論文2001-38TC-8-1

고속 interconnection을 위한 NIBI 선로 부호

(NIBI Line Code for High-Speed Interconnection)

高 在 贊 *, 李 範 哲 **, 金 奉 秀 **, 崔 恩 暢 **

(Jae Chan Koh, Bhum Cheol Lee, Bong Soo Kim, and Eun Chang Choi)

요 약

본 논문에서는 전송 분야 뿐만 아니라 interconnection 분야에서 사용할 수 있는 새로운 선로 부호를 제안한다. 제안된 선로 부호는 1 비트의 잉여(redundancy) 비트를 사용하면서도 선로 부호가 갖는 기본적인 특징을 보장하며, interconnection 분야에서 필요한 byte 또는 frame 동기를 위한 직렬 동기 패턴 제공, 특수 문자 또는 in-band signaling을 제공한다. 8비트 이상의 병렬 데이터를 부호화 하거나 직렬 비트 스트림(stream)으로 전송하기 쉽게 하여 주는 제안된 NIBI 부호 생성 알고리즘, 복호 알고리즘 및 부호 성능에 대해서 기술한다.

Abstract

This paper describes new line code algorithm, called NIBI(Nibble Inversion Block Inversion) which is well suited for interconnection and transmission technology. The proposed line code which includes only one redundancy-bit serves primary features of line code and synchronization patterns for byte or frame synchronization in interconnection. Also, this line code provides in-band signals and special characters.

Ⅰ. 서 론

선로 부호는 전송 특성을 보장하고 전송에서 수신 성능을 높이기 위해서 사용되어져 왔다[1,2]. 최근에는 시스템 또는 시스템 내에 모듈의 동작 속도가 높아짐에 따라 이들 시스템 또는 모듈간을 상호 연결하는 interconnection 분야에서도 선로 부호를 요구하게 되었다. Interconnection 분야는 전송 분야와는 달리 복수개로 서로 연동되어 연결되며, 전송에 비해 짧은 거리로 연결되며, 고신뢰성을 요구한다. 이 분야에서 사용되는 선로 부호는 전송에서와 같이 직렬 전송 특성을 보장하고 전송에서 수신 성능을 높이기 위해서 사용된다[3,4]. 또한, interconnection 분야에서도 물리층 접속 매체로 광섬유를 사용하는 경우가 있다. 그러나, 전송에서의 광섬유가 장거리 전송을 위해서 사용되는 반면에 interconnection에서의 광섬유는 주로 전송 비트 속도를 높이기 위해서 사용된다.

본 논문에서는 전송 분야뿐만 아니라 특히 interconnection 분야에서 유용하게 사용할 수 있는 새로운 선로 부호를 제안한다. 제안된 선로 부호는 적은 잉여 비트를 사용하면서도 선로 부호가 갖는 기본적인 특징인 풍부한 천이(transition) 수, 낮은 디스패리티(disparity), 짧은 run zeros/ones length, 간단한 구현을 보장하며, interconnection 분야에서 필요한 byte 또는 frame 동기를 위한 유일한 직렬 비트 스트림을 갖는 동기 패턴 제공, 특수 문자 또는 in-band signaling을 제공한다.

본론에서는 제안된 NIBI(Nibble Inversion Block

* 正會員, 全南科學大學
(Chunnam Techno. College)
** 正會員, 韓國電子通信硏究院
(Electronics and Telecommunications Research Institute)
接受日字:2000年2月23日, 수정완료일:2001年7月10日

Inversion) 선로 부호 생성 알고리즘, 복호 알고리즘, NIBI 선로 부호의 성능에 대해서 기술한다.

Ⅱ. 본 론

종래의 선로 부호에 대한 요구 조건은 클럭 복구를 용이하게 하기 위한 직렬 데이터의 transition 수를 보장하고, AC 결합 또는 광섬유 사용을 위해 낮은 디스패리티와 DC 균형을 보장하고, 짧은 run zeros/ones length를 보장하는 것이었다. 또한, interconnection 분야에서의 선로 부호 요구 조건은 상기 요구 조건 외에 byte 및 frame 동기를 용이하게 구현하기 위해 직렬 비트 스트림에서 유일한 패턴들이 검출되는 것을 보장하고, in-band signaling이 가능하도록 특수 문자를 지원하는 것이었다.

그러나, 최근 통신 분야 및 데이콤 분야에서 고속화 대용량화의 경향은 선로 부호에 대해서 새로운 요구 조건을 요청하고 있다. 이러한 새로운 요구는 앞으로 선로 부호를 개발할 때에 고려되어야 하는 사항들이 될 수 있다.

특히, 통신 및 데이콤 분야 모두에서 사용되는 interconnection에서는 복수개의 직렬 interconnection간에 동기를 맞추는 word alignment 기능을 요구하고 있다. 이를 선로 부호에서 효율적으로 지원하려면 선로 부호는 여러 개의 직렬 동기 패턴을 지원해야 한다. 또한, interconnection 분야는 다양한 시스템 또는 모듈간에 접속이기 때문에 인터페이스 표준화가 어렵고 8비트 이상의 병렬 인터페이스를 요구하고 있다. 마지막으로, 데이터 복구 회로의 성능이 향상됨에 따라 높은 전송 효율을 기대할 수 있기 때문에 부호화 효율이 높은 선로 부호가 요구되고 있다.

본 논문에서 제안하는 NIBI 선로 부호는 $(n)B(n+1)B$ (n은 자연수 홀수)방식의 일종으로, 전송 시스템에서 많이 사용하고 있는 scrambled NRZ 부호와 달리 보장된 부호 성능을 지원한다. Scrambled NRZ 부호는 부호화 효율은 탁월하지만 비트 패턴에 따라 부호 성능이 변하는 특성을 갖고 있다. Gigabit Ethernet, Fiber Channel, Interconnection 분야에서 많이 사용되고 있는 8B10B 부호는 부호 성능이 보장되어 있고 다양한 응용을 위해 직렬 동기 패턴 및 특수 문자도 지원하지만 부호화 효율이 낮은 문제점이 있다.[3]

1. NIBI 선로 부호 생성(Encoding) 알고리즘

제안된 NIBI 선로 부호는 $(n)B(n+1)B$와 같이 일반화하여 기술할 수 있으나 $n=9$인 경우가 가장 일반적이기 때문에 본 논문에서 n은 9라고 가정하여 기술한다. 일반적으로 병렬 8비트가 가장 널리 사용되고 있기 때문에 1비트의 여유 비트를 사용자는 갖게 된다. 즉, NIBI 선로 부호는 병렬 9비트의 원시 데이터(source data)를 10비트의 부호 단어(code word)로 만든다.

제안된 NIBI 선로 부호 생성 알고리즘을 기술하기 전에 본 논문에서 사용되는 용어를 정리하면 다음과 같다. 9비트의 원시 데이터를 문자(character)라 정한다. 9비트 원시 데이터에 '0'으로 고정된 잉여 비트를 추가한 10비트를 사전 부호(pre-code)라 정한다. 원시 데이터 B(8:0)(t)=B_8, B_7, B_6, B_5, B_4, B_3, B_2, B_1, B_0라하고 초기 값이 0으로 고정된 잉여 비트를 NII(Nibble Inversion Indication) 비트라고 했을 때 사전 부호 Bpc(t)= B_8, B_7, B_6, B_5, B_4, B_3, B_2, B_1, B_0, $NII(=0_{initial})$가 된다. 본 논문에서 10비트는 블록(block)이라 하고 특별히 디스패리티가 0이 아닌 블록을 B(t)로 표기한다. 특히 최종적으로 선로 부호화 된 블록을 부호 단어(code word)라 정하고 Bcw(t)로 표기한다. 또한 블록의 홀수 비트 니블(odd bit nibble)은 B_7,B_5,B_3,B_1,NII 비트를 나타내고, 짝수 비트 니블(even bit nibble)은 B_8,B_6,B_4,B_2,B_0을 나타낸다. 니블 반전(nibble inversion)이라 함은 블록의 홀수 비트 니블이 반전됨을 나타내고 그 블록을 BNI(t)로 표기한다. 그리고, 블록 반전(block inversion)이라 함은 블록의 각 비트가 반전됨을 나타내고 그 블록을 BBI(t)로 표기한다. 그림 1에서 Dpc는 사전 부호의 디스패리티를 의미하고, Dni는 니블 반전된 사전 부호의 디스패리티를 의미하며, Dpco는 사전 부호의 홀수 비트 니블의 디스패리티를 나타낸다.

NIBI 선로 부호 생성 알고리즘을 여러 가지 유형으로 나누어서 유형별로 Bcw(t)를 생성하는 과정을 순차적으로 설명하면 다음과 같다. 제일 먼저 10 비트의 사전 부호에서 각 비트 1의 개수와 0의 개수 차인 디스패리티를 계산한다. 디스패리티가 0인 부호 단어를 최대로 생성하기 위해서 Dpc=0인 경우는 사전 부호를 그대로 부호 단어를 생성하는 type 1A, Dpc≠0인 사전 부호를 니블 반전하여 그 중에서 Dni=0인 부호 단어를

생성한다. 이는 type 2 중 126가지의 경우가 있다. 또한, 대역내 신호(in-band signal)인 특수 문자를 제공하기 위해서 Dpc=0이면서 니블 반전된 블록으로 부호 단어를 생성하는 type 1B, 부호 단어로 사용되는 디스패리티를 D라고 하였을 때, 디스패리티 허용 범위(|D|≤4)를 넘는 블록에 대해서 Dpc=0이면서 니블 반전된 블록 중에서 대역 내 신호 및 특수 문자 부호로 사용되는 블록을 제외한 디스패리티가 부호 단어 생성 허용 범위에 있는 블록이 되도록 특정 비트를 조작하여 맵핑(mapping)시킨 부호 단어를 생성하는 type 3, 4가 있다. 본 논문에서 제시한 NIBI 선로 부호 생성 알고리즘에서는 디스패리티 절대치가 4이하인 선로 부호를 생성한다.

디스패리티가 0이 아닌 블록 B(t)는 반전된 블록

표 1. 규칙적인 부호 단어 유형

Table 1. Regular coding type.

부호단어 생성유형	Source Data Bit Format	Code Word	Complement Cdoe Word	Disparity (D/-D)	부호단어 수
type 1A	B8,b7,b6,b5,b4,b3,b2,b1,b0	b8,b7,b6,b5,b4,b3,b2,b1,b0,*NII*	없음	0	126
type 1B	B8,b7,b6,b5,b4,b3,b2,b1,b0	b8,b7,b6,b5,b4,b3,b2,b1,b0,NII	b8,b7,b6,b5,b4,b3,b2,b1,b0,*NII*	±2	20
type 2	B8,b7,b6,b5,b4,b3,b2,b1,b0	b8,b7,b6,b5,b4,b3,b2,b1,b0,NII	없음	D=0	126
			b8,b7,b6,b5,b4,b3,b2,b1,b0,*NII*	D≠0 (D/-D)	230

표 2. 규칙성이 없는 부호 단어 유형

Table 2. Irregular coding type.

부호단어 생성유형	Source Data Bit Format	Code Word	Complement Code Word	Disparity (D/-D)	부호단어 수
type 3A	000010101	0001101111	1110010000	2/-2	9
	001000101	0011001111	1100110000	2/-2	
	001010001	0011100111	1100011000	2/-2	
	001010100	0011101101	1100010010	2/-2	
	100000101	1001001111	0110110000	2/-2	
	100010001	1001100111	0110011000	2/-2	
	100010100	1001101101	0110010010	2/-2	
	101000001	1011000111	0100111000	2/-2	
	101000100	1011001101	0100110010	2/-2	
type 3B	000101010	0100001011	1011110100	-2/2	10
	010001010	0001001011	1110110100	-2/2	
	010100010	0000011011	1111100100	-2/2	
	010101000	0000001111	1111110000	-2/2	
	101011111	1111100001	0000011110	2/-2	
	101110111	1110110001	0001001110	2/-2	
	101111101	1110100101	0001011010	2/-2	
	111010111	1011110001	0100001110	2/-2	
	111011101	1011100101	0100011010	2/-2	
	111110101	1010110101	0101001010	2/-2	
type 3C	101010000	1110001101	0001110010	2/-2	1
type 4A	001010101	0110000011	1001111100	-2/2	6
	100010101	1100000011	0011111100	-2/2	
	101010100	1110000001	0001111110	-2/2	
	101010111	1110000111	0001111000	2/-2	
	101011101	1110010011	0001101100	2/-2	
	101110101	1111000011	0000111100	2/-2	
type 4B	010101010	0011111001	1100000110	2/-2	4
	101000101	1100100111	0011011000	2/-2	
	101010001	1100001111	0011110000	2/-2	
	111010101	1000000111	0111111000	-2/2	

BBI(t)를 함께 갖는다. 여기서 BBI(t)는 B(t)의 보수 부호 단어 판이라 한다. 보수 부호 단어 판이란 디스패리티의 절대치는 같으면서 디스패리티의 +, -극성이 반대가 되는 블록이다. 디스패리티 누적 결과에 따라 원래의 디스패리티 부호(D)를 갖는 블록 B(t)를 부호 단어로 선택할 것인지 보수 부호(-D) 단어 판인 BBI(t)를 부호 단어로 선택할 것인지를 결정한다.

유형별로 분류된 NIBI 선로 부호 생성 방법은 A에서 기술하고, NIBI 선로 부호 생성 방법에 의해 생성된 블록 중 디스패리티가 0이 아닌 블록은 B(t)와 BBI(t)중

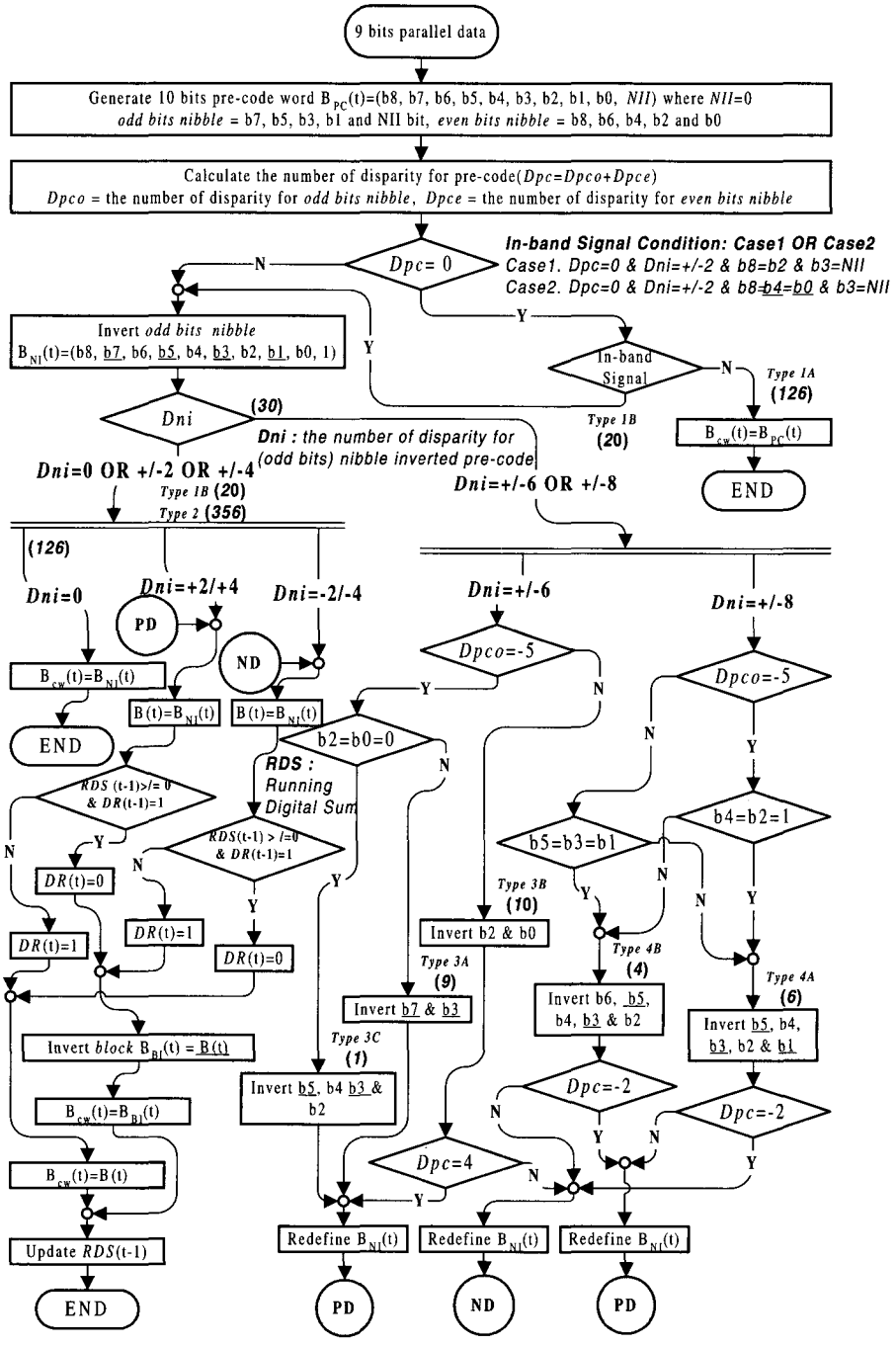


그림 1. NIBI 선로 부호 생성 과정 흐름도
Fig. 1. The NIBI encoding flow.

에서 +극성을 갖는 블록을 부호 단어로 선택할 것인지, -극성을 갖는 블록을 부호 단어로 선택할 것인지를 결정하는 디스패리티 +, -극성 결정 방법은 B에서 기술한다.

A. 유형(type)별 NIBI 선로 부호 생성

본 논문에서 부호 생성 및 복호 알고리즘의 이해를 돕기 위해서 표 1, 2와 같이 유형별로 나누어 설명한다.

표 1의 type 1, type 2는 논리적으로 규칙성을 가지고 있고, 표 2의 type 3, type 4는 논리적으로 규칙성을 가지고 있지 않다. 따라서, type3 및 type4에 속하는 문자는 Dpc=0이면서 사전 부호를 니블 반전하였을 때 디스패리티의 절대치가 4이하인 블록만 선택하여 그 선택된 블록으로 맵핑되도록 일일이 비트를 조작한다. 표 1에서 비트의 밑줄 표시는 반전(보수)을 의미한다. 예로 b7은 b7의 보수를 나타낸다. 그림 1에 표 1과 표 2에서 유형별로 나눈 NIBI 선로 부호 생성 알고리즘을 흐름도로 나타내었다.

그림 1과 같이, NIBI 선로 부호를 생성하는데, Dpc=0인 블록으로 부호 단어를 생성하는 유형(type 1), Dpc≠0이고 |Dni|≦4인 블록으로 부호 단어를 생성하는 유형(type 2), Dpc≠0이고 Dni=±6인 블록으로 부호 단어를 생성하는 유형(type 3)과 Dpc≠0이고 Dni=±8인 블록으로 부호 단어를 생성하는 유형(type 4)으로 각각 분류된다. 이렇게 분류된 유형을 NIBI 선로 부호 생성 알고리즘 흐름도를 사용하여 자세히 설명하면 다음과 같다.

1) type 1A

Dpc=0인 사전 부호를 그대로 부호 단어를 생성한다.

2) type 1B

대역내 신호를 제공하기 위해서 Dpc=0이면서 니블 반전된 블록 중에서 Dni=±2가 되는 블록 B(t)와 BBI(t)를 생성한다. 이는 표 3에서 자세히 나타낸다.

3) type 2

Dpc≠0인 사전 부호를 니블 반전하여 Dni=0, ±2, ±4인 블록을 생성하고, Dni≠0인 블록 B(t)는 반전된 블록 BBI(t)도 생성한다.

4) type 3A

Dni=±6이고 Dpco= -5이면서 B2 = B0 = 0이면 니블 반전된 블록에서 B7와 B3를 반전하여 B(t)와 BBI(t)를 생성한다. 여기서 B2=B0=0은 B2=B0=0의 논리적 부정 즉, B2≠0 or B0≠0 or B2≠B0을 의미한다.

5) type 3B

Dni=±6이고 Dpco -5이면 니블 반전된 블록에서 B2와 B0를 반전하여 B(t)와 BBI(t)를 생성한다.

6) type 3C

Dni=±6이고 Dpco= -5이면서 B2 = B0 = 0이면 니블 반전된 블록에서 B5, B4, B3, B2를 반전하여 B(t)와 BBI(t)를 생성한다.

7) type 4A

Dni=±8이고 ((Dpco= -5 이고 B4 = B2 = B1) 또는 (Dpco= -5 이고 B5 = B3 = B1))이면 니블 반전된 블록에서 B5, B4, B3, B2, B1을 반전하여 B(t)와 BBI(t)를 생성한다.

8) type 4B

Dni=±8이고 ((Dpco= -5 이고 B4 = B2 = B1) 또는 (Dpco -5 이고 B5 = B3 = B1))이면 니블 반전된 블록에서 B6, B5, B4, B3, B2를 반전하여 B(t)와 BBI(t)를 생성한다.

표 3. 대역내 신호 및 특수 문자 부호 단어
Table 3. In-band/special character codes.

Source Data Bit Format	Code word	Complement Code word	D/-D
000110111	0100111011	1011000100	2/-2
001110011	0110110011	1001001100	2/-2
010010111	0001111011	1110000100	2/-2
010110011	0000110011	1111001100	-2/2
010110101	0000111111	1111000000	2/-2
011010011	0011110011	1100001100	2/-2
011100011	0010010011	1101101100	-2/2
011110001	0010110111	1101001000	2/-2
011110010	0010110001	1101001110	-2/2
100100111	1100011011	0011100100	2/-2
100110110	1100111001	0011000110	2/-2
101100110	1110011001	0001100110	2/-2
110000111	1001011011	0110100100	2/-2
110010110	1001111001	0110000110	2/-2
110100101	1000011111	0111100000	2/-2
110100110	1000011001	0111100110	-2/2
110110100	1000111101	0111000010	2/-2
111000110	1011011001	0100100110	2/-2
111100010	1010010001	0101101110	-2/2
111100100	1010011101	0101100010	2/-2

동기 부호 생성은 상기의 방법을 따르지 않고 연속된 두 블록을 조합하여 블록 동기를 위해 유일한 패턴을 갖는 동기 부호를 생성한다. 이 동기 부호는 표 3에서의 대역내 신호와 동기 부호로 사용되어질 수 있

는 블록이 조합된 것이다. 표 4에서 동기 부호 패턴을 나타내었다.

표 4. 동기 부호 단어 패턴
Table 4. Synchronization characters and patterns.

Combination of Synchronization *Characters*		Bit Format of Synchronization Patterns
[i181s]	[···s]	$C_{i7}C_{i6}C_{i5}C_{i4}C_{i3}C_{i2}C_{i1}C_{i0}C_9C_8C_7C_6C_5C_4C_3C_2$ **(Complement Version)**
[i181s]	[46s]	0111111101111011 (1000000010000100)
[i181s]	[58s]	0111111101101111 (1000000010010000)
[i181s]	[139s]	0111111111011110 (1000000000100001)
[i181s]	[154s]	0111111111001111 (1000000000110000)
[i181s]	[163s]	0111111111110110 (1000000000001001)
[i181s]	[178s]	0111111111100111 (1000000000011000)
[i181s]	[226s]	0111111110110111 (1000000001001000)

표 4에서 [ ] 안의 s는 동기 패턴으로 사용이 가능하다는 것을 의미하고 i는 대역내 신호로 사용하고 있음을 의미한다. 그리고 숫자는 원시 데이터의 십진 값을 의미한다. 동기 패턴은 연속된 두개의 블록으로 생성되는데 이 경우, 두 블록 조합의 RDS(Running Digital Sum)는 +6 이거나 -6이다. 동기 패턴 [i181s]&[s]의 비트 열은 C_{i7}, C_{i6}, C_{i5}, C_{i4}, C_{i3}, C_{i2}, C_{i1}, C_{i0}, C_9, C_8, C_7, C_6, C_5, C_4, C_3, C_2로 구성되는데 여기서[i181s]의 비트 열은 C_{i9}, C_{i8}, C_{i7}, C_{i6}, C_{i5}, C_{i4}, C_{i3}, C_{i2}, C_{i1}, C_{i0}이고, [s]의 비트 열은 C_9, C_8, C_{i6}, C_{i5}, C_{i4}, C_{i3}, C_{i2}, C_{i1}, C_{i0}이다. 예로, [i181s] & [46s]에 의한 동기 패턴은 1000000010000100 또는 0111111101111011 이다.

B. 디스패리티 극성에 의한 부호 단어의 극성 결정과 동기 부호 극성 결정

시간 t인 순간에 누적되어 있는 디스패리티의 값을 표시하는 레지스터의 값을 RDS(t)라 하고, 그 순간 블록의 디스패리티 +,-극성을 표시하는 레지스터(disparity register)의 값을 DR(t)라고 하며, 그 다음 t+1인 순간의 RDS는 RDS(t+1)이고 그 순간의 DR은 DR(t+1)라 한다. 전송된 선로 부호의 RDS 절대치는 최소가 되게 하기 위해, 만약 RDS(t-1)이 값을 가지고 있다면 t인 순간에 생성된 디스패리티가 0이 아닌 블록 B(t)와 반전된 블록 BBI(t) 중에서 +극성의 디스패리티를 갖는 블록을 부호 단어로 선택한다. 그림 1의 PD(Positive Disparity process)와 ND(Negative Disparity process)에서 양수 디스패리티를 가진 블록은 PD과정을 거쳐 원래의 양수 디스패리티를 가지는 블록 B(t)와 음수 디스패리티를 가지는 반전된 블록 BBI(t) 중에서 어떤 디스패리티 극성을 가진 블록을 부호 단어로 생성할 것인지를 결정하게 되고 음수 디스패리티를 가진 블록은 ND과정을 거쳐 원래의 음수 디스패리티를 가지는 블록 B(t)과 양수 디스패리티를 가지는 반전된 블록 BBI(t) 중에서 어떤 디스패리티 극성을 가진 블록으로 부호 단어를 선택할 것인지를 결정한다. 그림 1의 PD와 ND에서 시작되는 디스패리티의 +, -극성 결정 과정과 이와는 다른 방법으로 디스패리티 극성을 결정하는 동기 부호의 극성 결정 방법을 요약하면 다음과 같은 규칙으로 수행된다.

1) RDS(t)>0이면, t+1인 순간에 블록 또는 반전된 블록에서 양수 디스패리티를 가진 블록을 선택하여 부호 단어를 결정한다.
2) RDS(t)<0이면, t+1인 순간에 블록 또는 반전된 블록에서 음수 디스패리티를 가진 블록을 선택하여 부호 단어를 결정한다.
3) RDS(t)=0이고 t인 순간에 디스패리티가 양의 값을 가지면(DR(t)=1), t+1인 순간에 블록 또는 반전된 블록에서 음수 디스패리티를 가진 블록을 선택하여 부호 단어를 결정한다.
4) RDS(t)=0이고 t인 순간에 디스패리티가 음의 값을 가지면(DR(t)=0), t+1인 순간에 블록 또는 반전된 블록에서 양수 디스패리티를 가진 블록을 선택하여 부호 단어를 결정한다.
5) 동기 부호인 경우에는 상기 부호화 규칙을 따르지 않고, 다음과 같은 별도의 부호화 규칙을 따른다.
   ① t순간에 선택된 동기 대역내 신호 부호 단어의 디스패리티가 +2이면, t+1인 순간에 선택되는 동기 부호 단어의 디스패리티는 +4이다.
   ② t순간에 선택된 동기 대역내 신호 부호 단어의 디스패리티가 -2이면, t+1인 순간에 선택되는 동기 부호 단어의 디스패리티는 -4이다.

2. NIBI 복호(Decoding) 알고리즘

그림 2의 NIBI 선로 부호 복호 과정을 설명하기에 앞서 용어를 설명하면, 부호 단어에 대한 디스패리티를 Dcw라 표기하고, 니블 반전된 부호 단어의 디스패리티를 Dni라 표기한다. CNI(t)는 부호 단어를 니블 반전한 블록을 나타내고 CBI(t)는 CNI(t)블록을 반전한 블록을 나타낸다.

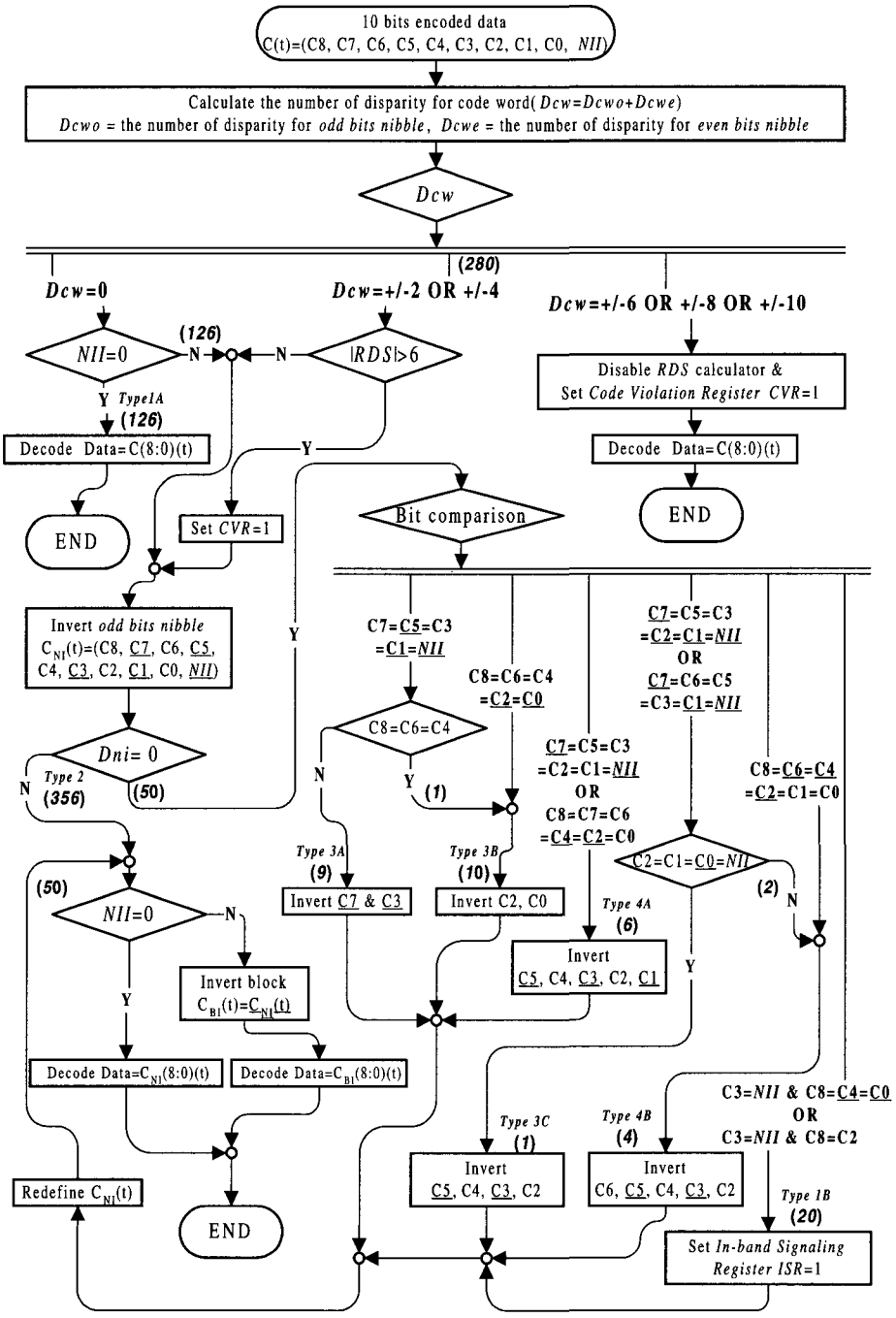


그림 2. NIBI 선로 부호 복호 과정 흐름도
Fig. 2. The NIBI decoding flow.

동기 부호 패턴을 검출하여 블록 동기가 맞추어진 이후에 NIBI 선로 부호 복호 과정이 진행된다고 가정한다. NIBI 선로 부호 복호 과정은 다음과 같다. 먼저 각 부호 단어에 대한 디스패리티를 계산한다. Dcw=0이고 NII=0인 부호 단어는 선로 부호 생성 과정에서 Dpc=0인 사전 부호로 부호 단어를 생성한 경우(type 1A)이므로 10비트의 부호 단어에서 NII비트를 제외하여 9비트의 원시 데이터로 복호한다. Dcw=0이고 NII=1인 부호 단어는 선로 부호 생성 과정에서 Dpc≠0인 사전 부호를 니블 반전하여 Dni=0인 경우의 사전 부호를 니블 반전한 블록으로 부호 단어를 생성한 경우(type 2 중 126개)이므로 복호 과정에서도 부호 단어를 니블 반전하면 선로 부호 생성 과정의 사전 부호가 복호된다. 이렇게 복호된 사전 부호 블록에서 NII비트를 제외하여 9비트 원시 데이터를 복호한다. 상기와 같이 NIBI 선로 부호 복호 과정은 NIBI 선로 부호 생성 방법의 역으로 복호한다.

그림 2에 NIBI 선로 부호 복호 알고리즘을 흐름도로 나타내었다. NIBI 선로 부호 생성 과정에서 유형별로 분류한 것과 같이 복호 과정도 유형별로 분류한다. 그림 2와 같이 유형별로 분류된 복호 방법은 다음과 같다.

1) type 1A

Dcw=0이고, NII=0이면 부호화 과정에서 디스패리티가 0인 사전 부호로 부호 단어를 생성한 경우이므로 C(t)에서 NII비트를 제외한 C(8:0)(t)로 원시데이터를 복호한다.

2) type 1B

Dcw=±2이면 부호 단어를 니블 반전한 후, Dni=0이면 각 비트를 비교한다. 비트의 구성이 (C3 = NII 이고 C8 = C4 = C0) 이거나 (C3 = NII 이고 C8 = C2)이면 그림 2와 같이 ISR 레지스터를 1로 설정하여 대역내 신호임을 알린 후, NII=0이면 $C_{NI}(8:0)(t)$로 원시 데이터를 복호하고, NII=1이면 $C_{BI}(8:0)(t)$로 원시 데이터를 복호한다.

3) type 2

Dcw=0이고 NII=1이거나 Dcw=±2,±4이면 부호 단어를 니블 반전한 후, 니블 반전된 블록($C_{NI}(t)$)의 Dni≠0이고 NII=0이면 $C_{NI}(8:0)(t)$로 원시 데이터를 복호하고, NII=1이면 $C_{BI}(8:0)(t)$로 원시 데이터를 복호한다.

4) type 3A

Dcw≠0이면 부호 단어를 니블 반전한 후, 니블 반전된 블록의 Dni=0이면 각 비트를 비교한다. 비트의 구성이 C7 = C5 = C3 = C1 = NII이고 C8 = C6 = C4이면 니블 반전된 블록의 C7과 C3 비트를 반전한 블록(CNI(t))을 생성하여 NII=0인 경우는 $C_{NI}(8:0)(t)$로 원시 데이터를 복호하고, NII=1인 경우는 $C_{BI}(8:0)(t)$로 원시 데이터를 복호한다.

5) type 3B

Dcw≠0이면 부호 단어를 니블 반전한 후, 니블 반전된 블록의 Dni=0이면 각 비트를 비교한다. 비트의 구성이 (C7 = C5 = C3 = C1 = NII이고 C8 = C6 = C4)이거나(C8 = C6 = C4 = C2 = C0)이면 니블 반전된 블록의 C2과 C0 비트를 반전한 블록($C_{NI}(t)$)을 생성하여 NII=0인 경우는 $C_{NI}(8:0)(t)$로 원시 데이터를 복호하고, NII=1인 경우는 $C_{BI}(8:0)(t)$로 원시 데이터를 복호한다.

6) type 3C

Dcw≠0이면 부호 단어를 니블 반전한 후, 니블 반전된 블록의 Dni=0이면 각 비트를 비교한다. 비트의 구성이 (C7 = C5 = C3 = C2 = C1 = NII 이거나 C7 = C6 = C5 = C3 = C1 = NII)이고 (C2 = C1 = C0 = NII)이면 니블 반전된 블록의 C5, C4, C3, C2 비트를 반전한 블록($C_{NI}(t)$)을 생성하여 NII=0인 경우는 $C_{NI}(8:0)(t)$로 원시 데이터를 복호하고, NII=1인 경우는 $C_{BI}(8:0)(t)$로 원시 데이터를 복호한다.

7) type 4A

Dcw≠0이면 부호 단어를 니블 반전한 후, Dni=0이면 각 비트를 비교한다. 비트의 구성이 (C7 = C5 = C3 = C2 = C1 = NII 이거나 C8 = C7 = C6 = C4 = C2 = C0)이면 C5, C4, C3, C2, C1 비트를 반전한 블록($C_{NI}(t)$)을 생성하여 NII=0인 경우는 $C_{NI}(8:0)(t)$로 원시 데이터를 복호하고, NII=1인 경우는 $C_{BI}(8:0)(t)$로 원시 데이터를 복호한다.

8) type 4B

Dcw≠0이면 부호 단어를 니블 반전한 후, Dni=0이면 각 비트를 비교한다. 비트의 구성이 ((C7 = C5 = C3 = C2 = C1 = NII 이거나 C7 = C6 = C5 = C3 = C1 = NII)이고 C2 = C1 = C0 = NII) 이거나 C8 = C6 = C4 = C2= C1= C0이면 C6, C5, C4, C3, C2 비트를 반전한 블록($C_{NI}(t)$)을 생성하여 NII=0인 경우는 $C_{NI}(8:0)(t)$로 원시 데이터를 복호하고, NII=1인 $C_{BI}(8:0)(t)$로 원시 데이터를 복호한다.

Dcw=±6,±8이거나 |RDS| > 6이면 그림 2에서와 같이 부호 위반을 표시하는 레지스터 CVR을 1로 설정

한다.

3. NIBI 선로 부호의 성능

NIBI 선로 부호 알고리즘에서는 부호 생성 과정에서 1비트의 잉여 비트만을 사용하고, 9비트 원시 데이터가 부호 단어가 되었을 때 RDS는 최대 6 및 최소 -6이며 (동기 부호 단어를 사용하지 않으면 최대 4 및 최소 -4), 최대 실행길이(0 또는 1의 최대 연속 비트열 수 ; running 0s or 1s length)는 15이며, 디지털 합 변량 (DSV)은 12이고, 이 디저털 합 변량은 동기 부호 단어를 사용하지 않으면 8이다. 종합적으로 분석을 하면, 앞서 기술한 NIBI 선로 부호의 특성은 8B10B 선로 부호에 비해서는 2~3배 성능 저하, CIMT(conditional invert with master transition) 선로 부호에 비해서는 2~3배 성능 개선이 된 것이며, 최근 반도체 집적화 기술의 발달로 데이터 복구 회로의 성능이 많이 개선되었기 때문에 $n=9$인 경우의 NIBI 선로 부호는 8B10B 부호에 대체되어 사용될 수 있음을 알 수 있다. 지면 관계로 여기서 기술하지는 않았지만 광 접속에서 문제시되는 AC 결합 시험인 2Gbps 데이터에 대해서 32MHz LPF(low pass filter)를 사용한 경우에서도 n=9인 경우의 NIBI 선로 부호는 8B10B 선로 부호에 비해 약 2% 정도의 eye pattern(eye height) 저하를 기록했다. 표 5는 기존의 8B/10B 선로 부호, CIMT 선로 부호와 NIBI 선로 부호를 비교한 것이다. 여기서 8B/10B 선로 부호는 2비트의 잉여비트를 통하여 8비트의 원시 데이터를 10비트의 부호 단어로 부호화하고 CIMT 선로 부호는 4비트의 잉여비트를 통하여 16비트의 원시 데이터를 20비트의 부호 단어로 부호화 한다. NIBI 선로 부호는 1비트의 잉여비트를 통하여 9비트의 원시데이터(n=9)를 10비트의 부호 단어로 부호화 한다. DSV, MRL, 동기문자 개수, 특수문자 개수 등에 관한 비교 결과도 표 5에서 자세히 나타내었다.

표 5. 기존의 선로 부호와 NIBI 선로 부호의 비교

Fig. 5. Comparison of conventional line code and NIBI line code.

성능 / 선로부호	잉여 비트 수	DSV	MRL	동기 문자 개수	특수 문자 개수
NIBI	1	12(8)	15	7	20
8B10B	2	6	5	3	9
CIMT	4	36	20	0	0

NIBI 선로 부호는 20개의 대역내 신호 또는 특수 문자를 제공하고, 직렬 비트 스트림에서 동기 패턴을 쉽게 찾기 위한 7종류의 동기 부호 단어를 제공한다. $n=9$라고 하였을 때 규칙성을 갖는 경우가 482개(94%)이고, 불규칙한 경우가 30개(6%)에 불과하다. 이는 원시 데이터의 비트 수가 바뀌어도 이 비율은 크게 변하지 않는다. 따라서 원시 데이터의 비트 수가 바뀌면 약 10%이내 부호 단어에 대해서 비트 조작 규칙을 다시 찾으면 되고 NIBI 선로 부호 생성 과정과 복호 과정이 서로 대칭성을 가지므로 하드웨어 구현이 상당히 용이하고 간단하게 된다.

III. 결 론

본 논문에서는 NIBI 부호 생성 알고리즘, NIBI 복호 알고리즘을 제안하고, NIBI 부호 성능에 대해서 알아보았다. 제안된 NIBI 부호는 종래의 선로 부호와 달리 1 비트의 잉여 비트만을 사용하고도 8B10B 부호에 상응하는 성능을 유지하였다. 보통의 원시 데이터가 8비트인 반면에 n=9인 경우에 NIBI 부호의 원시 데이터는 9비트로 운용되기 때문에 사용자는 1비트의 여유 비트를 확보할 수 있다. 또한, NIBI 부호는 풍부한 동기 패턴과 8비트 이상의 병렬 데이터를 부호화할 때도 부호 및 복호 알고리즘 체계가 유사하기 때문에 8비트 이상의 단어 정렬을 용이하게 구현할 수 있다.

참 고 문 헌

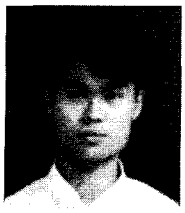
[1] R. M. Brooks, A. Jessop, "Line coding for optical fibra system," INT. J. Electronics, Vol. 55, No. 1, pp81-120, 1983.

[2] K. W. Cattermole, "Principles of digital line coding," INT. J. Electronics, Vol. 55, No. 1, pp3-33, 1983

[3] A. X. Widmer, P. A. Franaszek, "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code," IBM J. Res. Develop, Vol. 27, No. 5, 1983.

[4] Thomas Hornak, Richard C. Walker, "DC-free line code and bit and frame sunchronization for arbitrary data transmission," U.S. Pat. No. 5438621, 1995.

저 자 소 개



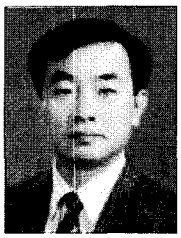
高 在 贊(正會員)

1994년 조선대학교 전자공학과 졸업(학사). 1996년 조선대학교 전자공학과 졸업(석사). 1999년~2000년 한국전자통신연구원(ETRI), 네트워크기술연구소, 차세대스위칭기술연구부, 고속스위치팀 위촉연구원. 현재, 전남과학대학 멀티미디어 정보과 전임강사

崔 恩 暢(正會員) 2월 論文 第 37卷 SD編 第 2號 參照
현재 한국전자통신연구원 고속스위치팀 선임연구원

李 範 哲(正會員)

1981년 경희대학교 전자공학과 졸업(학사). 1983년 연세대학원 전자공학과 졸업(석사). 1997년 연세대학원 전자공학과 졸업(박사). 1983년 한국전자통신연구원 입소. 현재, 한국전자통신연구원(ETRI), 네트워크기술연구소, 차세대스위칭기술연구부, 고속스위치팀장



金 奉 秀(正會員)

1982. 2 홍익대학교 전자공학과 졸업. 1984.2 홍익대학교 대학원 전자공학과 졸업 디지틀통신전공. 1984. 3~현재 한국전자통신연구원 근무. 라우팅프로토콜팀 책임연구원