission line is constant. The second one is the limited run length, which means that the maximum permitted number of consecutive ones or zeros is five. This feature facilitates the clock recovery issues at the receiver side.

The 8B/10B encoder, based on IBM proposed algorithm has been implemented in different ways, such as modifying coding table or using new architecture for general block to increase speed of operation and reduce power consumption. In [3], they have modified the coding table and disparity control system and achieved the speed of 343 MHz with 2.74 mw power consumption. In [1], the 8B/10B encoder is designed based on pipeline and parallel processing. It works at frequency of 1GHz and consumes power of 4.45mw. Finally, in [4] which was implemented on FPGA, the original large combinational logic is divided in two sub logic block with the help of adding two existing registers [4]. Its operating frequency is 500/1000 MHz.

This paper is organized as follows: section II gives an overview of conventional 8B/10B encoder. The proposed 8B/10B encoder is described in section III. Section IV discusses the circuit implementation and presents the simulation result. Finally, section V draws the conclusions.

II. CONVENTIONAL 8B/10B ENCODER

A conventional block diagram of this technique is shown in Fig. 1.

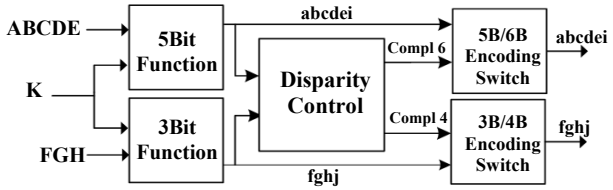

Fig1. Block Diagram of 8B/10B Encoder Proposed by IBM

To simplify the implementation, the 8B/10B encoding table was broken down to a 3B/4B and a 5B/6B encoding tables as shown in tables I and II. The codes are divided into two groups which are named data and special character. The data specified by D.x and the special characters defined by K.x. The special character generally used as data packet boundary detection and sometimes to signal control functions such as ABORT, RESET,…[2] .

The incoming byte is portioned into 5B/6B and 3B/4B sub-block and the disparity of encoded data was checked by disparity control sub-block. The running disparity (RD) computation was presented in Fig (2) [1]. Assuming the RD is "-", if the encoded data is neutral, the RD remains unchanged, and if it has "+" disparity, the RD will become (RD+), otherwise the error of DC imbalanced will be detected, which should be corrected by complementing order.

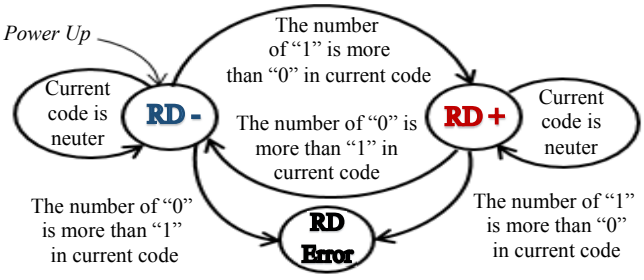

Fig2. RD State Conversion Diagram

978-1-5090-0916-9/16/$31.00 ©2016 IEEE

Table I. Original 5B/6B Encoding Table

Name	ABCDE K	Bit encoding	D_{-1}	abcdei	D_0	abcdei alternate
D.0	00000 0	L04	+	011000	-	100111
D.1	10000 0	L13.E'	+	100010	-	011101
D.2	01000 0	L13.E'	+	010010	-	101101
D.3	11000 0	L22.E'	X	110001	0	
D.4	00100 0	L13.E'	+	001010	-	110101
D.5	10100 0	L22.E'	X	101001	0	
D.6	01100 0	L22.E'	X	011001	0	
D.7	11100 0		-	111000	0	000111
D.8	00010 0	L13.E'	+	000110	-	111001
D.9	10010 0	L22.E'	X	100101	0	
D.10	01010 0	L22.E'	X	010101	0	
D.11	11010 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	-	010111
D.16	00001 0	L04, L04.E	-	011011	+	100100
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	+	000101
D.24	00011 0	L13.D.E	+	001100	-	110011
D.25	10011 0		X	100110	0	
D.26	01011 0		X	010110	0	
D/K.27	11011 X		-	110110	+	001001
D.28	00111 0		X	001110	0	
K.28	00111 1	L22.K	-	001111	+	110000
D/K.29	10111 X		-	101110	+	010001
D/K.30	01111 X		-	011110	+	100001
D.31	11111 0	L40, L40.E	-	101011	+	010100

Table II. Original 3B/4B Encoding Table

Name	FGH K	Bit encoding	D_{-1}	fghj	D_0	fghj alternate
D/K.x.0	000 x	F'.G'.H'	+	0100	-	1011
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	0011
D/k.x.4	001 x		+	0010	-	1101
D.x.5	101 0		0	1010	0	
D.x.6	011 0		0	0110	0	
D.x.P7	111 0		-	1110	+	0001
D/k.y.a7	111 x	F.G.H(S+K)	-	0111	+	1000
K.28.1	100 1	(F≠G).H'	+	1001	0	0110
K.28.2	010 1	(F≠G).H'	+	0101	0	0010
k.28.5	101 1		+	1010	0	0101
K.28.6	011 1		+	0110	0	1001

S = {e.i.(D_{-1} = -)} OR {e'.i'.(D_{-1}=+)

III. PROPOSED 8B/10B ENCODER

Fig. 3 shows the proposed 8B/10B encoder, which two different approaches was considered to improve the performance and boost the operating frequency. First, the general block is implemented based on parallel pipelined processing technique and second, each stage is optimized for high speed operation. In stage1, the coding table is reduced and improved to create simple relation between input and output which result low delay and low chip area. In stage2, the disparity and run length control sub-blocks are designed in parallel processing technology to work separately with minimum timing delay. The proposed encoder is described in the following.

A. *Proposed Structure*

In the first point of view, the conventional 8B/10B encoder block (Fig.1) is broken in two sub-blocks as shown in Fig.3. By the rising edge of the first clock, the 8bit input data and k character, are loaded on the input register and sent stage1. In this stage, the input data are distributed between The 5B/6B and 3B/4B encoders and encoding is done in parallel to produce the pre-encoded 6bit and 4bit with their pre-disparity which are labeled as Cur RD6 and Cur RD4. By second clock, these pre-products are loaded on the middle register and forwarded to the stage2, where the disparity and run length of pre-encoded data are controlled and the final encoded bits are determined which will be explained later. Eventually, the final encoded data are loaded and stored in the output register by next clock. After using this structure, the general delay is divided between two stages almost equally and the through-out rate is doubled.

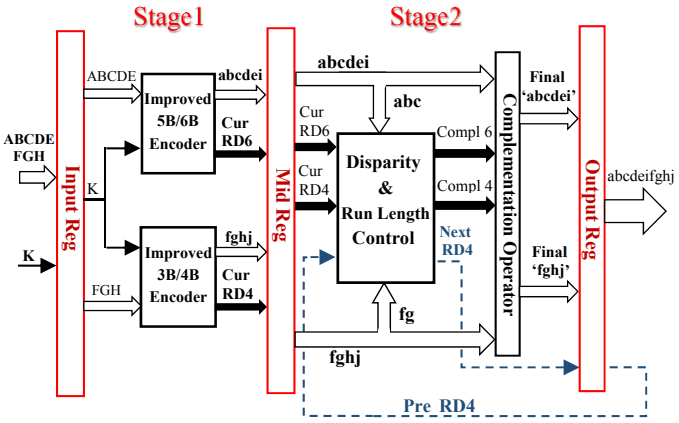

Fig3. Block Diagram of Proposed 8B/10B Structure

B. *Improved Coding Table*

The main table is designed in such a way that the encoded bits have minimal changes compared to initial state (table I and II). Some modifications could be done to create simple relation between input and output bits and reduce coding table. For explanation we consider the improved 5B/6B encoder table as depicted in table III. The encoded data and their disparity are generated without considering the amount of input data and just based on value of two significant bits [DE = 00, 01,10 or 11], sum of the next three bits [A+B+C = 0,1,2 or 3] and state of k character. In the table III, the output value with uppercase English means that the input and output are the same and no logic is required. The K value (x) means that the output bits, are independent of the K amount.

According to table III, the implementation of improved 5B/6B encoder sub-block diagram is shown in Fig.4 which consist of two section and each section sub-blocks work in parallel. In section 1 the adder block adds A, B and C and state identifier concurrently determines the state based on D and E values. The pre-encoded bits and their disparity are generated in parallel. In the section 2, According to the result of previous section and the K input, the pre-encoded output with current

RD6 are generated. By this procedure the logics and delay of 5B/6B encoder are reduced dramatically compared with conventional ones. For example, in [3] which is similar to our work and implemented based on own modified table [3], the logic of recognition of the two ones 'A+B+C+D = 2' (which creates the maximum delay in section1), could be implemented as Fig.5(a). In our work, this logic is defined as 'A+B+C = 2' and implemented as Fig.5 (b) which is miniaturized more than 55% compared to Fig.5(a). So the speed is optimized as well as possible in this sub-block.

Table III. Improved 5B/6B Coding Table

K	DE	Adder result of (A~C)	Pre- Encode output						Current RD
			a	b	c	d	e	i	
X	00	0	A	1	1	D	E	0	-
X	00	1	A	B	C	D	1	0	-
X	00	2	A	B	C	D	E	1	0
X	00	3	A	B	C	D	E	0	0
X	01	0	A	1	1	D	E	1	+
X	01	1	A	B	C	D	E	1	0
X	01	2	A	B	C	D	E	0	0
X	01	3	A	0	C	0	E	0	+
X	10	0	A	B	C	D	1	0	-
X	10	1	A	B	C	D	E	0	0
X	10	2	A	B	C	D	E	1	0
X	10	3	A	0	C	0	E	0	-
X	11	0	A	B	1	D	0	0	-
1	11	1	A	B	C	D	E	1	+
X	11	1	A	B	C	D	E	0	0
X	11	2	A	B	C	D	E	0	+
X	11	3	A	0	C	0	E	1	+

Table IV. Improved 3B/4B Coding Table

S+K	H	Adder result of (F, G)	Pre- Encode output				Current RD
			F	g	H	j	
X	0	0	F	1	H	0	-
X	0	1	F	G	H	1	0
X	0	2	F	G	H	0	0
X	1	0	F	G	H	0	-
X	1	1	F	G	H	0	0
0	1	2	F	G	H	0	+
1	1	2	0	G	H	1	+

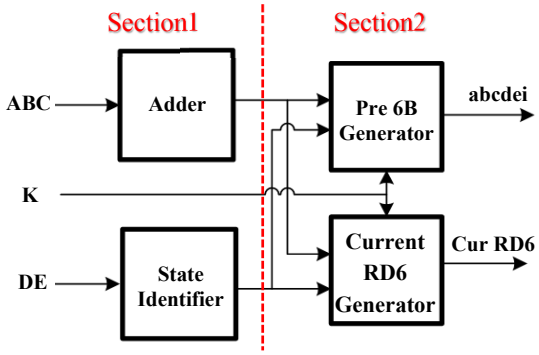

Fig4. Block Diagram of Improved 5B/6B Encoder

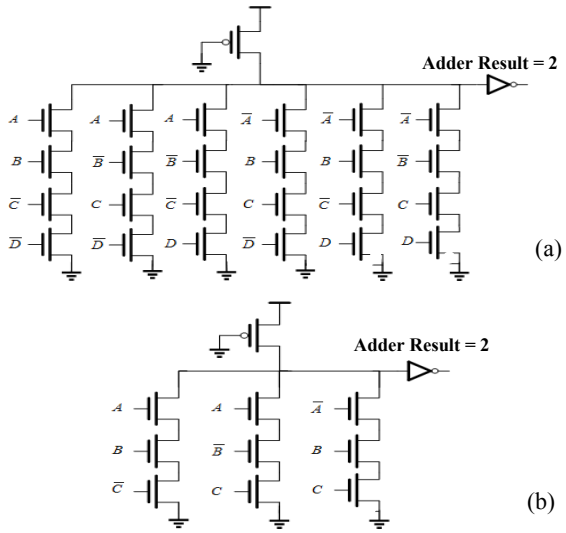

Fig5. Logic of 'Adder Result=2', (a) Work [3], (b) Our Work

To implement the 3B/4B encoder, we use the modified pre 3B/4B encoder proposed in [3] which is optimized enough and depicted in table IV. The pre 3B/4B encoder and pre 5B/6B encoder in stage 1, are processing Simultaneously (Fig.3), so the delay of this stage determined by pre 5B/6B block due to large size of logics. By proposed 5B/6B coding table, it is optimized for minimum delay.

The code of k28 in the original table is a special case, because it couldn't be reduced by proposed algorithm and should be handled in the improved table. Also, the code of D.x.p7 and D/k.y.A7 are handled in the improved 3B/4B table, based on value of S+K.

C. Disparity and Run Length Control Block

This block is placed in satge2 and composed of two sub-block which operate in parallel and control the disparity and run length independently (Fig. 6). The operation is described as bellow.

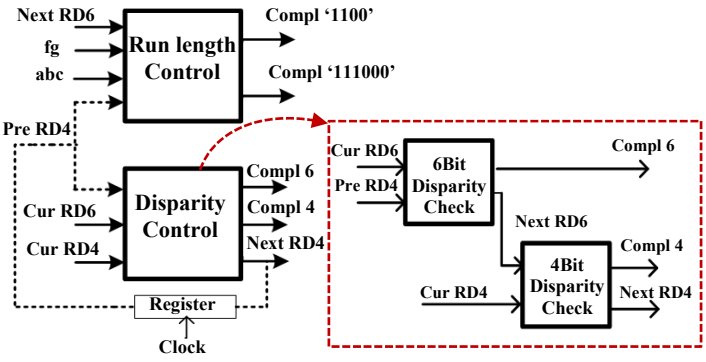

Fig6. Block Diagram of the Disparity & Run Length Control

1) Disparity Control: According to the modified table (table III and IV), most of the encoded 6bits and 4bits are non-neutral, which cause DC-unbalanced data stream. To prevent this problem, the disparity of pre-encoded data should be checked and determined whether to change the disparity or not. For this purpose, we use the disparity control sub-block proposed in [3] and its operation is summarized in table V. In

this block the disparity of pre-encoded data (curRD) compared with disparity of previous data (PreRD) and determined whether to complement it and change the disparity or not, using 'compl 6' and 'compl 4'. As shown in block diagram (Fig.6), the NextRD6 used as PreRD6 for 4bit disparity check block and the NextRD4 stored in register to use as PreRD4 after one clock cycle in the 6bit disparity check block.

Table V. Operation Algorithm of Disparity Control Block

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

2) *Run Length Control:* The original tables proposed with IBM [2] are arranged in the way that, the possible run length of 5 and more in data stream, have minimum chance. As mentioned, the run length for encoded data couldn't be more than five. According to the operation algorithm of disparity control block (table V) and amount of pre-encoded data, the cases which cause run length more than 5, can be considered in table VI. The run length control block predicts these modes based on disparity block results (PreRD4 , NextRD6) and value of some pre-encoded bits ('abc' , 'fg') and restrains the run length error by complementing '111000' and '1100'.

Table VI. Possible Run Length Error

Previous RD	Error condition
- or 0	001111 , 1100
+	111000 , 0001
- or 0	0111 , 111000

IV. IMPLEMENTATION AND PERFORMANCE ANALYSIS

The proposed encoder has been design in 180nm CMOS technology. To achieve high speed and low area, the circuits are implemented by Pzedo-Nmos logic. The registers are designed by TSPC Flip-Flop [5], which have dynamic structure and low delay. The reference clock is distributed between input, middle and output registers, based on H-tree clock distribution. The whole circuit is simulated using Hspice and Part of simulation is presented in Fig. 7. Table VII shows results and comparison among published 8B/10B encoders. At typical corner, the encoder could achieve 2 GHz operating frequency which is two times higher than the best reported speed in the references. The power consumption (including input and output registers and clock distribution too) is about 9.38 mw from 1.8v supply in 2 GHz. It occupies area of about 5080 um^2 by considering the input and output registers and clock distribution. With high reliability, the encoder could work at 1710 MHz and 2285 MHz at slow and fast corner respectively. Compared to [3] the operation frequency is improved by 5.8 times while the power consumption just increased 3.42 times. The proposed encoder shows 100% improvement in speed ،110% increase in power and 15% decrease in area compared to [1]. So it could be evaluated that, the proposed encoder has high speed and acceptable power consumption while it is suitable for very high speed serial-linked data transmission systems.

V. CONCLUSION

The proposed high speed 8B/10B encoder designed with reasonable power consumption. For this purpose, we design the main structure based on pipeline and parallel processing technology. The original 8B/10B encoder divided in two stages where the sub-blocks of each stage are processed in parallel. The main coding table is also modified for high-speed operation with low power and low area. It is carefully laid-out to decrease the parasitic capacitance and wiring metals. The simulation results confirm the advantages of proposed 8B/10B encoder to similar ones.

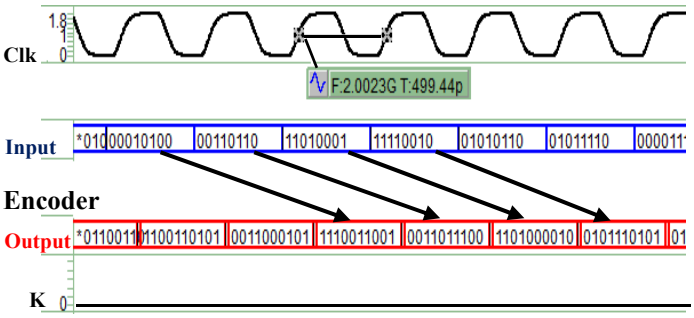

Fig.7 8B/10B Simulation Result

Table VII. Performance Summary

		This work	[1]	[3]	[4]
Technology		.18um	.18um	.18um	.18um
Max Frequency (MHz)		2000	1000	343	500/1000
Corner (MHz)	SS	1710	714	N.A	N.A
	FF	2285	N.A	N.A	N.A
Area (um^2)		5080	6010	1886	2000
Power (mW)		9.38 (in 2 GHz)	4.45	2.74	3.00

REFERENCES

[1] Si-liang Hua , Dong-hui Wang , "A 1.1 GHz 8B/10B encoder and decoder design," *2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics* , pp. 138-141, Nov. 22-24 Sept. 2010.

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

[3] Yong-woo Kim, Jin-ku Kang, "An 8B/10B Encoder With A Modified Coding Table", *Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on* , pp. 1522-1525, Nov. 30 2008-Dec. 3 2008.

[4] Sujata Pandey , Ashutosh Gupta, "Implementation of 8B/10B encoder-decoder for Gigabit Ethernet Frame," *2014 Eleventh International Conference on Wireless and Optical Communications Networks (WOCN)*, pp. 1-4, 11-13 Sept. 2014.

[5] sung-Hyum Yang, Younggap You and Kyoung-Rok Cho, "A New Dynamic D-flip-flop Aiming at Glitch and Change Sharing Free," IEICE TRANS. ELECTRON., VOL.E86-C, NO.3 MARCH 2003.