논문 2008-45SD-4-7

코딩테이블 축소방법에 의한 8B/10B 인코더 설계

(8B/10B Encoder Design by Coding Table Reduction)

신 범 석*, 김 용 우**, 윤 광 섭***, 강 진 구***

(Beom-seok Shin, Yong-woo Kim, Kwang-sub Yoon, and Jin-ku Kang)

요 약

본 논문은 기존의 8B/10B 코딩테이블을 축소하여 단순화 방법에 의한 8B/10B 인코더 설계를 제안하였다. 제안하는 방법은 기존의 코딩 테이블을 덧셈기를 이용하여 축소하고 디스패리티 제어 블록의 알고리즘을 수정하였다. 제안한 인코더를 로직 시뮬레이션 및 로직 합성을 진행하여 Magna CMOS 0.18μm 공정에서 최대 동작 속도는 343MHz와 칩 면적 1886μm^2의 결과를 얻을 수 있었다.

Abstract

This paper presents a design of 8B/10B encoder by the coding table reduction. The proposed encoder has reduced coding table modified disparity control block. Logic simulation and synthesis have been done for the proposed design. After synthesized using Magna CMOS 0.18μm process, the proposed design achieved the operating frequency of 343MHz and chip area of 1886μm^2.

Keywords : 8B/10B, encoder, encoding table, disparity, run length

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, ServerNet, 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, defacto 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

* 정회원, 삼성종합기술원
(Assistant Research Staff, Samsung Advanced Institute of Technology)
** 학생회원, *** 정회원, 인하대학교 전자공학과 및 정보전자 공동연구소
(School of Electronic Engineering & Institute for Information and Electronics Research, Inha University)

※ This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R01-2006-000-10596-0). Hardware and software tools were supported by IT-SOC and the IDEC program.
접수일자: 2007년9월17일, 수정완료일: 2008년3월31일

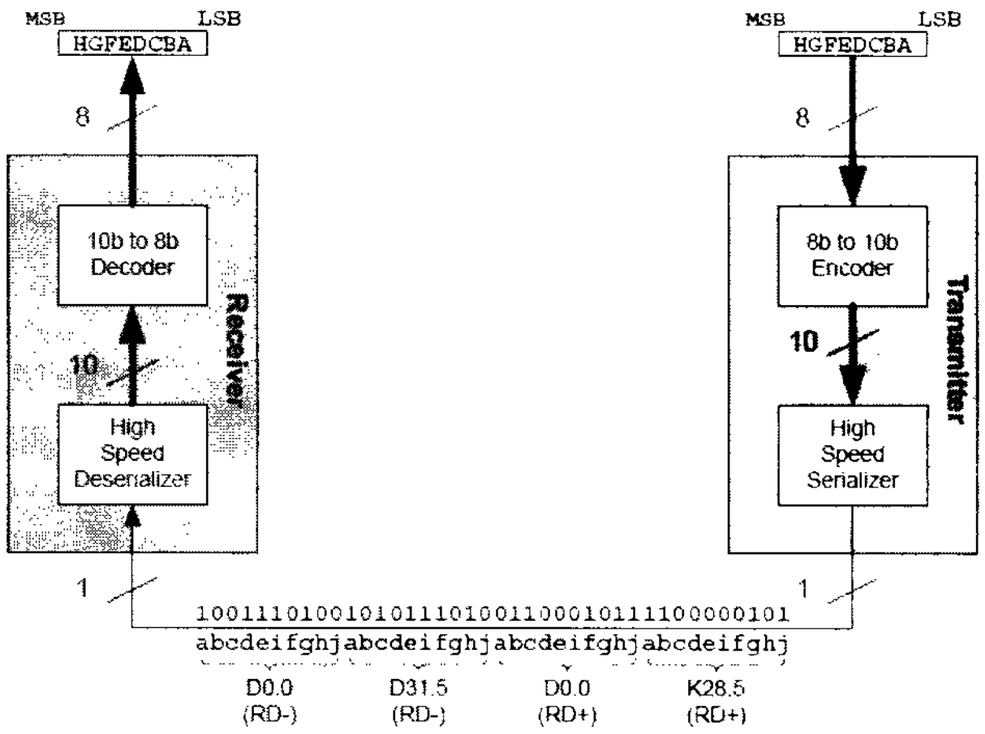

그림 1. 시스템에서 8B/10B 인코더/디코더 블록
Fig. 1. 8B/10B encoder/decoder block in communication system.

optimization, and utilizing a memory tables for coding and disparity inside ASIC or FPGA were investigated.[2~4]

This paper presents a design of 8B/10B encoder by coding table reduction. The proposed encoder has a reduced 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

1. 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 1.

The encoding table in Table 1 is most widely accepted for realizing the encoding function. However, implementation of the original encoding table given in Table 1 requires many logic stages, which lead to a low operating speed. These problems need to be improved for higher speed operation system applications.

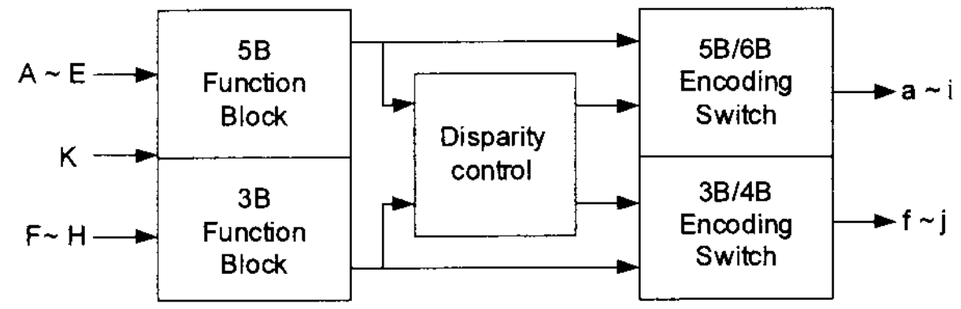

그림 2. 일반적인 8B/10B 인코더 블록도
Fig. 2. Block diagram of conventional 8B/10B encoder.

표 1. 일반적인 인코딩테이블 (5B/6B(위), 3B/4B(아래))
Table 1. Original Encoding table (5B/6B(Top), 3B/4B(Bottom)).

Name	ABCDEK	Classification		D-1	abcdei	D0	abcdei Alternate
		Bit encoding	Disparity				
D.0	00000 0	L04	L22'·L31'·E'	+	011000	-	100111
D.1	10000 0	L13·E'	L22'·L31'·E'	+	100010	-	011101
D.2	01000 0	L13·E'	L22'·L31'·E'	+	010010	-	101101
D.3	11000 0	L22·E'		^	110001	0	
D.4	00100 0	L13·E'	L22'·L31'·E'	+	001010	-	110101
D.5	10100 0	L22·E'		^	101001	0	
D.6	01100 0	L22·E'		^	011001	0	
D.7	11100 0		L31·D'·E'	-	111000	0	000111
D.8	00010 0	L13·E'	L22'·L31'·E'	+	000110	-	111001
D.9	10010 0	L22·E'		^	100101	0	
D.10	01010 0	L22·E'		^	010101	0	
D.11	11010 0			^	110100	0	
D.12	00110 0	L22·E'		^	001101	0	
D.13	10110 0			^	101100	0	
D.14	01110 0			^	011100	0	
D.15	11110 0	L40	L22'·L31'·E'	+	101000	-	010111
D.16	00001 0	L04, L04·E	L22'·L13'·E	-	011011	+	100100
D.17	10001 0	L13·D'·E		^	100011	0	
D.18	01001 0	L13·D'·E		^	010011	0	
D.19	11001 0			>	110010	0	
D.20	00101 0	L13·D'·E		^	001011	0	
D.21	10101 0			^	101010	0	
D.22	01101 0			^	011010	0	
D·K.23	11101 x		L22'·L13'·E	-	111010	+	000101
D.24	00011 0	L13·D·E	L13·D·E	+	001100	-	110011
D.25	10011 0			^	100110	0	
D.26	01011 0			^	010110	0	
D·K.27	11011 x		L22'·L13'·E	-	110110	+	001001
D.28	00111 0			^	001110	0	
K.28	00111 1	L22·K	k	-	001111	+	110000
D·K.29	10111 x		L22'·L13'·E	-	101110	+	010001
D/k.30	01111 x		L22'·L13'·E	-	011110	+	100001
D.31	11111 0	L40. L40·E	L22'·L13'·E	-	101011	+	010100

Name	FGHK	Classifications		D-1	fghj	D0	fghj Alternate
		Bit encoding	Disparity				
D/K.x.0	000x	F'·G'·H'	F'·G'	+	0100	-	1011
D.x.1	1000	(F≠G)·H'		>	1001	0	
D.x.2	0100	(F≠G)·H'		>	0101	0	
D/K.x.3	110x		F·G	-	1100	0	0011
D/K.x.4	001x		F'·G'	+	0010	-	1101
D.x.5	1010			>	1010	0	
D.x.6	0110			>	0110	0	
D.x.P7	1110		F·G. F·G·H	-	1110	+	0001
D/K.y.A7	111x	F·G·H·(S+K)	F·G. F·G·H	-	0111	+	1000
K.28.1	1001	(F≠G)·H'	(F≠G)·K	+	1001	0	0110
K.28.2	0101	(F≠G)·H'	(F≠G)·K	+	0101	0	1010
K.28.5	1011		(F≠G)·K	+	1010	0	0101
K.28.6	0111		(F≠G)·K	+	0110	0	1001

2. 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 1 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 reduced encoding table is given in Table 2.

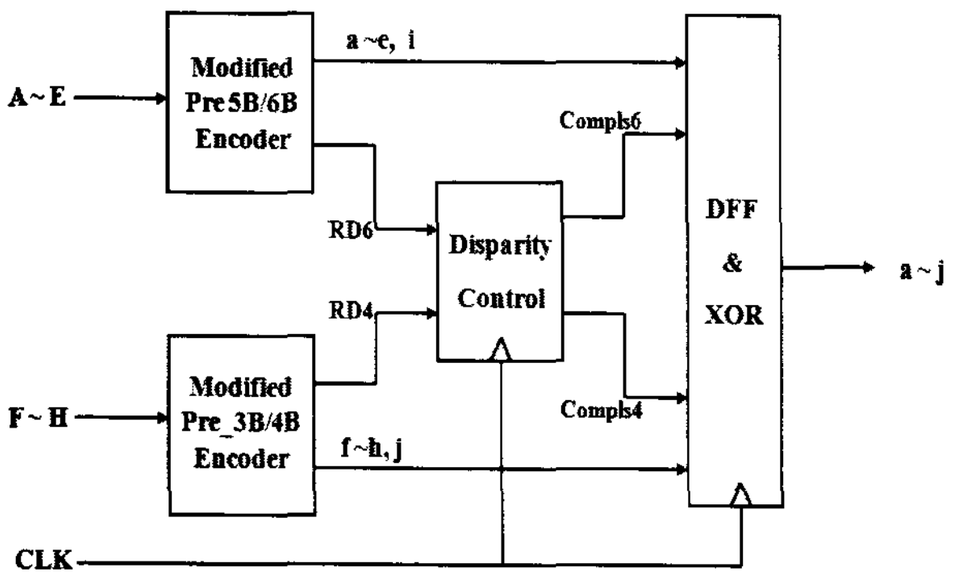

그림 3. 제안하는 8B/10B 인코더 블록도
Fig. 3. Block diagram of proposed 8B/10B encoder.

표 2. 축소된 5B/6B 인코딩 테이블
Table 2. Reduced 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'b001)	x	x	x	x	1	0	-
0	2(3'b010)	x	x	x	x	0	1	0
0	3(3'b011)	x	x	x	x	0	0	0
0	4(3'b100)	x	0	x	0	0	0	-
1	0(3'b000)	x	1	1	x	1	1	+
1	1(3'b001)	x	x	x	x	1	1	0
1	2(3'b010)	x	x	x	x	1	0	0
1	3(3'b011)	x	x	x	x	1	0	+
1	4(3'b100)	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)

In the Table 2, 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 3 shows the reduced 3B/4B encoding table with the reduction algorithm used in 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.

표 3. 축소된 3B/4B 인코딩 테이블
Table 3. Reduced 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)

3. Modified Disparity Control Block

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 5B/6B pre_encoder and 3B/4B pre_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 4.

Pre_RD and Cur_RD in Table 4 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 DP7.7.P7 code to be '1' after checking the disparity.

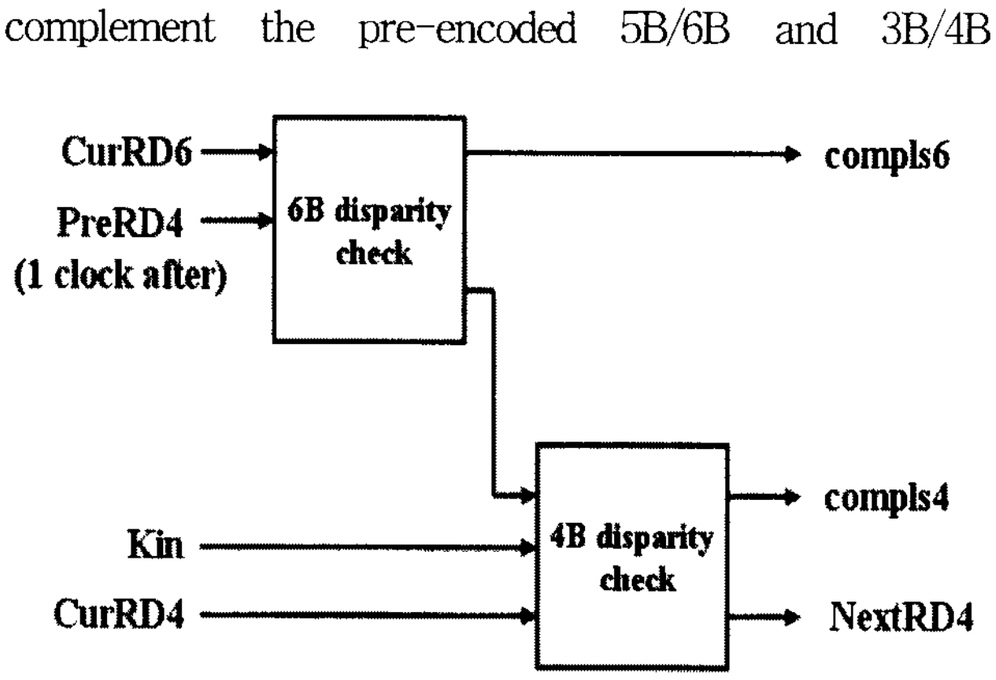

그림 4. 디스패리티 제어 블록도
Fig. 4. Block diagram of disparity control block.

표 4. 디스패리티/보수표현 결정 테이블
Table 4. Table for disparity and complement decision.

Pre RD	Cur RD6(4)	Compls6(4)	Next RD
–	–	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')

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 Magna CMOS 0.18 μm 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)

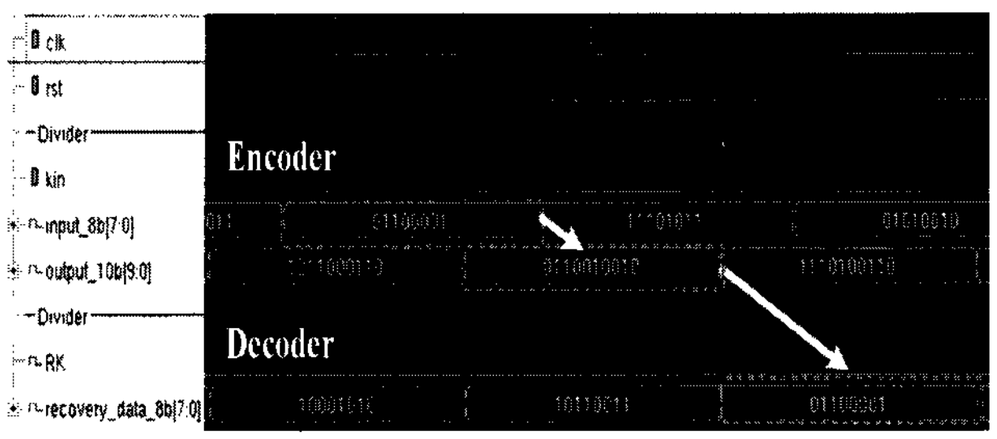

그림 5. 8B/10B 인코더 모의실험 결과
Fig. 5. 8B/10B Encoder/Decoder Simulation Result.

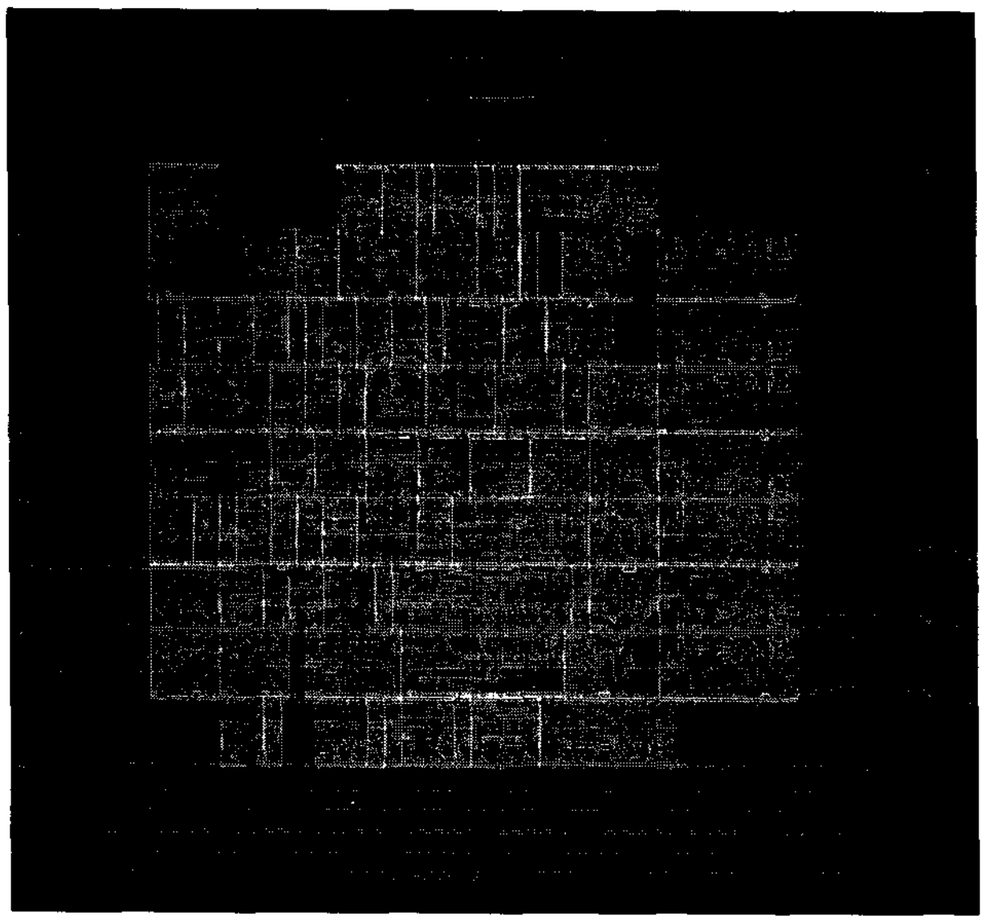

그림 6. 8B/10B 인코더 레이아웃
Fig. 6. 8B/10B encoder layout.

표 5. 제안하는 구조와 일반적인 구조의 성능비교
Table 5. Table for Performance Comparison.

	IBM	Xilinx	Proposed
Maximum Frequency (MHz)	273	330	343
Cell Area (μm^2)	1847	3308	1886

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 5, 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 8B10B encoder proposed by Xilinx[5] has been also compared. Xilinx used a state machine based algorithm. The operating frequency is improved by 3.9%, the cell area has been decreased to 43% compared to the Xilinx approach.

As a result, the proposed 8B/10B encoder verifies that the operating frequency and chip area can be improved by the reduced coding table.

Ⅳ. Conclusion

This work presents a design of 8B/10B encoder by the coding table reduction. The proposed encoder improved operating frequency and reduced the cell area at the same time. The operating frequency of 343MHz and cell area of 1886μm^2 are obtained when synthesized using Magna CMOS 0.18μm process.

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.

저 자 소 개

신 범 석(정회원)

2004년 인하대학교 전자공학과 학사 졸업
2006년 인하대학교 전자공학과 석사 졸업
2006년~현재 삼성종합기술원 연구원 재직
<주관심분야 : NoC, SoC>

윤 광 섭(정회원)

1981년 인하대학교 전자공학과 학사 졸업.
1983년 Georgia Institute of Technology 공학석사.
1989년 Georgia Institute of Technology 공학박사.
1989년 3월~1992년 2월 Silicon Systems Inc, Tustin Calif. U.S.A Senior Design Engineer
1992년 3월~현재 인하대학교 전자공학과 교수.
<주관심분야 : ADC, DAC, PLL 등 아날로그 및 혼성신호 집적회로 설계>

김 용 우(학생회원)

2007년 인하대학교 전자공학과 학사 졸업
2008년~현재 인하대학교 전자공학과 석사 과정
<주관심분야 : High Speed Interface 설계, SoC 설계>

강 진 구(정회원)

1983년 서울대학교 공학사
1990년 New Jersey Institute of Technology 전기 및 컴퓨터공학 석사
1996년 North Carolina State University 전기 및 컴퓨터 공학 박사
1983년~1988년 삼성전자(반도체)
1996년~1997년 미국 INTEL Senior Design Engineer
1997년 3월~현재 인하대학교 전자전기공학부 교수
<주관심분야 : 고속 CMOS 회로 설계, Mixed IC 설계, PLL/DLL/CDR, High Speed Interface IC, Display IC>