IEICE Electronics Express, Vol.5, No.16, 581–585

High-speed 8B/10B encoder design using a simplified coding table

Yong-Woo Kim, Beomseok Shin, and Jin-Ku Kang[a)]
School of Electronics Engineering, Inha University,
253 Yonghyun-dong, Nam-Gu, Incheon 402–751, Korea
[a)] *jkang@inha.ac.kr*

Abstract: This letter presents a high-speed 8B/10B encoder design using a simplified coding table. The proposed encoder also includes a modified disparity control block. Logic simulation and synthesis have been done for the performance verification. After synthesized with a CMOS 0.18 μm process, the proposed design shows the operating frequency of 343 MHz with no latency. The synthesized chip area is 1886 μm^2 with 189 logic gates. The proposed 8B/10B encoder shows the overall performance improvement compared to previous approaches.

Keywords: 8B/10B encoder, simplified encoding table, disparity, CMOS
Classification: Integrated circuits

References

[1] A. X. Widmer, "A DC-balanced, Partitioned-Block, 8B/10B transmission Code," *IBM J. Res. Develop*, vol. 27, pp. 440–451, Sept. 1983.
[2] Actel, "Implementing an 8b/10b Encoder/Decoder for Gigabit Ethernet," *Application Note*, Oct. 1998.
[3] Lattice, "8b/10b Encoder/Decoder," Reference Design, Nov. 2002.
[4] Xilinx, Logic core, 8B/10B Encoder v5.0, May 2004.

1 Introduction

The 8B/10B encoder is used to generate sufficient data transitions for facilitating clock and data recovery in the various networks. Also it provides a DC balance by trying to equalize the number of '0' and '1' in the data stream. Most popular, de facto standard, 8B/10B encoder is based on the scheme proposed by IBM [1]. However, the logic implemented based on the IBM's encoding table needs deep logic depths, which limits the operating speed. To be adopted in high-speed serial links, different 8B/10B encoders were suggested [2, 3, 4]. In this letter, a design of high-speed 8B/10B encoder by a simplified coding table is presented. The proposed encoder includes a modified disparity control block which also reduced the disparity control

DOI: 10.1587/elex.5.581
Received June 01, 2008
Accepted July 22, 2008
Published August 25, 2008

logic.

2 Proposed 8B/10B Encoder

2.1 Proposed encoder and coding table reduction algorithm

Block diagram for the proposed 8B/10B encoder is shown in Fig. 1. The block has simplified Pre_5B/6B encoder, Pre_3B/4B encoder, and modified disparity control block by the proposed reduction algorithm.

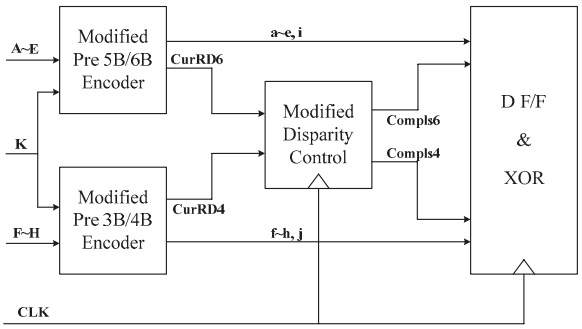

Fig. 1. Block diagram of proposed 8B/10B encoder

The coding table reduction algorithm is described as follows. As seen in the original 5B/6B encoding given in ref. [1], the first four bits of encoded outputs (denoted as 'abcd') are not changed except the corresponding bits of input data 'ABCD' are either all zero's or all one's. In order to reduce the number of input patterns to consider the first four bits ('ABCD') are added. And the input patterns are grouped with their addition result. Depending on the summing value of the four bits and the fifth bit 'E', a new simplified 5B/6B coding table is generated as shown Fig. 2 (a). By checking the other part of bit sequence ('f' and 'g' from 3B/4B encoding), the bit length of consecutive '1' or '0' should not be controlled under five. As a result, the conventional 31 different cases in 5B/6B encoding are reduced to only 10 encoding plus two special cases.

The encoded results from 5B/6B Pre_encoder are the encoded outputs ('abcdei') and the current running disparity value (CurRD). The output 'x' means that no encoding logic is required because input and encoded output patterns are the same. After encoding, the current running disparity value is assigned for disparity control block. Complementing operation depending on encoded values is processed in the final stage with D-F/F and XOR gates. The special cases (D.24, K.28) should be handled exceptionally in the proposed encoding scheme. Based on the proposed scheme the D.24 (00011) code will be encoded to '000111' instead of '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 the comma value. Therefore additional logic for handling D.24 and K.28 cases is required.

Fig. 2 (b) shows the simplified 3B/4B encoding table with the same reduction algorithm applied to the 5B/6B encoding reduction. The conventional

DOI: 10.1587/elex.5.581
Received June 01, 2008
Accepted July 22, 2008
Published August 25, 2008

13 encoding cases are reduced to only 7 encoding cases plus one exception. Here, it is also necessary to take care of the special code for preventing more than five consecutive same bit patterns.

(a)

E	Adder Result (for 4 input (A~D))	Encode Output						Cur RD6	E	Adder Result (for 4 input (A~D))	Encode Output						Cur RD6
		a	*b*	*c*	*d*	*e*	*i*				*a*	*b*	*c*	*d*	*e*	*i*	
0	0(3'b000)	x	1	1	x	0	0	-	1	0(3'b000)	x	1	1	x	1	1	+
0	1(3'b000)	x	x	x	x	1	0	-	1	1(3'b000)	x	x	x	x	1	1	0
0	2(3'b000)	x	x	x	x	0	1	0	1	2(3'b000)	x	x	x	x	1	0	0
0	3(3'b000)	x	x	x	x	0	0	0	1	3(3'b000)	x	x	x	x	1	0	+
0	4(3'b000)	x	0	x	0	0	0	-	1	4(3'b000)	x	0	x	0	1	1	+

(Except for D.24 case => abd=x, c=1, e,i == 00, CurRD6 = -, K.28 case => i=1, CurRD6 = + others same D.28)

(b)

(S+K)	H	Adder Result (input F,G)	Encode Output				Cur RD4
			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.24 case => abd=x, c=1, e,i == 00, CurRD6 = - K.28 case => i=1, CurRD6 = + others same D.28)

(c)

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

Fig. 2. (a) Simplified 5B/6B encoding table (b) Simplified 3B/4B encoding table (c) Modified disparity and complement decision table

2.2 Modified Disparity Control Block

In disparity control, the decision on whether complementing the pre-encoded output or not should be made and new running disparity values must be generated. In conventional scheme, those values are obtained from another long disparity coding table. However, in the proposed scheme, the new disparity value and complementing the final output are determined by only checking the polarity of the current and previous running disparity. By this scheme the overall operating speed can be improved. Current Running disparity values of 'CurRD6' and 'CurRD4' are coming from the modified Pre_5B/6B encoder and Pre_3B/4B encoder, respectively. The 'NextRD6' is generated from 5B/6B disparity check block by evaluating 'CurRD6' and 'PreRD4' and it is provided to the 3B/4B disparity check block. The 'NextRD4' is coming from 3B/4B disparity check block by evaluating 'CurRD4' and K value and it is provided to the 5B/6B disparity check block after one clock cycle. Thus 'CurRD4' becomes "PreRD4' after one clock cycle. The 'Compls6' and 'Compls4' signals are generated from each disparity check block. When 'Compls6 (or 4)' is equal to '1', then the encoder complements the pre-encoded output. The operation of the disparity control is summarized in Fig. 2 (c). Original

DOI: 10.1587/elex.5.581
Received June 01, 2008
Accepted July 22, 2008
Published August 25, 2008

23 different disparity coding cases are reduced to 9 general plus 3 special cases only depending on running disparities. For the K28.3 code, the disparity value in the previous encoding stage is minus (−) and the disparity value of current encoding stage becomes plus (+). Thus the encoder generates the final output without complementing since the sign of 'Compls4' is '0'. This results in a run length of 6 in the final encoded output of K28.3 code, which is not allowed in the 8B/10B encoding scheme. In order to solve the problem, the logic has to set the 'Compls4' bit for K28.3, K28.2 code, and the 'Compls6' bit for D7.7.P7 code after checking the disparity.

3 Results and Performance Analysis

The proposed encoder has been designed using and simulated. Logic Synthesis and P&R (Place and Route) are also performed using the CMOS 0.18 μm technology library. A PRBS (Pseudo Random Bit Sequence) input patterns are used to verify the correct encoding operation. The simulation on synthesized circuit has verified the operation on the proposed 8B/10B encoder. For the performance comparison, encoders that based on the original IBM's encoding table and Xilinx's scheme have been also designed and synthesized with the same 0.18 μm CMOS technology library. In order to compare the recent approach, the Xilink's encoder design was synthesized and simulated using 0.18 μm CMOS by using the RTL code given in the Xilink homepage. The Xilinx's encoder adopted a state-machine based operation to enhance the operating speed. In Table I, the performance comparison is given. It shows that the operating frequency of the proposed encoder is improved by 25.6% compared to IBM. And the operating frequency of the proposed encoder is improved by 3.9% and the cell area is decreased by 43% compared to the Xilinx approach. In terms of power consumption, the proposed scheme shows about 70% power reduction compared to IBM's. The Xilinx's encoder shows the comparable power consumption but it has a latency of five clock cycles. As a result, the proposed 8B/10B encoder shows the overall performance improvement.

Table I. Performance comparison

	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)	1.14	3.83	1.06

4 Conclusion

This work presents a design of 8B/10B encoder by the simplified coding table and a modified disparity control. The proposed encoder improved operating

DOI: 10.1587/elex.5.581
Received June 01, 2008
Accepted July 22, 2008
Published August 25, 2008

frequency and reduced the cell area at the same time. The circuit shows the operating frequency of 343 MHz with no latency and occupies area of 1886 μm^2 using CMOS 0.18 μm process. The proposed 8B/10B encoder shows the overall performance improvement compared to previous approaches.

Acknowledgments

This work was supported by Inha University. Hardware and software design tools were supported by IT-SOC and the IDEC program.

DOI: 10.1587/elex.5.581
Received June 01, 2008
Accepted July 22, 2008
Published August 25, 2008