논문 98-23-10-3 한국통신학회논문지 '98-10 Vol.23 No.10

코드율 8/10의 (0, 6) MTR 코드 및 인코딩/디코딩 알고리즘

정회원 이 재 진*

A Rate 8/10 (0, 6) MTR Code and Its Encoding/Decoding Algorithm

Jaejin Lee* *Regular Member*

요 약

고밀도 스토리지 시스템에서 동작하는 시퀀스 검출방식(sequence detectors)의 거리 특성을 개선하기 위한 하나의 방법으로, 시퀀스 검출방식에서 가장 많은 에러를 발생시키는(다시 말해, 세 개이상의 연속되는 transitions을 갖는) 입력 패턴을 변조코딩 단계에서 아예 제거시키는 코드를 쓰면 된다. 이런 조건을 maximum-transition-run (MTR) 조건이라 한다. 이 논문에서는 코드율 8/10의 (0, 6) MTR 코드와 그것의 인코딩 및 디코딩 알고리즘에 대해 제안한다. MTR 조건을 만족하는 282 코드워드로 부터 선택된 256개의 코드워드를 갖는 코드가 복잡성을 최소화하도록 인코딩 및 디코딩 알고리즘을 구성하였다.

ABSTRACT

One way of improving the distance properties of sequence detectors operating at high user density is to eliminate the input patterns causing most errors in sequence detectors (i.e., three or more consecutive transitions). We refer this constraint to maximum transition-run(MTR) constraint.

This paper proposes a rate 8/10 (0, 6) MTR code and its encoder/decoder. The 256 codeword set selected out of 282 available codewords satisfying the MTR constraint is to minimize the complexity of its encoder and decoder.

I. Introduction

Recently, a (0, k) run-length limited(RLL) code having maximum transition run(MTR) constraint has been proposed [1]. Conventional (d, k) RLL codes constrain the minimum(maximum) number of consecutive zeros which is at least(most) d(k) between two neighboring ones. In addition, the MTR constraint, simply, does not allow the modulation coded sequence containing three or more consecutive transitions since those consecutive transitions mostly cause detection errors, especially, in high density storage systems. In other words, the MTR constraint improves the distance properties of sequence detectors operating at high user densities(e.g., EPRML or FDTS/DF). Moon and Brickner [1] confirmed that the rate 16/19 MTR code has 1.5-2 dB coding gain compared to the conventional 8/9 (0, k) codes which care mostly applied in recent HDD systems.

We have focused on byte input data, and an appropriate MTR code is the rate 8/10 (0, 6) MTR code. In Section II, we list all possible codewords satisfying the MTR constraint while they also satisfy d = 0 and k = 6 constraints. Then, the 256 codewords to represent byte data are heuristically selected in a direction to simplify the encoder and decoder. Section III presents the encoder/decoder combinational logic equations. Conclusion is drawn in Section IV.

II. The rate 8/10 (0, 6) MTR code

* 동국대학교 전자공학과(zlee@cakra.dongguk.ac.kr) 정회원
논문번호 : 97211-0621, 접수일자 : 1997년 6월 21일
※이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

There are many ways to find a codeword set, but we have found 282 codewords satisfying the MTR constraint and d = 0, k = 6 run-length constraints via computer search since the number of codewords needed for representing byte data which is our target is only 256.

When we search those, they must satisfy the constraints by themself and at the connecting boundaries between two codewords. The available codewords satisfying minimum and maximum run-length constraints of d = 0 and k = 6, respectively, and MTR constraint are listed in Table 1.

While we have investigated the 282 available codewords, we found that there are many codewords have some correlations. The proposed code listed in Table 2 can be partitioned into four groups (each group contains 64 codewords) so that one can directly assign two input bits. The third and eighth bit has the same in each group. For instance, codewords in the first group has $c_2 = 0$ and $c_7 = 0$, and codewords in the second group has $c_2 = 0$ and $c_7 = 1$, and so on when a code $\mathbf{c} = (c_0\ c_1\ c_2\ c_3\ c_4\ c_5\ c_6\ c_7\ c_8\ c_9)$. In addition, some blocks can share the same logic when we partition again each 64 codeword group into four 16 codeword subgroups, respectively, and so on.

Table 1. Available MTR codewords with d = 0, k = 6

	041	042	044	045	046	048	049	04A
04C	04D	051	052	054	055	056	058	059
05A	061	062	064	065	066	068	069	06A
06C	06D	081	082	084	085	086	088	089
08A	08C	08D	091	092	094	095	096	098
099	09A	0A1	0A2	0A4	0A5	0A6	0A8	0A9
0AA	0AC	0AD	0B1	0B2	0B4	0B5	0B6	0C1
0C2	0C4	0C5	0C6	0C8	0C9	0CA	0CC	0CD
0D1	0D2	0D4	0D5	0D6	0D8	0D9	0DA	102
104	105	106	108	109	10A	10C	10D	111
112	114	115	116	118	119	11A	121	122
124	125	126	128	129	12A	12C	12D	131
132	134	135	136	141	142	144	145	146
148	149	14A	14C	14D	151	152	154	155
156	158	159	15A	161	162	164	165	166
168	169	16A	16C	16D	181	182	184	185
186	188	189	18A	18C	18D	191	192	194
195	196	198	199	19A	1A1	1A2	1A4	1A5
1A6	1A8	1A9	1AA	1AC	1AD	1B1	1B2	1B4
1B5	1B6	204	205	206	208	209	20A	20C
20D	211	212	214	215	216	218	219	21A
221	222	224	225	226	228	229	22A	22C
22D	231	232	234	235	236	241	242	244
245	246	248	249	24A	24C	24D	251	252
254	255	256	258	259	25A	261	262	264
265	266	268	269	26A	26C	26D	281	282
284	285	286	288	289	28A	28C	28D	291
292	294	295	296	298	299	29A	2A1	2A2
2A4	2A5	2A6	2A8	2A9	2AA	2AC	2AD	2B1
2B2	2B4	2B5	2B6	2C1	2C2	2C4	2C5	2C6
2C8	2C9	2CA	2CC	2CD	2D1	2D2	2D4	2D5
2D6	2D8	2D9	2DA					

Table 2. 256 codewords for a rate 8/10 MTR code with d = 0, k = 6

041	042	048	049	04A	051	052	058
059	05A	061	062	068	069	06A	228
102	108	109	10A	111	112	118	119
11A	121	122	128	129	12A	131	132
241	242	248	249	24A	251	252	258
259	25A	261	262	268	269	26A	22A
141	142	148	149	14A	151	152	158
159	15A	161	162	168	169	16A	229
064	065	066	06C	164	165	166	16C
144	145	146	14C	14D	154	155	156
104	105	106	10C	10D	114	115	116
124	125	126	12C	12D	134	135	136
204	205	206	20C	20D	214	215	216
224	225	226	22C	22D	234	235	236
044	045	046	04C	04D	054	055	056
244	245	246	24C	24D	254	255	256
081	082	088	089	08A	091	092	098
099	0A1	0A2	0A8	0A9	0AA	0B1	0B2
181	182	188	189	18A	191	192	198
199	1A1	1A2	1A8	1A9	1AA	1B1	1B2
0C1	0C2	0C8	0C9	0CA	0D1	0D2	0D8
0D9	2C1	2C2	2C8	2C9	2CA	2D1	2D2
084	085	086	08C	08D	094	095	096
0A4	0A5	0A6	0AC	0AD	0B4	0B5	0B6
184	185	186	18C	18D	194	195	196
1A4	1A6	1A6	1AC	1AD	1B4	1B5	1B6
284	285	286	28C	28D	294	295	296
2A4	2A5	2A6	2AC	2AD	2B4	2B5	2B6
0C4	0C5	0C6	0CC	0CD	0D4	0D5	0D6
2C4	2C5	2C6	2CC	2CD	2D4	2D5	2D6

III. Encoder and decoder

1. Encoding algorithm

After we remove the third and eighth bit from the codewords, there are three subgroups such that c_4, c_5, c_6, c_8 and c_9 are the same pattern (we refer this to Module A, and it is shown in Table 3.) when $(m_0, m_1, m_2, m_3) = (0, 0, 0, 1)$ and $m_4m_5m_6m_7 \neq 1$. When $(m_0, m_1) \neq (0, 0)$, the output pattern of module Y in Table 5 frequently appears. The module X in Table 4 is for the output bits of c_6, c_8 and c_9 which are not shared. The corresponding combinational logic

equations to each module are given in (1)-(17), and the encoding rule is summarized in Figure 1.

$$a_1 = m_4 m_5 + m_4 m_6 \quad (1)$$

$$a_2 = \overline{m_4}\ \overline{m_5} + \overline{m_4}\ m_6\ \overline{m_7} \quad (2)$$

$$a_3 = \overline{m_4}\ m_6 + m_4 m_5 + m_4\ \overline{m_7} \quad (3)$$

$$a_4 = \overline{m_5}\ m_7 + \overline{m_6}\ m_7 \quad (4)$$

$$a_5 = \overline{m_5}\ \overline{m_7} + \overline{m_6}\ \overline{m_7} \quad (5)$$

$$b_1 = m_4 m_5 + m_4 m_6 + m_4 m_7 \quad (6)$$

$$b_2 = \overline{m_4}\ \overline{m_5} + \overline{m_5}\ \overline{m_6} + \overline{m_5}\ \overline{m_7} \quad (7)$$

$$b_3 = \overline{m_4}\ \overline{m_6} + \overline{m_4}\ m_7 + m_5\ \overline{m_6} + m_5 m_7 \quad (8)$$

$$b_4 = m_5 m_6 + m_6\ \overline{m_7}\, i + \overline{m_4}\ \overline{m_5}\ \overline{m_7} \quad (9)$$

$$b_5 = \overline{m_6}\ m_7 + m_4\ \overline{m_5}\ \overline{m_6} + m_4\ \overline{m_5}\ m_7 \quad (10)$$

$$x_1 = m_6 m_7 \quad (11)$$

$$x_2 = m_6\ \overline{m_7} \quad (12)$$

$$x_3 = \overline{m_6}\ m_7 \quad (13)$$

$$y_1 = m_5 m_6 + m_5 m_7 \quad (14)$$

$$y_2 = \overline{m_5}\ m_6 \quad (15)$$

$$y_3 = m_5\ \overline{m_7} \quad (16)$$

$$y_4 = m_6 m_7 + \overline{m_5}\ m_7 \quad (17)$$

Table 3. Module A and B for the encoder

input $m_4\ m_5\ m_6\ m_7$	*output* A $a_1\ a_2\ a_3\ a_4\ a_5$	*output* B $b_1\ b_2\ b_3\ b_4\ b_5$
0 0 0 0	0 1 0 0 1	0 1 1 1 0
0 0 0 1	0 1 0 1 0	0 1 1 0 1
0 0 1 0	0 1 1 0 1	0 1 0 1 0
0 0 1 1	0 1 1 1 0	0 1 1 0 0
0 1 0 0	0 0 0 0 1	0 0 1 0 0
0 1 0 1	0 0 0 1 0	0 0 1 0 1
0 1 1 0	0 1 1 0 0	0 0 0 1 0
0 1 1 1	0 0 1 0 0	0 0 1 1 0
1 0 0 0	0 0 1 0 1	0 1 0 0 1
1 0 0 1	0 0 1 1 0	1 1 0 0 1
1 0 1 0	1 0 0 0 1	1 1 0 1 0
1 0 1 1	1 0 0 1 0	1 0 0 0 1
1 1 0 0	1 0 1 0 1	1 0 1 0 0
1 1 0 1	1 0 1 1 0	1 0 1 0 1
1 1 1 0	1 0 1 0 0	1 0 0 1 0
1 1 1 1		1 0 1 1 0

Table 4. Module X for the encoder

input $m_6\ m_7$	*output* A $x_1\ x_2\ x_3$
0 0	0 0 0
0 1	0 0 1
1 0	0 1 0
1 1	1 0 0

Table 5. Module Y for the encoder

input $m_4\ m_5\ m_6\ m_7$	*output* A $a_1\ a_2\ a_3\ a_4\ a_5$
0 0 0 0	0 1 0 0 1
0 0 0 1	0 1 0 1 0
0 0 1 0	0 1 1 0 1
0 0 1 1	0 1 1 1 0
0 1 0 0	0 0 0 0 1
0 1 0 1	0 0 0 1 0
0 1 1 0	0 1 1 0 0
0 1 1 1	0 0 1 1 0

data bits condition			*corresponding output bits* $c_2 = m_0,\ c_7 = m_1$							
m_0m_1	m_2m_3	$m_4m_5m_6m_7$	c_0	c_1	c_3	c_4	c_5	c_6	c_8	c_9
00	00	=1	1	0	0	1	0	1	0	0
			0	0	1	a_1	a_2	a_3	a_4	a_5
	01	=1	1	0	0	1	0	1	0	1
			0	1	1	a_1	a_2	a_3	a_4	a_5
	10	=1	1	0	0	1	0	1	1	0
			1	0	1	a_1	a_2	a_3	a_4	a_5
	11		0	1	0	b_1	b_2	b_3	b_4	b_5
01	00	m_4=1	0	m_5	1	m_4	0	x_1	x_2	x_3
		m_4=0		1			y_1	y_2	y_3	y_4
	01		0	1	0	m_4				
	10		1	0	0	m_4				
	11		m_4	0	1	0				
10	00	m_4=0	0	0	0	m_5	m_6	m_4	m_7	m_7
		m_4=1				y_1	y_2		y_3	y_4
	01	m_4=0	0	1	0	m_5	m_6		m_7	m_7
		m_4=1				y_1	y_2		y_3	y_4
	10	m_4=0	1	0	0	m_5	m_6		m_7	m_7
		m_4=1				y_1	y_2		y_3	y_4
	11	m_4=0	m_5	0	1	0	m_6		m_7	m_7
		m_4=1	y_1				y_2		y_3	y_4
11	00		0	0	0	m_4	y_1	y_2	y_3	y_4
	01		0	1	0					
	10		1	0	0					
	11		m_4	0	1	0				

Figure 1. Encoding rule for the 8/10 (0, 6) MTR code

2. Decoding Algorithm

Decoding rule is just derived from the encoding rule as in Figure 2. Therefore, there are still many common circuits can be shared, for instance, (d_1, d_2, d_3, d_4) and (u_1, u_2, u_3) and (s_1, s_2, s_3) modules. The combinational logic equations for the decoder are given below:

$$d_1 = c_4 + \overline{c_5} c_6 \overline{c_8} c_9 + \overline{c_5} c_6 c_8 \overline{c_9} \tag{18}$$

$$d_2 = \overline{c_8}\,\overline{c_9} + \overline{c_4}\,\overline{c_5}\,\overline{c_6} + c_4 \overline{c_5} c_6 \tag{19}$$

channel bit's condition							*decoded bits* $m_0 = c_2$, $m_1 = c_7$					
c_2c_7	c_0	c_1	c_3	c_4	c_5	c_6	m_2	m_3	m_4	m_5	m_6	m_7
00	1	0	0	1	0	1	c_8	c_9	1	1	1	1
	0	0	1				0	0	d_1	d_2	d_3	d_4
	0	1	1				0	1				
	1	0	1				1	0				
	0	1	0				1	1	h_1	h_2	h_3	h_4
01	0	1	1	0			0	0	c_4	u_1	u_2	u_3
	0		1	1			0	0			v_1	v_2
	0	1	0				0	1		u_1	u_2	u_3
	1	0	0				1	0				
		0	1	0			1	1	c_0			
10	0	0	0			0	0	0	c_6	c_4	c_5	c_8
						1				s_1	s_2	s_3
	0	1	0			0	0	1		c_4	c_5	c_8
						1				s_1	s_2	s_3
	1	0	0			0	1	0		c_4	c_5	c_8
						1				s_1	s_2	s_3
		0	1	0		0	1	1		c_0	c_5	c_8
						1				r_1	r_2	r_3
11	0	0	0				0	0	c_4	u_1	u_2	u_3
	0	1	0				0	1				
	1	0	0				1	0				
		0	1	0			1	1	c_0			

Figure 2. Decoding rule for the 8/10 (0, 6) MTR code

$$d_3 = \overline{c_4} c_5 c_6 + c_4 \overline{c_5}\,\overline{c_6} + c_6 \overline{c_8}\,\overline{c_9} \tag{20}$$

$$d_4 = c_8 + \overline{c_4}\,\overline{c_5}\,\overline{c_8}\,\overline{c_9} \tag{21}$$

$$h_1 = c_4 + c_5 \overline{c_6}\,\overline{c_8} \tag{22}$$

$$h_2 = \overline{c_5} c_6 + \overline{c_5} c_8 \tag{23}$$

$$h_3 = \overline{c_5} c_8 + \overline{c_6} c_8 + \overline{c_5}\,\overline{c_6}\,\overline{c_8} + \overline{c_4} c_5 c_8 \overline{c_9} \tag{24}$$

$$h_4 = c_4 c_9 + c_6 c_9 + c_4 c_6 c_8 + \overline{c_4}\,\overline{c_5} c_6 c_8 + \overline{c_4} c_5 \overline{c_8}\,\overline{c_9} \tag{25}$$

$$u_1 = c_5 + c_8 \tag{26}$$

$$u_2 = c_6 + c_5 c_8 + c_5 c_9 \tag{27}$$

$$u_3 = c_9 + c_5 \overline{c_8} c_9 \tag{28}$$

$$s_1 = c_4 + c_8 \tag{29}$$

$$s_2 = c_5 + c_4 c_8 + c_4 c_9 \tag{30}$$

$$s_3 = c_9 + c_4 \overline{c_8}\,\overline{c_9} \tag{31}$$

$$r_1 = c_0 + c_8 \tag{32}$$

$$r_2 = c_5 + c_0 c_8 + c_0 c_9 \tag{33}$$

$$r_3 = c_9 + c_0 \overline{c_8}\,\overline{c_9} \tag{34}$$

$$v_1 = c_6 + c_8 \tag{35}$$

$$v_2 = c_6 + c_9 \tag{36}$$

IV. Conclusion

In this paper, we have proposed a rate 8/10 MTR code and its encoding and decoding algorithms. Since the MTR code does not contain three or more consecutive transitions, in high user density storage, the combination of MTR code and sequence detectors can be more robust to errors. The proposed code is a fixed length block code for byte input data, and it has the run-length constraints of d = 0 and k = 6. The codewords are selected to reduce the complexity of the encoder and decoder.

Reference

1. J. Moon and B. Brickner, "Maximum transition run codes for data storage systems," IEEE Trans. on Magnetics, vol. 32, no. 4, pp. 3992-3994, September 1996.

이 재 진(Jaejin Lee) 정회원
한국통신학회논문지 제22권 9호
현재 동국대학교 전자공학과 전임강사
<연구분야> 통신시스템, 스토리지시스템, 비밀통신