RC24220 (W0703-097) March 21, 2007
Electrical Engineering

IBM Research Report

Circuit Implementation of a dc-Balanced 9B10B Transmission Line Code

Albert X. Widmer
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

IBM
Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Circuit Implementation of a dc-balanced 9B10B Transmission Line Code

Albert X. Widmer

IBM T.J. Watson Research Center
1101 Kitchawan Rd. Route 134, Yorktown Heights, NY 10598-0218
c/o:ritter@us.ibm.com

Abstract

This report describes a hardware implementation using combinational logic for the encoding and decoding circuits and the validity check of a dc-balanced 9B10B transmission line code similar to one described the in US Patent 6,614,369. No encoded data vector consists of a string of five 10 or five 01 bit patterns which is helpful for systems using differential encoding with decision feedback equalization (DFE). Vectors which require selective bit changes for encoding and decoding are confined to dc-balanced disparity independent vectors which have no alternate representation. About 350 inverting type primitive logic gates are required in each direction arranged in logic paths at most seven deep. The circuits have been structured so pipe-lining can be used with modest overhead to reduce the logic depth to 6, 5, 4, or even 3 per stage.

Table of Contents

I. INTRODUCTION

Ref. 1 describes the principles for the construction of a partitioned 16B16B transmission line code and Ref. 2 describes the components including a suitable 9B10B code. This report presents the encoding and decoding equations and the corresponding circuits a of a similar 9B10B code which has been significantly improved for easier implementation. There are also other applications for the 9B10B code, either alone or in combination with other balanced binary codes such as the 5B6B and 3B4B codes of Ref. 3 and 4 for the serialization of existing bus structures which are often not modulo 8 in width. A common bus width is 72 bits. The 9B10B code could be used to serialize such a bus on just eight channels rather than the nine needed with 8B10B code. For a better fit for a variety of data structures such as those of Ref. 5 and 6 which are demonstrated with 5B6B code, the codes of Ref. 2 have been slightly modified with minimal added complexity to help in the definition of suitable control and comma sequences for these other applications. These changes are also applicable for the 16B18B code so a single 9B10B macro can be built for all applications. To limit the error recovery time of decision feedback equalization (DFE) circuits, the two encoded vectors consisting of alternating ones and zeros have been defined as control vectors rather than data vectors.

A. Outline of Report

The first chapters describe notation and concepts used in defining and characterizing the code. This is followed by a description of the several sets of valid encoded vectors. Then the methodical assignment of source vectors to encoded vectors is done resulting in an encoding table. From the table, encoding equations for each of the encoded bits are derived in minimized form. Separate equations for compliance with the disparity rules are developed. Similarly, decoding equations and equations for validity checks are generated. From the encoding and disparity equations, an exemplary encoding circuit is constructed, followed by the construction of a decoding circuit and a circuit to detect invalid vectors.

B. Notation

Please note that the capital “B” in 9B10B refers to “Binary Symbol”, not bit, as a distinction from codes which use symbols with more than two levels, e.g. ternary symbols with three levels, commonly referred to by the capital letter “T”. Also, the number of inputs is actually ten to accommodate control characters, and the number 9 refers to the data vectors only.

The bits of the uncoded 9B data vectors are labelled with the upper case letters ‘ABCDEFGHI’ and the control input for special non-data characters is labelled with ‘K’. The bits of the coded 10B vectors are labelled with the lower case letters ‘abcdefghij’.

C. Disparity Diagrams (FIG. 1)

For easy reference, some of the trellis diagrams of Ref. 1 and 2 are reproduced here in slightly modified form as explained below. In the trellis diagrams such as shown in FIG. 1, an upwards sloping line for one interval represents a bit with a value of one, conversely, a slope downwards represents a zero. The horizontal coordinates on the time axis of the left trellis of FIG. 1 are labelled by a number in ascending order from left to right. Each unit increment represents one additional bit. The vertical coordinates which represent the running disparity are expressed by a lower case letter as follows:

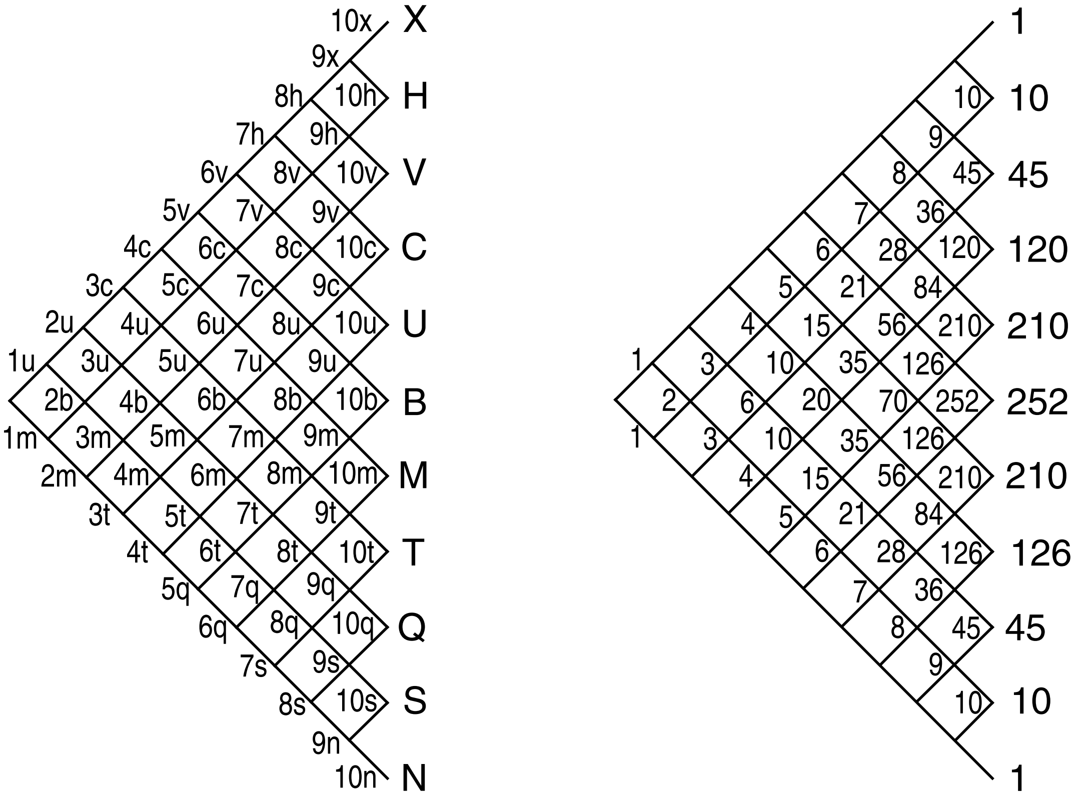

FIG. 1. Trellis Node Notation and Number of Vectors to Nodes for up to 10 Bits

- b (**b**alance) indicates a disparity of 0
- u (**u**p, **u**ni) indicates a disparity of +1 when paired with an odd preceding number and a disparity of +2 when paired with an even preceding number
- m (**m**inus) indicates a disparity of –1 when paired with an odd preceding number and a disparity of –2 when paired with an even preceding number
- c (**c**ube) indicates a disparity of + 3 when paired with an odd preceding number and a disparity of +4 when paired with an even preceding number
- t (**t**hree) indicates a disparity of –3 when paired with an odd preceding number and a disparity of –4 when paired with an even preceding number
- v (Roman numeral **V**) indicates a disparity of + 5 when paired with an odd preceding number and a disparity of +6 when paired with an even preceding number
- q (**q**uint) indicates a disparity of –5 when paired with an odd preceding number and a disparity of –6 when paired with an even preceding number
- h (**h**epta) indicates a disparity of +7 when paired with an odd preceding number and a disparity of +8 when paired with an even preceding number

- s (**s**even) indicates a disparity of –7 when paired with an odd preceding number and a disparity of –8 when paired with an even preceding number
- x (Roman numeral I**X**) indicates a disparity of +9 when paired with an odd preceding number and a disparity of +10 when paired with an even preceding number
- n (**n**ine, **n**egative) indicates a disparity of –9 when paired with an odd preceding number and a disparity of –10 when paired with an even preceding number

As an example, the expression “5c” in the left trellis of FIG. 1 refers to a disparity value of +3 after the end of the fifth bit (e) and the expression “6c” refers to a disparity value of +4 after the end of the sixth bit (f).

FIG. 1 shows the trellis diagrams for vectors comprising up to 10 bits. The left-side trellis lists the node names and is used to define the vector classifications and the right-side trellis shows the number of different paths or vectors leading from the origin to each node. Note that these numbers are identical to the binomial coefficients.

D. Vector Classification

The following notation is used for names attached to sets of source vectors or encoded vectors:

- The first capital letter B, P, D, or F indicates the disparity of the coded vectors:

  B indicates disparity independent **B**alanced coded vectors.

  P indicates a complementary pair of disparity dependent balanced coded vectors which are selected based on the **P**olarity of the running disparity.

  D indicates a complementary pair of coded vectors with a disparity of two.

  F indicates a complementary pair of coded vectors with a disparity of **F**our.
- A second capital letter, if present, indicates the block disparity of the uncoded vector or the vertical coordinate after bit 9 (I) in the left-side trellis of Fig. 1 using the capital version of the disparity values listed above.
- A third capital letter, if present, indicates the value of the control input bit K
- Up to three leading capital letters may be followed by one or more sets of a number paired with a lower case letter to indicate trellis nodes through which the members of the class must go, or not go if negated. Vectors going through negated nodes, e.g. 4t’, must not be part of the specified class of vectors. This notation is illustrated in the left-side trellis of Fig. 1.
- The third and following capital letters, other than K, mark the uncoded bits, if any, which must be complemented to obtain the respective coded primary vector. The last coded bit *j* is appended with a default value zero and complemented, if indicated by a classification name ending in J.

II. DESCRIPTION OF 9B10B CODE

At all 10B boundaries, the running disparity D can assume one of four values D=±1, or D=±3. Encoded vectors in this code are either balanced and disparity independent, *balanced and disparity dependent (new),* or have a disparity of ±2, or a disparity of ±4. If the current running disparity is positive (+1 or +3), only disparity independent vectors or vectors with a required positive entry disparity may be entered and complementary rules apply for a negative running disparity. Almost half the source vectors are translated into a single balanced disparity independent encoded vector. All other 9B vectors are translated into one of a pair of complementary 10B vectors, respectively, according to the disparity rules above.

A. 9B10B Code Definition

The 9B10B code comprises a total of *530* code points with 828 coded 10B vectors as illustrated by the trellis diagrams of FIG. 2.

1) 232 Balanced disparity independent 10B Vectors (FIG.2A.1)

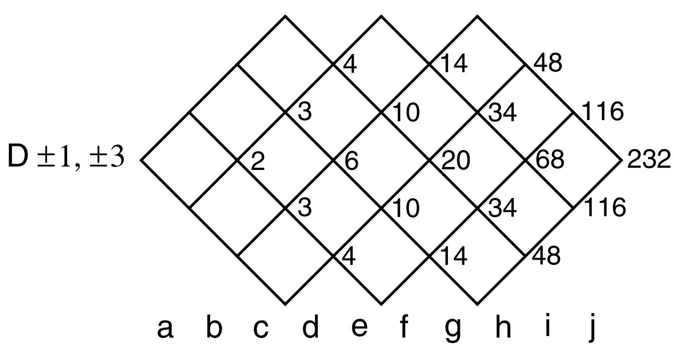

FIG. 2A.1

There are 232 disparity independent balanced vectors. Disparity independence means that they can be entered in a sequence regardless of the current starting disparity (one of the 4 values defined above). Balance means that the running disparities at the start and end of the vector are identical. The subset (232) of all possible 10B vectors (1024) chosen is the set of balanced vectors with a run length of no more than three at the leading and trailing boundaries as shown in FIG. 2A1.

2) 2x9 Balanced, Disparity Dependent 10B Vectors (FIG. 2A.2)

These 9 data vectors have been added as a partial replacement of 10 vectors from FIG. 2B which have been reassigned for control characters. For a negative running disparity, 8 balanced vectors with either four leading ones or four trailing zeros and one vector with both four leading ones and four trailing zeros are included. For a positive running disparity, the complementary vectors on the right side of FIG. 2A.2 are used.

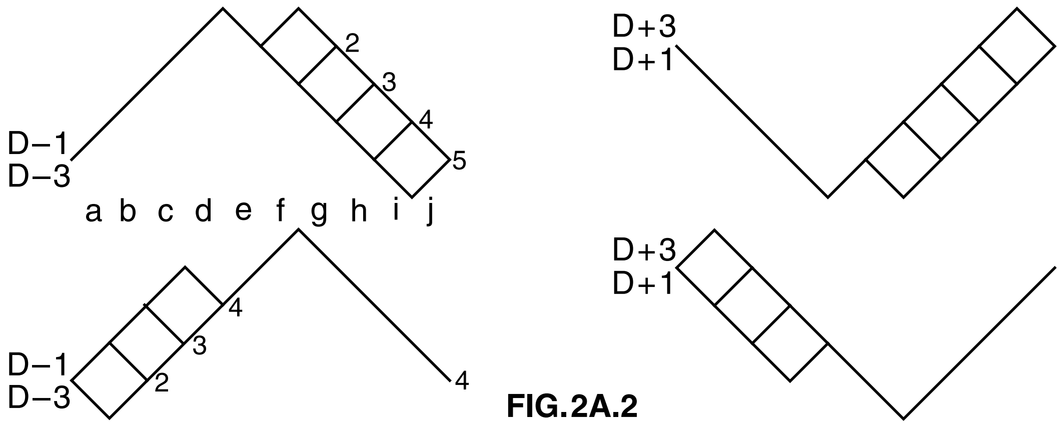

FIG.2A.2

3) 2x190 (180*) 10B Vectors with Disparity ±2 (FIG.2B)

A set of 190 10B vectors illustrated in FIG.2B comprises all bit patterns with a disparity of +2, a run length of no more than three at the front end and no more than three zeros or four ones at the trailing end. An exact complementary set of another 190 vectors on the right side has a disparity of −2.

*In FIG.2B, *the set of 10 vectors with four trailing ones is reserved for control characters* in the 16B18B environment and is not used for applications where it could generate false commas, e.g. for contiguous 10B vectors.

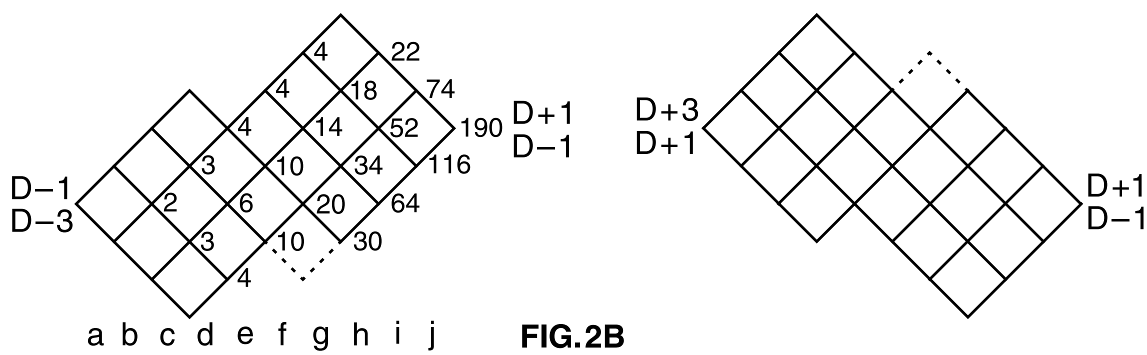

FIG.2B

4) 2x99 10B Vectors with Disparity ±4 (FIG.2C.1 and FIG.2C.2)

The set of 95 10B vectors of FIG.2C.1 comprises all bit patterns with a disparity of +4, no more than four ones or two zeros at the front end and no more than one zero or four ones at the trailing end. An exact complementary set of another 95 vectors on the right side has a disparity of −4.

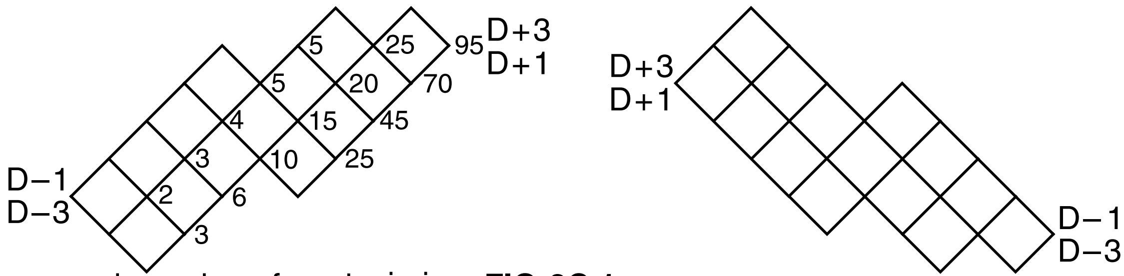

FIG.2C.1

The set of four 10B vectors of FIG.2C.2 comprises all bit patterns with a disparity of +4, no more than 3 ones or one zero at the front end and exactly two zeros at the trailing end. An exact complementary set of another 4 vectors on the right side has a disparity of −4.

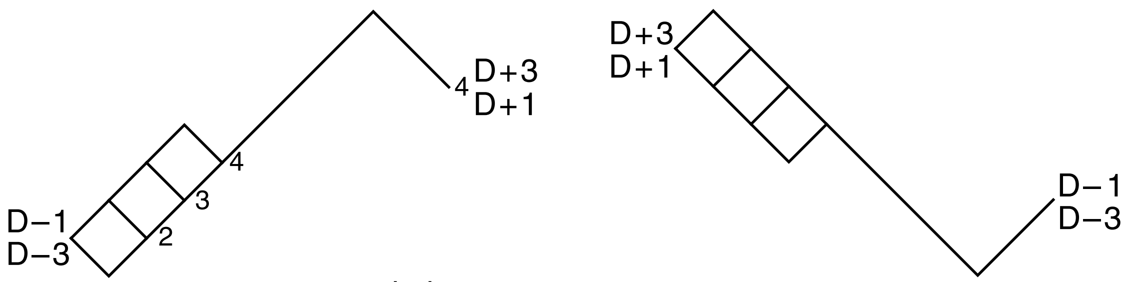

FIG.2C.2

Coded 10B vectors from the revised 9B10B code can be concatenated with 10B or 8B vectors without any change. The maximum run length remains at 7, and the digital sum variation is constrained to 12. The comma pattern is also unchanged as shown in FIG.3.

5) Control and Comma Characters

Table 1M lists seventeen 10B vectors Kx and C508 which can be reserved for information other than normal data. If any of the 18 control characters is to be encoded, a control line K must be asserted together with an appropriate data field. The control vector C508 is reserved for the generation of a singular comma sequence for quick synchronization. For the 16B18B code of Ref. 1, the comma extends over a first 10B field and the first three bits of the next following vector which may belong also to the 9B10B code, to the 7B8B code, or other similar compatible codes. The construction of a complete 18B comma character is discussed in Ref. 1 and Ref. 2.

6) Comma Characters for concatenated 9B10B Vectors (FIG.3)

FIG.3 below illustrates how the complete comma of either polarity fits into the trellis diagram. For purposes of the comma function, the possible location of the sequence at different disparity levels is irrelevant. The comma bit pattern is 00**1111111**0’**111** for a negative starting disparity, or its complement for a positive starting disparity. To acquire the 2-byte word synchronization, the circuits may limit the search to either one or both of the bit sequences ‘**1111111**x**111**’ and ‘**0000000**x**000**’, assuming a synchronization enabling circuit is activated only after a majority of misaligned commas has been received.

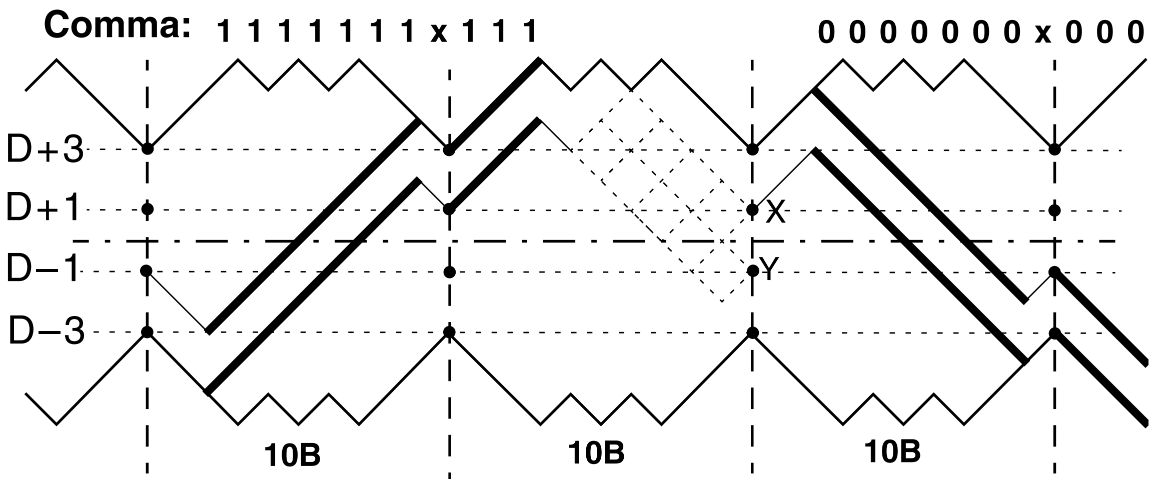

FIG.3

The input to the encoder should be the specified bit patterns, but for a preferred implementation, only the first source vector for C508 should be accompanied with a K value of one. The second part is provided by selected 10B vectors as follows:

Name	Coded Vector
D71	1110001000
D135	1110000100
D263	1110000010
D504	0001111110

a) Basic Set of 2-vector comma sequences

The C508 vector (0011111110/1100000001) can be paired with one of the disparity dependent vectors D71, D135, D263, or D504 to end at node Y in FIG. 3. Four different 20-bit control blocks which include the comma sequence can be generated regardless of the running disparity and without the special disparity controls needed for the second vector of the comma in the 16B18B code.

b) Extended Set of 2-vector comma sequences

If more than four 20-bit control blocks with a comma are useful, up to 14 additional ones can be provided using 14 balanced complementary vectors pairs with a leading run of three from the trellis of FIG. 2A.1. For the generation of the comma sequence, this subset of balanced 10B vectors must be made disparity dependent if they follow C508 of Table 1M, similar to what is done for balanced 4B vectors in 8B10B control characters of Ref. 3 or 4 and for the second part of the comma sequence of contiguous 7B8B vectors of Ref. 7. One or the other of the complements must be chosen depending on the polarity of the running disparity at the end of the C508 vector. This extended set is not included in the tables, equations and circuits of this report.

The primary and alternate 10B bit patterns from Table 1 suitable for comma generation together with the required polarity in front of the 10B vector are listed below:

D488	–	0001011110	+	1110100001	D23
D472	–	0001101110	+	1110010001	D39
D440	–	0001110110	+	1110001001	D0
D376	–	0001111010	+	1110000101	D503
D248	–	0001111100	+	1110000011	D7
D87	+	1110101000	–	0001010111	D40
D103	+	1110011000	–	0001100111	D24
D151	+	1110100100	–	0001011011	D495
D167	+	1110010100	–	0001101011	D8
D199	+	1110001100	–	0001110011	D264
D279	+	1110100010	–	0001011101	D239
D295	+	1110010010	–	0001101101	K216
D327	+	1110001010	–	0001110101	D136
D391	+	1110000110	–	0001111001	D72

The alternate vectors of the right column are decoded by full vector complementation if they contiguously follow C508.

B. Properties of the 9B10B Code

The important characteristics of the code can be directly extracted from the trellis diagram of FIG. 3 which also shows four possible configurations for the comma sequence. Using FIG. 3 together with the trellis diagrams defining the code (FIGS. 2x.y) one can verify that the comma sequence is singular, i.e. it cannot be reproduced in any other position relative to the vector boundaries neither within a 20B block nor across 20B block boundaries. Ref. 2 shows an identical comma sequence satisfying the singularity requirement for a 16B18B code comprising a 9B10B and a 7B8B part.

1) Clocking and Synchronization Parameters

The maximum run length is seven and no contiguous runs of seven are possible. The minimum transition density is two per 10B block for an indefinite length. The code includes a singular comma sequence.

2) Compatibility with Decision Feedback Equalization (DFE)

Any run of alternating ones and zeros in a sequence of data vectors is less than two vectors long. However, such sequences of length nX10 with an arbitrary n-value can be generated by a steady sequence of either the K170 or the K341 control character.

3) Low Frequency Characteristics

The code is DC balanced. The maximum digital sum variation is 12. The normalized DC offset as defined in reference 3 is 4.9. The low frequency cut-off point for high pass filters must be located about 2.5 times lower than for Fibre Channel 8B10B code for equal eye closure. The low frequency wander can be reduced on a statistical basis by *scrambling the data before encoding*. 8B10B coded, scrambled data can operate with a 50% higher low frequency cut-off point than a coded worst case pattern. For 16B18B code, the gain from scrambling before encoding is expected to be more.

4) 18B Control Characters

The 10B and 8B fields include 18 and 7 control characters, respectively, so it possible to generate a total of [(18x135) + (7x530)] = 6140 control characters in the 18-bit domain. The code includes four 18B comma sequences. Depending on the application, the user may relegate some of the unused control characters to the class of invalid vectors.

C. Encoding Table

Table 1 represents a specific coding assignment between uncoded and coded vectors in the 9B10B domain.

1) Design Principles

The coding tables are created in steps as follows:

1. Generate a list of all source vectors and all valid encoded vectors. Assume a default value for the appended bit. This design assumes a default value of zero. An alternate, equivalent code can be constructed by choosing complementary values for the appended bit and the vector sets.
2. In the coded domain, reserve the vector required for the comma generation (0011111110). Assign it a source vector which matches the first n-1 coded bits.
3. Assign all source vectors which match the first 9 bits of encoded vectors ending with the default value of j=0 to the respective matching vectors and remove them from both lists.
4. The remaining source vectors are assigned to the class of disparity independent balanced vectors which end with j=1, the complement of the default value. Assign the source vectors which match the first 9 bits of this set to the respective encoded vectors.
5. Find sets of several source vectors, preferably complementary sets, which can be made to match an encoded vector in this class by complementing just one common bit position in the source vector and make the assignment.
6. The remaining uncoded vectors are sorted into complementary pairs to the extent possible, and the remaining available encoded vectors are also sorted into pairs which are complementary in all or most of the leading 9 bits.
7. Find close matches between the two sets and change one or more bit positions in the source pair to obtain a match with the closest unassigned encoded pair.
8. Look for single vectors which can be made to match a coded vector by changing just one bit, then look for matches based on 2-bit changes, and so on.
9. Once all data vectors have been assigned, assign the remaining coded vectors to control characters and choose a corresponding source vector which matches the first n-1 bits.

9B10B Encoding

Name	ABCDEFGH I K	Coding Class	Primary a b c d e f g h i j	Pri DR	Pri DB
D0	000000000 x	BNABCGJ	**111**000**1**00**1**	±	0
D1	100000000 x	BS1uEGIJ	1000**1**0**1**0**11**	±	0
D2	010000000 x	BS1m2bEGIJ	0100**1**0**1**0**11**	±	0
D3	110000000 x	BQ4t'6mEGJ	1100**1**0**1**00**1**	±	0
D4	001000000 x	BS2m3mEGIJ	0010**1**0**1**0**11**	±	0
D5	101000000 x	BQ4t'6mEGJ	1010**1**0**1**00**1**	±	0
D6	011000000 x	BQ4t'6mEGJ	0110**1**0**1**00**1**	±	0
D7	111000000 x	BT1u5uIJ	11100000**11**	±	0
D8	000100000 x	BS3t4mEGIJ	0001**1**0**1**0**11**	±	0
D9	100100000 x	BQ4t'6mEGJ	1001**1**0**1**00**1**	±	0
D10	010100000 x	BQ4t'6mEGJ	0101**1**0**1**00**1**	±	0
D11	110100000 x	BT1u5uIJ	11010000**11**	±	0
D12	001100000 x	BQ4t'6mEGJ	0011**1**0**1**00**1**	±	0
D13	101100000 x	BT1u5uIJ	10110000**11**	±	0
D14	011100000 x	BT1m4uIJ	01110000**11**	±	0
D15	111100000 x	BM4cABFGJ	**00**110**11**00**1**	±	0
D16	000010000 x	BS4t5tADIJ	**1**00**1**1000**11**	±	0
D17	100010000 x	BQ4t'6mHIJ	1000100**111**	±	0
D18	010010000 x	BQ4t'6mHIJ	0100100**111**	±	0
D19	110010000 x	BT1u5uHJ	1100100**1**0**1**	±	0
D20	001010000 x	BQ4t'6mHIJ	0010100**111**	±	0
D21	101010000 x	BT1u5uHJ	1010100**1**0**1**	±	0
D22	011010000 x	BT1m3u4b5uHJ	0110100**1**0**1**	±	0
D23	111010000 x	BM4c'4t'6t'J	111010000**1**	±	0
D24	000110000 x	BQ4t'6mHIJ	0001100**111**	±	0
D25	100110000 x	BT1u5uHJ	1001100**1**0**1**	±	0
D26	010110000 x	BT1m2b3m5uHJ	0101100**1**0**1**	±	0
D27	110110000 x	BM4c'4t'6t'J	110110000**1**	±	0
D28	001110000 x	BT2m5uHJ	0011100**1**0**1**	±	0
D29	101110000 x	BM4c'4t'6t'J	101110000**1**	±	0
D30	011110000 x	BM4c'4t'6t'J	011110000**1**	±	0
D31	111110000 x	BU5vABIJ	**00**111000**11**	±	0
D32	000001000 x	BS5q6tADIJ	**1**00**1**0100**11**	±	0
D33	100001000 x	BQ4t'6mHIJ	1000010**111**	±	0
D34	010001000 x	BQ4t'6mHIJ	0100010**111**	±	0
D35	110001000 x	FT5u'5q'	1100010000	+	–4
D36	001001000 x	BQ4t'6mHIJ	0010010**111**	±	0
D37	101001000 x	FT5u'5q'	1010010000	+	–4
D38	011001000 x	FT5u'5q'	0110010000	+	–4
D39	111001000 0	BMK'4c'4t'6t'J	111001000**1**	±	0
D40	000101000 x	BQ4t'6mHIJ	0001010**111**	±	0
D41	100101000 x	FT5u'5q'	1001010000	+	–4

Table 1A

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D42	010101000 x	FT5u'5q'	0101010000	+	–4
D43	110101000 0	BMK'4c'4t'6t'J	110101000**1**	±	0
D44	001101000 x	FT5u'5q'	0011010000	+	–4
D45	101101000 0	BMK'4c'4t'6t'J	101101000**1**	±	0
D46	011101000 0	BMK'4c'4t'6t'J	011101000**1**	±	0
D47	111101000 x	PU4c5c	1111010000	–	0
D48	000011000 x	BQ4t6mBIJ	0**1**01100**11**	±	0
D49	100011000 x	FT5u'5q'	1000110000	+	–4
D50	010011000 x	FT5u'5q'	0100110000	+	–4
D51	110011000 0	BMK'4c'4t'6t'J	110011000**1**	±	0
D52	001011000 x	FT5u'5q'	0010110000	+	–4
D53	101011000 0	BMK'4c'4t'6t'J	101011000**1**	±	0
D54	011011000 0	BMK'4c'4t'6t'J	011011000**1**	±	0
D55	111011000 x	PU4u6c	1110110000	–	0
D56	000111000 x	FT5u'5q'	0001110000	+	–4
D57	100111000 0	BMK'4c'4t'6t'J	100111000**1**	±	0
D58	010111000 0	BMK'4c'4t'6t'J	010111000**1**	±	0
D59	110111000 x	PU4u6c	1101110000	–	0
D60	001111000 0	BMK'4c'4t'6t'J	001111000**1**	±	0
D61	101111000 x	PU4u6c	1011110000	–	0
D62	011111000 x	PU4u6c	0111110000	–	0
D63	111111000 x	BC6vABEIJ	**00**11**0**100**11**	±	0
D64	000000100 x	BS6q8qADIJ	**1**00**1**0010**11**	±	0
D65	100000100 x	BQ3m6t7tEFJ	1000**11**100**1**	±	0
D66	010000100 x	BQ3m6t7tEFJ	0100**11**100**1**	±	0
D67	110000100 x	FT5u'5q'	1100001000	+	–4
D68	001000100 x	BQ3m6t7tEFJ	0001**11**100**1**	±	0
D69	101000100 x	FT5u'5q'	1010001000	+	–4
D70	011000100 x	FT5u'5q'	0110001000	+	–4
D71	111000100 x	DM4t'6u'	1110001000	+	–2
D72	000100100 x	BQ3t4m6t7tEFJ	0001**11**100**1**	±	0
D73	100100100 x	FT5u'5q'	1001001000	+	–4
D74	010100100 x	FT5u'5q'	0101001000	+	–4
D75	110100100 x	DM4t'6u'	1101001000	+	–2
D76	001100100 x	FT5u'5q'	0011001000	+	–4
D77	101100100 0	DMK'4t'6u'	1011001000	+	–2
D78	011100100 x	DM4t'6u'	0111001000	+	–2
D79	111100100 x	PU4c5c	1111001000	–	0
D80	000010100 x	BQ4t6m'8tBCDEJ	0**1110**0100**1**	±	0
D81	100010100 x	FT5u'5q'	1000101000	+	–4
D82	010010100 x	FT5u'5q'	0100101000	+	–4
D83	110010100 x	DM4t'6u'	1100101000	+	–2

Table 1B

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D84	001010100 x	FT5u'5q'	0010101000	+	–4
D85	101010100 x	DM4t'6u'	1010101000	+	–2
D86	011010100 x	DM4t'6u'	0110101000	+	–2
D87	111010100 x	BU4c'6c'4t'	1110101000	±	0
D88	000110100 x	FT5u'5q'	0001101000	+	–4
D89	100110100 x	DM4t'6u'	1001101000	+	–2
D90	010110100 x	DM4t'6u'	0101101000	+	–2
D91	110110100 x	BU4c'6c'4t'	1101101000	±	0
D92	001110100 x	DM4t'6u'	0011101000	+	–2
D93	101110100 x	BU4c'6c'4t'	1011101000	±	0
D94	011110100 x	BU4c'6c'4t'	0111101000	±	0
D95	111110100 x	BC5v6cCEJ	11**0**1**0**0100**1**	±	0
D96	000001100 x	BQ4t6m'8tACJ	**1**0**1**001100**1**	±	0
D97	100001100 x	FT5u'5q'	1000011000	+	–4
D98	010001100 x	FT5u'5q'	0100011000	+	–4
D99	110001100 x	DM4t'6u'	1100011000	+	–2
D100	001001100 x	FT5u'5q'	0010011000	+	–4
D101	101001100 x	DM4t'6u'	1010011000	+	–2
D102	011001100 x	DM4t'6u'	0110011000	+	–2
D103	111001100 x	BU4c'6c'4t'	1110011000	±	0
D104	000101100 x	FT5u'5q'	0001011000	+	–4
D105	100101100 0	DMK'4t'6u'	1001011000	+	–2
D106	010101100 x	DM4t'6u'	0101011000	+	–2
D107	110101100 x	BU4c'6c'4t'	1101011000	±	0
D108	001101100 x	DM4t'6u'	0011011000	+	–2
D109	101101100 x	BU4c'6c'4t'	1011011000	±	0
D110	011101100 x	BU4c'6c'4t'	0111011000	±	0
D111	111101100 x	BC4c5c7u'CDJ	11**00**01100**1**	±	0
D112	000011100 x	FT5u'5q'	0000111000	+	–4
D113	100011100 x	DM4t'6u'	1000111000	+	–2
D114	010011100 x	DM4t'6u'	0100111000	+	–2
D115	110011100 x	BU4c'6c'4t'	1100111000	±	0
D116	001011100 x	DM4t'6u'	0010111000	+	–2
D117	101011100 x	BU4c'6c'4t'	1010111000	±	0
D118	011011100 x	BU4c'6c'4t'	0110111000	±	0
D119	111011100 x	DC4c'	1110111000	–	+2
D120	000111100 x	DM4t'6u'	0001111000	+	–2
D121	100111100 x	BU4c'6c'4t'	1001111000	±	0
D122	010111100 x	BU4c'6c'4t'	0101111000	±	0
D123	110111100 x	DC4c'	1101111000	–	+2
D124	001111100 x	BU4c'6c'4t'	0011111000	±	0
D125	101111100 x	DC4c'	1011111000	–	+2

Table 1C

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D126	011111100 x	DC4c’	0111111000	–	+2
D127	111111100 x	BV6v8vACEJ	0101011001	±	0
D128	000000010 x	BS6q8qADIJ	1001000111	±	0
D129	100000010 x	BQ4m7q8tEFJ	1000110101	±	0
D130	010000010 x	BQ4m7q8tEFJ	0100110101	±	0
D131	110000010 x	FT5u’5q’	1100000100	+	–4
D132	001000010 x	BQ4m7q8tEFJ	0010110101	±	0
D133	101000010 x	FT5u’5q’	1010000100	+	–4
D134	011000010 x	FT5u’5q’	0110000100	+	–4
D135	111000010 x	DM4t’6u’	1110000100	+	–2
D136	000100010 x	BQ4m7q8tEFJ	0001110101	±	0
D137	100100010 x	FT5u’5q’	1001000100	+	–4
D138	010100010 x	FT5u’5q’	0101000100	+	–4
D139	110100010 x	DM4t’6u’	1101000100	+	–2
D140	001100010 x	FT5u’5q’	0011000100	+	–4
D141	101100010 x	DM4t’6u’	1011000100	+	–2
D142	011100010 x	DM4t’6u’	0111000100	+	–2
D143	111100010 x	PU4c5c	1111000100	–	0
D144	000010010 x	BQ4t6m’8tCGJ	0010101101	±	0
D145	100010010 x	FT5u’5q’	1000100100	+	–4
D146	010010010 x	FT5u’5q’	0100100100	+	–4
D147	110010010 x	DM4t’6u’	1100100100	+	–2
D148	001010010 x	FT5u’5q’	0010100100	+	–4
D149	101010010 x	DM4t’6u’	1010100100	+	–2
D150	011010010 x	DM4t’6u’	0110100100	+	–2
D151	111010010 x	BU4c’6c’4t’	1110100100	±	0
D152	000110010 x	FT5u’5q’	0001100100	+	–4
D153	100110010 x	DM4t’6u’	1001100100	+	–2
D154	010110010 x	DM4t’6u’	0101100100	+	–2
D155	110110010 x	BU4c’6c’4t’	1101100100	±	0
D156	001110010 x	DM4t’6u’	0011100100	+	–2
D157	101110010 x	BU4c’6c’4t’	1011100100	±	0
D158	011110010 x	BU4c’6c’4t’	0111100100	±	0
D159	111110010 x	BC5v6cABEGJ	0011001101	±	0
D160	000001010 x	BQ4t6m’8tACJ	1010010101	±	0
D161	100001010 x	FT5u’5q’	1000010100	+	–4
D162	010001010 x	FT5u’5q’	0100010100	+	–4
D163	110001010 x	DM4t’6u’	1100010100	+	–2
D164	001001010 x	FT5u’5q’	0010010100	+	–4
D165	101001010 x	DM4t’6u’	1010010100	+	–2
D166	011001010 x	DM4t’6u’	0110010100	+	–2
D167	111001010 x	BU4c’6c’4t’	1110010100	±	0

Table 1D

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D168	000101010 x	FT5u'5q'	0001010100	+	–4
D169	100101010 x	DM4t'6u'	1001010100	+	–2
D170	010101010 0	DMK'4t'6u'	0101010100	+	–2
D171	110101010 x	BU4c'6c'4t'	1101010100	±	0
D172	001101010 x	DM4t'6u'	0011010100	+	–2
D173	101101010 x	BU4c'6c'4t'	1011010100	±	0
D174	011101010 x	BU4c'6c'4t'	0111010100	±	0
D175	111101010 x	BC4c5c7u'CDJ	11**00**01010**1**	±	0
D176	000011010 x	FT5u'5q'	0000110100	+	–4
D177	100011010 x	DM4t'6u'	1000110100	+	–2
D178	010011010 x	DM4t'6u'	0100110100	+	–2
D179	110011010 x	BU4c'6c'4t'	1100110100	±	0
D180	001011010 x	DM4t'6u'	0010110100	+	–2
D181	101011010 x	BU4c'6c'4t'	1010110100	±	0
D182	011011010 x	BU4c'6c'4t'	0110110100	±	0
D183	111011010 x	DC4c'	1110110100	–	+2
D184	000111010 x	DM4t'6u'	0001110100	+	–2
D185	100111010 x	BU4c'6c'4t'	1001110100	±	0
D186	010111010 x	BU4c'6c'4t'	0101110100	±	0
D187	110111010 x	DC4c'	1101110100	–	+2
D188	001111010 x	BU4c'6c'4t'	0011110100	±	0
D189	101111010 x	DC4c'	1011110100	–	+2
D190	011111010 x	DC4c'	0111110100	–	+2
D191	111111010 x	BV6v8vADEJ	**0**11**00**1010**1**	±	0
D192	000000110 x	BQ4t6m'8tACJ	**1**0**1**000110**1**	±	0
D193	100000110 x	FT5u'5q'	1000001100	+	–4
D194	010000110 x	FT5u'5q'	0100001100	+	–4
D195	110000110 x	DM4t'6u'	1100001100	+	–2
D196	001000110 x	FT5u'5q'	0010001100	+	–4
D197	101000110 x	DM4t'6u'	1010001100	+	–2
D198	011000110 x	DM4t'6u'	0110001100	+	–2
D199	111000110 x	BU4c'6c'4t'	1110001100	±	0
D200	000100110 x	FT5u'5q'	0001001100	+	–4
D201	100100110 0	DMK'4t'6u'	1001001100	+	–2
D202	010100110 x	DM4t'6u'	0101001100	+	–2
D203	110100110 x	BU4c'6c'4t'	1101001100	±	0
D204	001100110 x	DM4t'6u'	0011001100	+	–2
D205	101100110 x	BU4c'6c'4t'	1011001100	±	0
D206	011100110 x	BU4c'6c'4t'	0111001100	±	0
D207	111100110 x	DC4c5c7u'CDJ	11**00**00110**1**	±	0
D208	000010110 x	FT5u'5q'	0000101100	+	–4
D209	100010110 0	DMK'4t'6u'	1000101100	+	–2

Table 1E

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D210	010010110 x	DM4t'6u'	0100101100	+	–2
D211	110010110 x	BU4c'6c'4t'	1100101100	±	0
D212	001010110 x	DM4t'6u'	0010101100	+	–2
D213	101010110 x	BU4c'6c'4t'	1010101100	±	0
D214	011010110 x	BU4c'6c'4t'	0110101100	±	0
D215	111010110 x	DC4c'	1110101100	–	+2
D216	000110110 0	DMK'4t'6u'	0001101100	+	–2
D217	100110110 x	BU4c'6c'4t'	1001101100	±	0
D218	010110110 x	BU4c'6c'4t'	0101101100	±	0
D219	110110110 x	DC4c'	1101101100	–	+2
D220	001110110 x	BU4c'6c'4t'	0011101100	±	0
D221	101110110 x	DC4c'	1011101100	–	+2
D222	011110110 x	DC4c'	0111101100	–	+2
D223	111110110 x	BV5v6c7vACEJ	**0**1**0**1**0**0110**1**	±	0
D224	000001110 x	BT5q8mAJ	**1**00001110**1**	±	0
D225	100001110 x	DM4t'6u'	1000011100	+	–2
D226	010001110 x	DM4t'6u'	0100011100	+	–2
D227	110001110 x	BU4c'6c'4t'	1100011100	±	0
D228	001001110 x	DM4t'6u'	0010011100	+	–2
D229	101001110 x	BU4c'6c'4t'	1010011100	±	0
D230	011001110 x	BU4c'6c'4t'	0110011100	±	0
D231	111001110 x	DC4c'	1110011100	–	+2
D232	000101110 x	DM4t'6u'	0001011100	+	–2
D233	100101110 x	BU4c'6c'4t'	1001011100	±	0
D234	010101110 x	BU4c'6c'4t'	0101011100	±	0
D235	110101110 x	DC4c'	1101011100	–	+2
D236	001101110 x	BU4c'6c'4t'	0011011100	±	0
D237	101101110 x	DC4c'	1011011100	–	+2
D238	011101110 x	DC4c'	0111011100	–	+2
D239	111101110 x	BV4c5c8vABCJ	**000**101110**1**	±	0
D240	000011110 x	BM4t8bCEJ	00**1**0**0**1110**1**	±	0
D241	100011110 x	BU4c'6c'4t'	1000111100	±	0
D242	010011110 x	BU4c'6c'4t'	0100111100	±	0
D243	110011110 x	DC4c'	1100111100	–	+2
D244	001011110 x	BU4c'6c'4t'	0010111100	±	0
D245	101011110 x	DC4c'	1010111100	–	+2
D246	011011110 x	DC4c'	0110111100	–	+2
D247	111011110 x	FV4u8v	1110111100	–	+4
D248	000111110 x	BU4c'6c'4t'	0001111100	±	0
D249	100111110 x	DC4c'	1001111100	–	+2
D250	010111110 x	DC4c'	0101111100	–	+2
D251	110111110 x	FV4u8v	1101111100	–	+4

Table 1F

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D252	001111110 x	DC4c’	0011111100	–	+2
D253	101111110 x	FV4u8v	1011111100	–	+4
D254	011111110 x	FV4u8v	0111111100	–	+4
D255	111111110 x	BH8hADEHJ	**0**11**00**11**0**0**1**	±	0
D256	000000001 x	BS8sCDHJ	00**11**000**1**1**1**	±	0
D257	100000001 x	BQ4m8qEFJ	1000**11**001**1**	±	0
D258	010000001 x	BQ4m8qEFJ	0100**11**001**1**	±	0
D259	110000001 x	FT5u’5q’	1100000010	+	–4
D260	001000001 x	BQ4m8qEFJ	0010**11**001**1**	±	0
D261	101000001 x	FT5u’5q’	1010000010	+	–4
D262	011000001 x	FT5u’5q’	0110000010	+	–4
D263	111000001 x	DM4t’6u’	1110000010	+	–2
D264	000100001 x	BQ4m8qEFJ	0001**11**001**1**	±	0
D265	100100001 x	FT5u’5q’	1001000010	+	–4
D266	010100001 x	FT5u’5q’	0101000010	+	–4
D267	110100001 x	DM4t’6u’	1101000010	+	–2
D268	001100001 x	FT5u’5q’	0011000010	+	–4
D269	101100001 x	DM4t’6u’	1011000010	+	–2
D270	011100001 x	DM4t’6u’	0111000010	+	–2
D271	111100001 x	PU4c5c	1111000010	–	0
D272	000010001 x	BQ4t5t8qACJ	**1**0**1**010001**1**	±	0
D273	100010001 x	FT5u’5q’	1000100010	+	–4
D274	010010001 x	FT5u’5q’	0100100010	+	–4
D275	110010001 x	DM4t’6u’	1100100010	+	–2
D276	001010001 x	FT5u’5q’	0010100010	+	–4
D277	101010001 x	DM4t’6u’	1010100010	+	–2
D278	011010001 x	DM4t’6u’	0110100010	+	–2
D279	111010001 x	BU4c’6c’4t’	1110100010	±	0
D280	000110001 x	FT5u’5q’	0001100010	+	–4
D281	100110001 x	DM4t’6u’	1001100010	+	–2
D282	010110001 x	DM4t’6u’	0101100010	+	–2
D283	110110001 x	BU4c’6c’4t’	1101100010	±	0
D284	001110001 x	DM4t’6u’	0011100010	+	–2
D285	101110001 x	BU4c’6c’4t’	1011100010	±	0
D286	011110001 x	BU4c’6c’4t’	0111100010	±	0
D287	111110001 x	BC5v6cCDJ	11**00**10001**1**	±	0
D288	000001001 x	BQ5q7q8qACJ	**1**0**1**001001**1**	±	0
D289	100001001 x	FT5u’5q’	1000010010	+	–4
D290	010001001 x	FT5u’5q’	0100010010	+	–4
D291	110001001 x	DM4t’6u’	1100010010	+	–2
D292	001001001 x	FT5u’5q’	0010010010	+	–4
D293	101001001 x	DM4t’6u’	1010010010	+	–2

Table 1G

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D294	011001001 x	DM4t'6u'	0110010010	+	–2
D295	111001001 x	BU4c'6c'4t'	1110010010	±	0
D296	000101001 x	FT5u'5q'	0001010010	+	–4
D297	100101001 x	DM4t'6u'	1001010010	+	–2
D298	010101001 x	DM4t'6u'	0101010010	+	–2
D299	110101001 x	BU4c'6c'4t'	1101010010	±	0
D300	001101001 x	DM4t'6u'	0011010010	+	–2
D301	101101001 x	BU4c'6c'4t'	1011010010	±	0
D302	011101001 x	BU4c'6c'4t'	0111010010	±	0
D303	111101001 x	BC4c5c7u'CDJ	11**00**10100**1**	±	0
D304	000011001 x	FT5u'5q'	0000110010	+	–4
D305	100011001 x	DM4t'6u'	1000110010	+	–2
D306	010011001 x	DM4t'6u'	0100110010	+	–2
D307	110011001 x	BU4c'6c'4t'	1100110010	±	0
D308	001011001 x	DM4t'6u'	0010110010	+	–2
D309	101011001 x	BU4c'6c'4t'	1010110010	±	0
D310	011011001 x	BU4c'6c'4t'	0110110010	±	0
D311	111011001 x	DC4c'	1110110010	–	+2
D312	000111001 x	DM4t'6u'	0001110010	+	–2
D313	100111001 x	BU4c'6c'4t'	1001110010	±	0
D314	010111001 x	BU4c'6c'4t'	0101110010	±	0
D315	110111001 x	DC4c'	1101110010	–	+2
D316	001111001 x	BU4c'6c'4t'	0011110010	±	0
D317	101111001 x	DC4c'	1011110010	–	+2
D318	011111001 x	DC4c'	0111110010	–	+2
D319	111111001 x	BV6v8cACEJ	**0**1**0**1**0**1001**1**	±	0
D320	000000101 x	BQ5q7q8qACJ	**1**0**1**000101**1**	±	0
D321	100000101 x	FT5u'5q'	1000001010	+	–4
D322	010000101 x	FT5u'5q'	0100001010	+	–4
D323	110000101 x	DM4t'6u'	1100001010	+	–2
D324	001000101 x	FT5u'5q'	0010001010	+	–4
D325	101000101 x	DM4t'6u'	1010001010	+	–2
D326	011000101 x	DM4t'6u'	0110001010	+	–2
D327	111000101 x	BU4c'6c'4t'	1110001010	±	0
D328	000100101 x	FT5u'5q'	0001001010	+	–4
D329	100100101 x	DM4t'6u'	1001001010	+	–2
D330	010100101 x	DM4t'6u'	0101001010	+	–2
D331	110100101 x	BU4c'6c'4t'	1101001010	±	0
D332	001100101 x	DM4t'6u'	0011001010	+	–2
D333	101100101 x	BU4c'6c'4t'	1011001010	±	0
D334	011100101 x	BU4c'6c'4t'	0111001010	±	0
D335	111100101 x	BC4c6u8uCDJ	11**00**10010**1**	±	0

Table 1H

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D336	000010101 x	FT5u'5q'	0000101010	+	–4
D337	100010101 x	DM4t'6u'	1000101010	+	–2
D338	010010101 x	DM4t'6u'	0100101010	+	–2
D339	110010101 x	BU4c'6c'4t'	1100101010	±	0
D340	001010101 x	DM4t'6u'	0010101010	+	–2
D341	101010101 0	BUK'4c'6c'4t'ADEJ	0011001011	±	0
D342	011010101 x	BU4c'6c'4t'	0110101010	±	0
D343	111010101 x	DC4c'	1110101010	–	+2
D344	000110101 x	DM4t'6u'	0001101010	+	–2
D345	100110101 x	BU4c'6c'4t'	1001101010	±	0
D346	010110101 x	BU4c'6c'4t'	0101101010	±	0
D347	110110101 x	DC4c'	1101101010	–	+2
D348	001110101 x	BU4c'6c'4t'	0011101010	±	0
D349	101110101 x	DC4c'	1011101010	–	+2
D350	011110101 x	DC4c'	0111101010	–	+2
D351	111110101 x	BV5v6c7vACEJ	0101001011	±	0
D352	000001101 x	BT5q7t8tAJ	1000011011	±	0
D353	100001101 x	DM4t'6u'	1000011010	+	–2
D354	010001101 x	DM4t'6u'	0100011010	+	–2
D355	110001101 x	BU4c'6c'4t'	1100011010	±	0
D356	001001101 x	DM4t'6u'	0010011010	+	–2
D357	101001101 x	BU4c'6c'4t'	1010011010	±	0
D358	011001101 x	BU4c'6c'4t'	0110011010	±	0
D359	111001101 x	DC4c'	1110011010	–	+2
D360	000101101 x	DM4t'6u'	0001011010	+	–2
D361	100101101 x	BU4c'6c'4t'	1001011010	±	0
D362	010101101 x	BU4c'6c'4t'	0101011010	±	0
D363	110101101 x	DC4c'	1101011010	–	+2
D364	001101101 x	BU4c'6c'4t'	0011011010	±	0
D365	101101101 x	DC4c'	1011011010	–	+2
D366	011101101 x	DC4c'	0111011010	–	+2
D367	111101101 x	FV5v'8v'	1111011010	–	+4
D368	000011101 x	BM4t5t8mCEJ	0010011011	±	0
D369	100011101 x	BU4c'6c'4t'	1000111010	±	0
D370	010011101 x	BU4c'6c'4t'	0100111010	±	0
D371	110011101 x	DC4c'	1100111010	–	+2
D372	001011101 x	BU4c'6c'4t'	0010111010	±	0
D373	101011101 x	DC4c'	1010111010	–	+2
D374	011011101 x	DC4c'	0110111010	–	+2
D375	111011101 x	FV5v'8v'	1110111010	–	+4
D376	000111101 x	BU4c'6c'4t'	0001111010	±	0
D377	100111101 x	DC4c'	1001111010	–	+2

Table 1I

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D378	010111101 x	DC4c’	0101111010	–	+2
D379	110111101 x	FV5v’8v’	1101111010	–	+4
D380	001111101 x	DC4c’	0011111010	–	+2
D381	101111101 x	FV5v’8v’	1011111010	–	+4
D382	011111101 x	FV5v’8v’	0111111010	–	+4
D383	111111101 x	BH7h8vADEFJ	**0**11**000**101**1**	±	0
D384	000000011 x	BQ7sACJ	**1**0**1**000011**1**	±	0
D385	100000011 x	FT5u’5q’	1000000110	+	–4
D386	010000011 x	FT5u’5q’	0100000110	+	–4
D387	110000011 x	DM4t’6u’	1100000110	+	–2
D388	001000011 x	FT5u’5q’	0010000110	+	–4
D389	101000011 x	DM4t’6u’	1010000110	+	–2
D390	011000011 x	DM4t’6u’	0110000110	+	–2
D391	111000011 x	BU4c’6c’4t’	1110000110	±	0
D392	000100011 x	FT5u’5q’	0001000110	+	–4
D393	100100011 x	DM4t’6u’	1001000110	+	–2
D394	010100011 x	DM4t’6u’	0101000110	+	–2
D395	110100011 x	BU4c’6c’4t’	1101000110	±	0
D396	001100011 x	DM4t’6u’	0011000110	+	–2
D397	101100011 x	BU4c’6c’4t’	1011000110	±	0
D398	011100011 x	BU4c’6c’4t’	0111000110	±	0
D399	111100011 x	BC4c6u8uCDJ	11**00**00011**1**	±	0
D400	000010011 x	FT5u’5q’	0000100110	+	–4
D401	100010011 x	DM4t’6u’	1000100110	+	–2
D402	010010011 x	DM4t’6u’	0100100110	+	–2
D403	110010011 x	BU4c’6c’4t’	1100100110	±	0
D404	001010011 x	DM4t’6u’	0010100110	+	–2
D405	101010011 x	BU4c’6c’4t’	1010100110	±	0
D406	011010011 x	BU4c’6c’4t’	0110100110	±	0
D407	111010011 x	DC4c’	1110100110	–	+2
D408	000110011 x	DM4t’6u’	0001100110	+	–2
D409	100110011 x	BU4c’6c’4t’	1001100110	±	0
D410	010110011 x	BU4c’6c’4t’	0101100110	±	0
D411	110110011 x	DC4c’	1101100110	–	+2
D412	001110011 x	BU4c’6c’4t’	0011100110	±	0
D413	101110011 x	DC4c’	1011100110	–	+2
D414	011110011 x	DC4c’	0111100110	–	+2
D415	111110011 x	BV5v7cACEJ	**0**1**0**1**0**0011**1**	±	0
D416	000001011 x	BT5q6t7qBGIJ	0**1**0001**1**1**01**	±	0
D417	100001011 x	DM4t’6u’	1000010110	+	–2
D418	010001011 x	DM4t’6u’	0100010110	+	–2
D419	110001011 x	BU4c’6c’4t’	1100010110	±	0

Table 1J

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D420	001001011 x	DM4t’6u’	0010010110	+	–2
D421	101001011 x	BU4c’6c’4t’	1010010110	±	0
D422	011001011 x	BU4c’6c’4t’	0110010110	±	0
D423	111001011 x	DC4c’	1110010110	–	+2
D424	000101011 x	DM4t’6u’	0001010110	+	–2
D425	100101011 x	BU4c’6c’4t’	1001010110	±	0
D426	010101011 x	BU4c’6c’4t’	0101010110	±	0
D427	110101011 x	DC4c’	1101010110	–	+2
D428	001101011 x	BU4c’6c’4t’	0011010110	±	0
D429	101101011 x	DC4c’	1011010110	–	+2
D430	011101011 x	DC4c’	0111010110	–	+2
D431	111101011 x	FV5v’8v’	1111010110	–	+4
D432	000011011 x	BM4t5t8mBCFHJ	0**11**01**0**0**0**1**1**	±	0
D433	100011011 x	BU4c’6c’4t’	1000110110	±	0
D434	010011011 x	BU4c’6c’4t’	0100110110	±	0
D435	110011011 x	DC4c’	1100110110	–	+2
D436	001011011 x	BU4c’6c’4t’	0010110110	±	0
D437	101011011 x	DC4c’	1010110110	–	+2
D438	011011011 x	DC4c’	0110110110	–	+2
D439	111011011 x	FV5v’8v’	1110110110	–	+4
D440	000111011 x	BU4c’6c’4t’	0001110110	±	0
D441	100111011 x	DC4c’	1001110110	–	+2
D442	010111011 x	DC4c’	0101110110	–	+2
D443	110111011 x	FV5v’8v’	1101110110	–	+4
D444	001111011 x	DC4c’	0011110110	–	+2
D445	101111011 x	FV5v’8v’	1011110110	–	+4
D446	011111011 x	FV5v’8v’	0111110110	–	+4
D447	111111011 x	BH6v7vADEHJ	**0**11**00**10**0**1**1**	±	0
D448	000000111 x	BT6qBCIJ	0**11**00011**01**	±	0
D449	100000111 x	DM4t’6u’	1000001110	+	–2
D450	010000111 x	DM4t’6u’	0100001110	+	–2
D451	110000111 x	BU4c’6c’4t’	1100001110	±	0
D452	001000111 x	DM4t’6u’	0010001110	+	–2
D453	101000111 x	BU4c’6c’4t’	1010001110	±	0
D454	011000111 x	BU4c’6c’4t’	0110001110	±	0
D455	111000111 x	DC4c’	1110001110	–	+2
D456	000100111 x	DM4t’6u’	0001001110	+	–2
D457	100100111 x	BU4c’6c’4t’	1001001110	±	0
D458	010100111 x	BU4c’6c’4t’	0101001110	±	0
D459	110100111 x	DC4c’	1101001110	–	+2
D460	001100111 x	BU4c’6c’4t’	0011001110	±	0
D461	101100111 x	DC4c’	1011001110	–	+2

Table 1K

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D462	011100111 x	DC4c’	0111001110	–	+2
D463	111100111 x	FV5v’8v’	1111001110	–	+4
D464	000010111 x	BM4t5t8mBIJ	0**1**001011**01**	±	0
D465	100010111 x	BU4c’6c’4t’	1000101110	±	0
D466	010010111 x	BU4c’6c’4t’	0100101110	±	0
D467	110010111 x	DC4c’	1100101110	–	+2
D468	001010111 x	BU4c’6c’4t’	0010101110	±	0
D469	101010111 x	DC4c’	1010101110	–	+2
D470	011010111 x	DC4c’	0110101110	–	+2
D471	111010111 x	FV5v’8v’	1110101110	–	+4
D472	000110111 x	BU4c’6c’4t’	0001101110	±	0
D473	100110111 x	DC4c’	1001101110	–	+2
D474	010110111 x	DC4c’	0101101110	–	+2
D475	110110111 x	FV5v’8v’	1101101110	–	+4
D476	001110111 x	DC4c’	0011101110	–	+2
D477	101110111 x	FV5v’8v’	1011101110	–	+4
D478	011110111 x	FV5v’8v’	0111101110	–	+4
D479	111110111 x	BH5v6cADEGJ	**0**11**00**0**0**11**1**	±	0
D480	000001111 x	BM5qBHJ	0**1**00011**0**1**1**	±	0
D481	100001111 x	BU4c’6c’4t’	1000011110	±	0
D482	010001111 x	BU4c’6c’4t’	0100011110	±	0
D483	110001111 x	DC4c’	1100011110	–	+2
D484	001001111 x	BU4c’6c’4t’	0010011110	±	0
D485	101001111 x	DC4c’	1010011110	–	+2
D486	011001111 x	DC4c’	0110011110	–	+2
D487	111001111 x	FV5v’8v’	1110011110	–	+4
D488	000101111 x	BU4c’6c’4t’	0001011110	±	0
D489	100101111 x	DC4c’	1001011110	–	+2
D490	010101111 x	DC4c’	0101011110	–	+2
D491	110101111 x	FV5v’8v’	1101011110	–	+4
D492	001101111 x	DC4c’	0011011110	–	+2
D493	101101111 x	FV5v’8v’	1011011110	–	+4
D494	011101111 x	FV5v’8v’	0111011110	–	+4
D495	111101111 x	BH4c5cABCHJ	**000**1011**0**1**1**	±	0
D496	000011111 x	PU4t	0000111110	+	0
D497	100011111 x	DC4c’	1000111110	–	+2
D498	010011111 x	DC4c’	0100111110	–	+2
D499	110011111 x	FV5v’8v’	1100111110	–	+4
D500	001011111 x	DC4c’	0010111110	–	+2
D501	101011111 x	FV5v’8v’	1010111110	–	+4
D502	011011111 x	FV5v’8v’	0110111110	–	+4
D503	111011111 x	BH3c4uEFGIJ	1110**000**1**01**	±	0

Table 1L

9B10B Encoding

Name	ABCDEFGHI K	Coding Class	Primary abcdefghij	Pri DR	Pri DB
D504	000111111 x	DC4c’	0001111110	–	+2
D505	100111111 x	FV5v’8v’	1001111110	–	+4
D506	010111111 x	FV5v’8v’	0101111110	–	+4
D507	110111111 x	BH1u3uEFGIJ	1101**000**1**01**	±	0
D508	001111111 0	BVK’2mEGIJ	0011**0**1**0**1**01**	±	0
D509	101111111 x	BH1u3uEFGIJ	1011**000**1**01**	±	0
D510	011111111 x	BH1mEFGIJ	0111**000**1**01**	±	0
D511	111111111 x	BXBCEGIJ	1**00**1**0**1**0**1**01**	±	0
C508	001111111 1	FVK	0011111110	–	+4
K39[o]	111001000 1	DMK	1110010000	+	–2
K43[o]	110101000 1	DMK	1101010000	+	–2
K45[o]	101101000 1	DMK	1011010000	+	–2
K46[o]	011101000 1	DMK	0111010000	+	–2
K51[o]	110011000 1	DMK	1100110000	+	–2
K53[o]	101011000 1	DMK	1010110000	+	–2
K54[o]	011011000 1	DMK	0110110000	+	–2
K57[o]	100111000 1	DMK	1001110000	+	–2
K58[o]	010111000 1	DMK	0101110000	+	–2
K60[o]	001111000 1	DMK	0011110000	+	–2
K77	101100100 1	BMKJ	1011001001	±	0
K105	100101100 1	BMKJ	1001011001	±	0
K170	010101010 1	BMKJ	0101010101	±	0
K201	100100110 1	BMKJ	1001001101	±	0
K209	100010110 1	BMKJ	1000101101	±	0
K216	000110110 1	BMKJ	0001101101	±	0
K341	101010101 1	BUK	1010101010	±	0

Table 1M

[o] Optional control vector for 16B18B code, not valid for contiguous 9B10B vectors.

2) Construction of the 9B10B Coding Table 1

This section describes auxiliary graphs and diagrams which were used for the assignment of coded 10B vectors to uncoded 9B vectors in Table 1.

a) 414 9B Vectors congruent with the first 9 Bits of the 10B encoded Vectors (FIGS.4-9)

For 414 vectors (402 data, 12 control), represented by the trellis diagrams of FIG.4 to 9, the first nine bits of the primary encoded vectors are identical to the corresponding source vectors and the bit ‘j’ is appended with the default value (0).

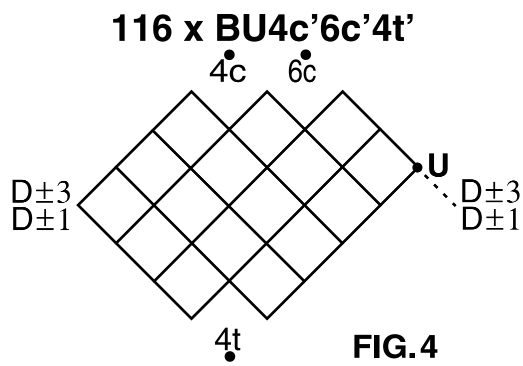

FIG. 4

FIG. 4 represents the subset of 116 balanced, disparity independent vectors of FIG. 2A.1 which end with a zero. The vector with all alternating ones and zeros is a control vector (K341).

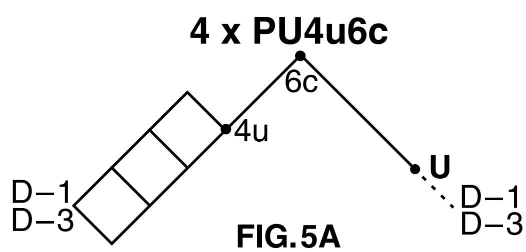

FIG. 5A

FIG. 5A, 5B, and 5C represent the 9 balanced, disparity dependent vectors of FIG. 2A.2.

FIG. 5A is a copy of the lower left side of FIG. 2A.2 and is assigned to the balanced primary data vectors D55, D59, D61, and D62 which require a negative entry disparity.

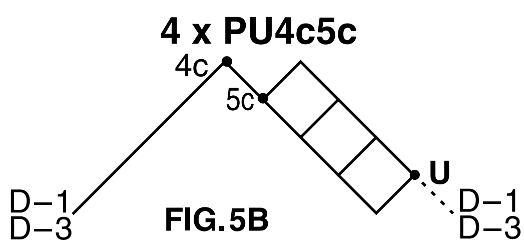

FIG. 5B

FIG. 5B represents those 4 vectors of the upper left side of FIG. 2A.2 which end with zero and are assigned to the balanced primary data vectors D47, D79, D143, and D271 which require a negative entry disparity.

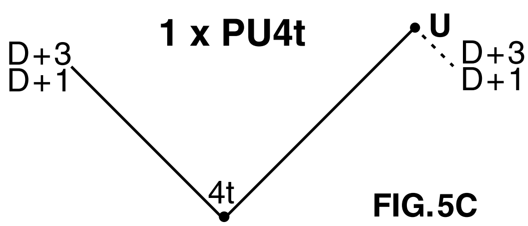

FIG. 5C

FIG. 5C is from the upper right side of FIG. 2A.2 ending with a zero and is assigned to the balanced data primary vector D496 which requires a positive entry disparity.

FIG. 6A uses all 95 vectors of FIG. 2C.1 with a disparity of four. The bold lines on the left side represent the control vector used for comma generation.

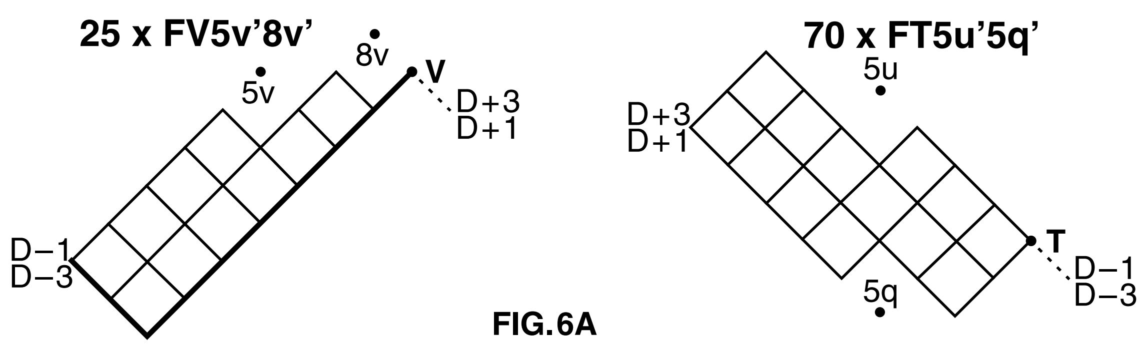

FIG. 6A

Enumeration of 25 primary Vectors FV5v'8v' of FIG. 6A(L) which require a negative entry disparity:

D367	D375	D379	D381	D382	D431	D439	D443	D445	D446
D463	D471	D475	D477	D478	D487	D491	D493	D494	D499
D501	D502	D505	D506	**C508***					

* The source vector C508 = 001111111 with K=1 is coded into **0011111110**. This represents the special character C508 and is part of the comma sequence. The same source vector D508 with K=0 represents the data vector D508 is coded into 0011010101.

Enumeration of 70 primary Vectors FT5u'5q' of FIG. 6A(R) which require a positive entry disparity:

D35	D37	D38	D41	D42	D44	D49	D50	D52	D56
D67	D69	D70	D73	D74	D76	D81	D82	D84	D88
D97	D98	D100	D104	D112	D131	D133	D134	D137	D138
D140	D145	D146	D148	D152	D161	D162	D164	D168	D176
D193	D194	D196	D200	D208	D259	D261	D262	D265	D266
D268	D273	D274	D276	D280	D289	D290	D292	D296	D304
D321	D322	D324	D328	D336	D385	D386	D388	D392	D400

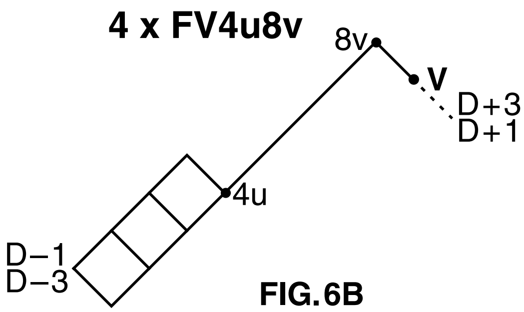

FIG. 6B

The 4 vectors of FIG. 6B with disparity of plus four correspond to the 4 vectors of FIG. 2C.2 and are assigned to the primary data vectors D247, D251, D253, and D254 and require a negative entry disparity.

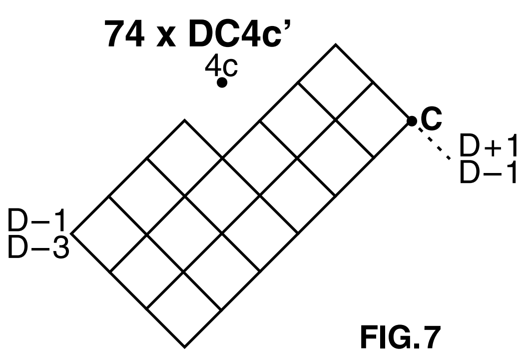

FIG. 7

The 74 vectors of FIG. 7 with a disparity of +2 are the subset of the vectors of FIG. 2B(L) which end with a zero and require a negative entry disparity.

Enumeration of 74 Vectors DC4c' of FIG. 7:

D119	D123	D125	D126	D183
D187	D189	D190	D215	D219
D221	D222	D231	D235	D237
D238	D243	D245	D246	D249

D250	D252	D311	D315	D317	D318	D343	D347	D349	D350
D359	D363	D365	D366	D371	D373	D374	D377	D378	D380
D407	D411	D413	D414	D423	D427	D429	D430	D435	D437
D438	D441	D442	D444	D455	D459	D461	D462	D467	D469
D470	D473	D474	D476	D483	D485	D486	D489	D490	D492
D497	D498	D500	D504						

106 x DM4t'6u'

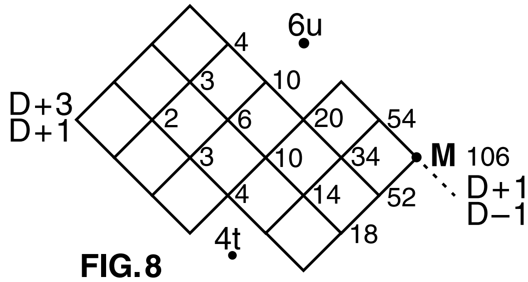

FIG. 8

The 106 primary vectors of FIG. 8 are the subset of vectors of FIG. 2B(R) with one to three trailing zeros, a disparity of –2 and require a positive entry disparity.

D71	D75	D77	D78	D83
D85	D86	D89	D90	D92
D99	D101	D102	D105	D106
D108	D113	D114	D116	D120
D135	D139	D141	D142	D147
D149	D150	D153	D154	D156
D163	D165	D166	D169	D170
D172	D177	D178	D180	D184
D195	D197	D198	D201	D202
D204	D209	D210	D212	D216
D225	D226	D228	D232	D263
D267	D269	D270	D275	D277
D278	D281	D282	D284	D291
D293	D294	D297	D298	D300
D305	D306	D308	D312	D323
D325	D326	D329	D330	D332
D337	D338	D340	D344	D353
D354	D356	D360	D387	D389
D390	D393	D394	D396	D401
D402	D404	D408	D417	D418
D420	D424	D449	D450	D452
D456				

10 x DMK5u6u

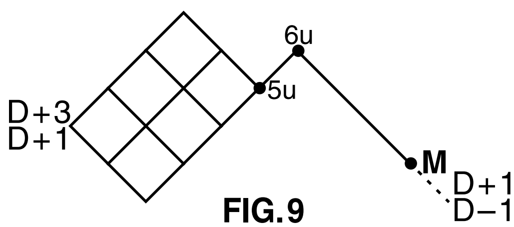

FIG. 9

FIG. 9 defines a set of 10 primary vectors with a disparity of –2 from FIG. 2B(R) with four trailing zeros as optional control vectors. They require a positive entry disparity. *These 10 control vectors can be used in the context of the 16B18B code.* If 10B vectors are directly concatenated, they would generate false commas and are invalid vectors for that application. For all other applications, their use must be specifically evaluated.

Enumeration of 10 optional Control Vectors DMK5u6u of FIG. 9:

K39	K43	K45	K46	K51	K53	K54	K57	K58	K60

The table 1M includes another set of 7 control characters. There are no restrictions on the use of those 7 control characters, and the previously defined comma character C509.

The control source vectors are chosen so there is no need to ever change any source bits for encoding except the J-bit of the 6 vectors listed in Table 2B at the bottom right side.

b) 116 Vectors with individual bit changes (FIG. 10)

FIG. 10 represents the subset of 116 balanced, disparity independent vectors of FIG. 2A.1 which end with one. The appended J-bit of FIG. 10 is marked with a fat dotted line to indicate complementation from the default value for encoding.

All source vectors which require individual bit changes for encoding are assigned to this class of balanced, disparity independent vectors. This important feature allows bit-encoding and whole vector inversions to proceed independently of each other in parallel for both encoding and decoding, greatly reducing circuit delay.

116 x BM4c'4t'6t'J

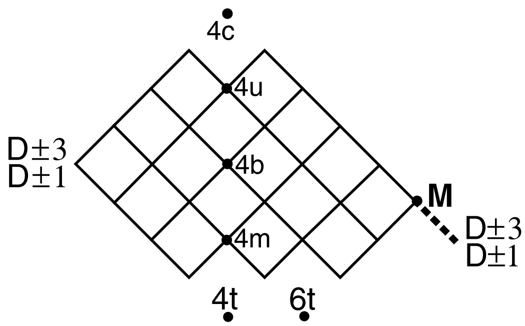

FIG. 10

The 116 vectors of FIG. 10 are listed explicitly with their assigned source vectors in Table 2. The bit values in the encoded domain which are obtained by complementation of the respective source bit or the default value of bit J are shown in bold type. A value of 1 in the column S of Table 2 indicates that the source bits on the right side are the exact complements of the left side and there are also symmetries in the coded domain which can be exploited for a simplified circuit implementation.

116 Coded Vectors of FIG. 10 with assigned Source Vectors

NAME	ABCDEFGHI K	abcdefghij	S	NAME	ABCDEFGHI K	abcdefghij
D23	111010000 x	111010000**1**	0	D17	100010000 x	1000100**111**
D27	110110000 x	110110000**1**	0	D18	010010000 x	0100100**111**
D29	101110000 x	101110000**1**	0	D20	001010000 x	0010100**111**
D30	011110000 x	011110000**1**	0	D24	000110000 x	0001100**111**
D39	111001000 x	111001000**1**	0	D33	100001000 x	1000010**111**
D43	110101000 x	110101000**1**	0	D34	010001000 x	0100010**111**
D45	101101000 x	101101000**1**	0	D36	001001000 x	0010010**111**
D46	011101000 x	011101000**1**	0	D40	000101000 x	0001010**111**
D51	110011000 x	110011000**1**	0	D65	100000100 x	1000**11**100**1**
D53	101011000 x	101011000**1**	0	D66	010000100 x	0100**11**100**1**
D54	011011000 x	011011000**1**	0	D68	001000100 x	0010**11**100**1**
D57	100111000 x	100111000**1**	0	D72	000100100 x	0001**11**100**1**
D58	010111000 x	010111000**1**	0	D129	100000010 x	1000**11**010**1**
D60	001111000 x	001111000**1**	0	D130	010000010 x	0100**11**010**1**
D19	110010000 x	1100100**101**	0	D132	001000010 x	0010**11**010**1**
D21	101010000 x	1010100**101**	0	D136	000100010 x	0001**11**010**1**
D22	011010000 x	0110100**101**	0	D257	100000001 x	1000**11**001**1**
D25	100110000 x	1001100**101**	0	D258	010000001 x	0100**11**001**1**
D26	010110000 x	0101100**101**	0	D260	001000001 x	0010**11**001**1**
D28	001110000 x	0011100**101**	0	D264	000100001 x	0001**11**001**1**
D224	000001110 x	**1**00001110**1**	0	D240	000011110 x	00**1**0**0**1110**1**
D352	000001101 x	**1**00001101**1**	0	D368	000011101 x	00**1**0**0**1101**1**
D96	000001100 x	**1**0**1**001100**1**	1	D415	111110011 x	**0**1**0**1**0**0011**1**
D384	000000011 x	**1**0**1**000011**1**	1	D127	111111100 x	**0**1**0**1**0**1100**1**
D160	000001010 x	**1**0**1**001010**1**	1	D351	111110101 x	**0**1**0**1**0**0101**1**
D192	000000110 x	**1**0**1**000110**1**	1	D319	111111001 x	**0**1**0**1**0**1001**1**
D288	000001001 x	**1**0**1**001001**1**	1	D223	111110110 x	**0**1**0**1**0**0110**1**
D320	000000101 x	**1**0**1**000101**1**	1	D191	111111010 x	**0**11**00**1010**1**

Table 2A

116 Coded Vectors of FIG.10 with assigned Source Vectors

NAME	ABCDEFGHI K	abcdefghij	S	NAME	ABCDEFGHI K	abcdefghij
D1	100000000 x	1000101011	1	D510	011111111 x	0111000101
D2	010000000 x	0100101011	1	D509	101111111 x	1011000101
D4	001000000 x	0010101011	1	D507	110111111 x	1101000101
D8	000100000 x	0001101011	1	D503	111011111 x	1110000101
D16	000010000 x	1001100011	1	D495	111101111 x	0001011011
D32	000001000 x	1001010011	1	D479	111110111 x	0110000111
D64	000000100 x	1001001011	1	D447	111111011 x	0110010011
D128	000000010 x	1001000111	1	D383	111111101 x	0110001011
D256	000000001 x	0011000111	1	D255	111111110 x	0110011001
D3	110000000 x	1100101001	0	D7	111000000 x	1110000011
D12	001100000 x	0011101001	0	D11	110100000 x	1101000011
D5	101000000 x	1010101001	0	D13	101100000 x	1011000011
D6	011000000 x	0110101001	0	D14	011100000 x	0111000011
D9	100100000 x	1001101001	0	D416	000001011 x	0100011101
D10	010100000 x	0101101001	0	D448	000000111 x	0110001101
D111	111101100 x	1100011001	0	D480	000001111 x	0100011011
D399	111100011 x	1100000111	0	D48	000011000 x	0101100011
D175	111101010 x	1100010101	0	D80	000010100 x	0111001001
D207	111100110 x	1100001101	0	D432	000011011 x	0110100011
D303	111101001 x	1100010011	0	D464	000010111 x	0100101101
D335	111100101 x	1100001011	0	D0	000000000 x	1110001001
D287	111110001 x	1100100011	0	D508	001111111 0	0011010101
D95	111110100 x	1101001001	0	D511	111111111 x	1001010101
D15	111100000 x	0011011001	0	D341	101010101 0	0011001011
D239	111101110 x	0001011101	0	K77	101100100 1	1011001001
D31	111110000 x	0011100011	0	K105	100101100 1	1001011001
D63	111111000 x	0011010011	0	K170	010101010 1	0101010101
D159	111110010 x	0011001101	0	K201	100100110 1	1001001101
D144	000010010 x	0010101101	0	K209	100010110 1	1000101101
D272	000010001 x	1010100011	0	K216	000110110 1	0001101101

Table 2B

c) Value of control bit K

For a majority of data vectors, the value of the K-bit can be ignored as indicated by x in the K column. It must be included for all classifications and logic equations which include vectors with common values ABCDEFGHI for a data and a control vector.

III. LOGIC EQUATIONS FOR IMPLEMENTATION (FIGS. 11, 12)

FIGS. 11A and 12A show a conceptual view of encoding and decoding, respectively. They illustrate the parallelism in the processing of various vector classes which is the key to the a simple implementation with low latency. Note that full vector complementation and changes in individual bits are completely separate and independent of each other.

Conceptual View of 9B10B Encoding

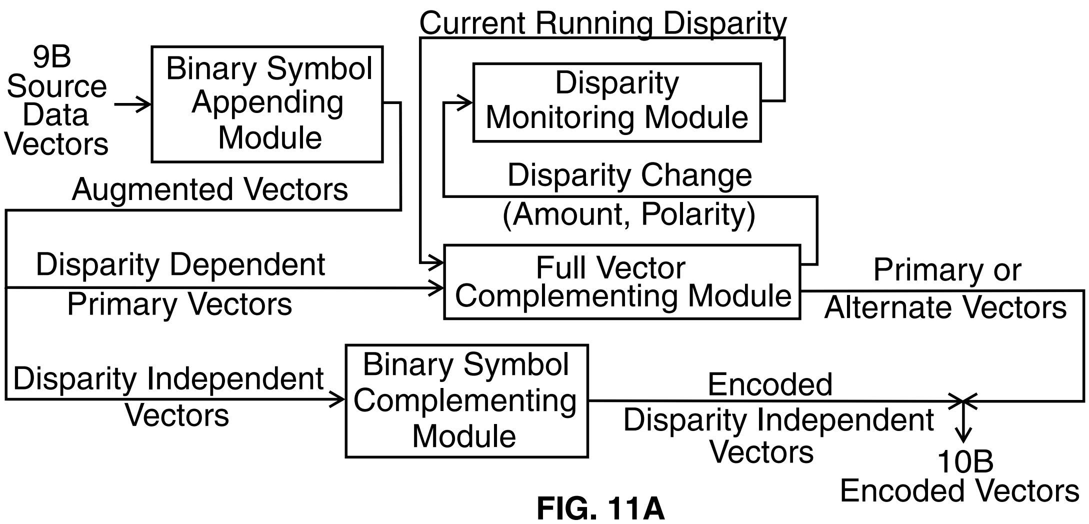

FIG. 11A

Circuit View of 9B10B Encoder

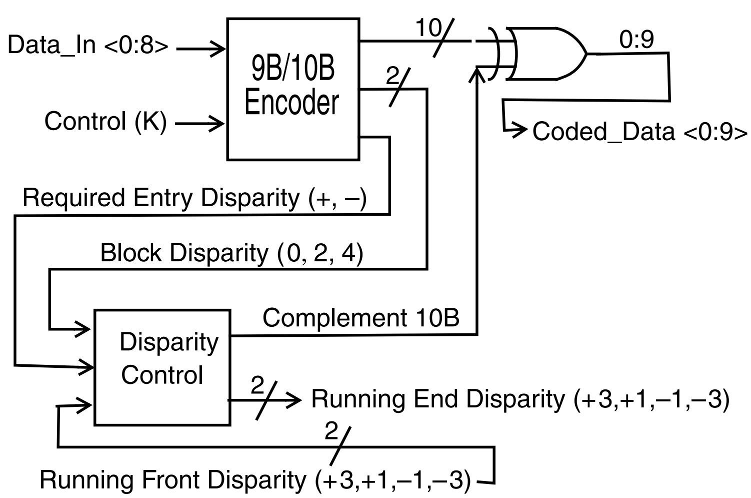

Figure 11B

FIGS. 11B and 12B present another view of encoding and decoding which is more circuit oriented.

Conceptual View of 10B9B Decoding

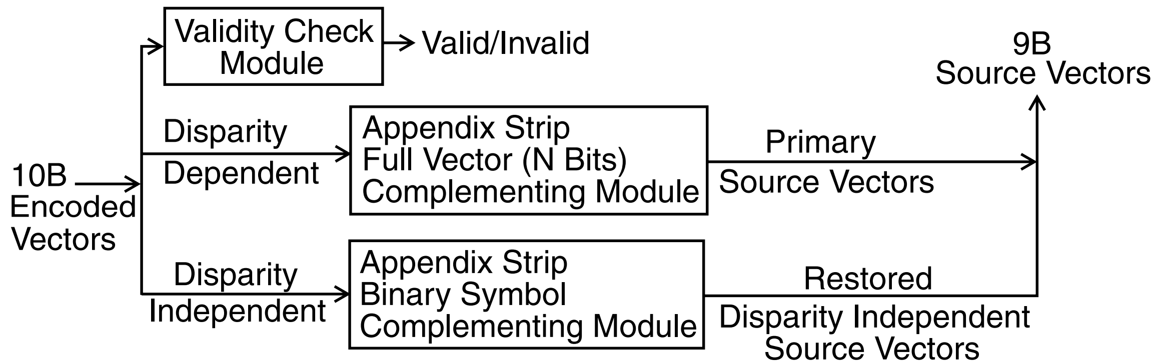

FIG. 12A

Circuit View of 10B9B Decoder

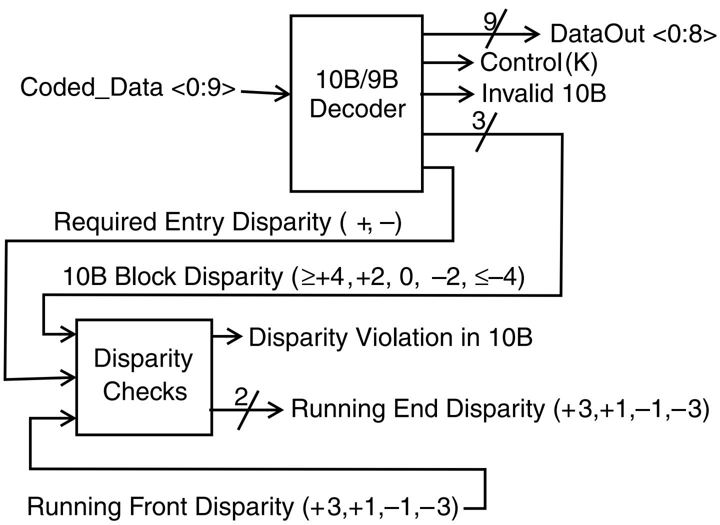

Figure 12B

The decoder circuit includes a check for invalid vectors. In the presence of errors, the received blocks may have a disparity of ±6, ±8, or ±10, which are outside the normal range but are assigned a disparity value of ±4 for purposes of the running disparity. The disparity monitoring circuit shown in FIG. 12B has not been included in this design because it may not contribute enough to the overall error checking schemes to justify the added complexity.

The implementation problems to be solved for Encoder and Decoder are circuit area and delay reduction.

Design principles illustrated for the simpler case of the partitioned 8B10B_P code with local parity of Ref. 8 are applicable here as well:

1. All vectors with individual bit changes are relegated to a class of vectors which is balanced and disparity independent.
2. Assignment of uncoded source vectors to coded vectors such that the number of vectors with individual bit changes is minimized.
3. Extensive sorting of vectors into groups with commonalities.

Notation: In the equations below, the EXCLUSIVE OR (⊕) function is executed first, followed by the AND (•), and then the OR (+) function. The EXOR function is defined with a single parameter on each side, i.e. $x \oplus y$ is equivalent to $(x \oplus y)$.

In the coding equations and tables, some vectors are included redundantly for simplification. Redundant vector names are preceded by an asterisk. In the encoding and decoding tables, the bit patterns common to several vectors usually are marked by bold type. Some of the table rows show a mixture of complementary bit sets and identical bits in the left and the right column; usually, the complementary bits are then illustrated in italic and equal bits in bold face type. The Coding labels are used to write the coding equations. In any of the Exclusive OR relationships between two groups of contiguous bits, any bit in the first and second group can be selected as the first and second input, respectively, of the XOR2 gate. The inputs have been selected to maximize commonality among the several encoding equations. The expressions in parentheses at the right edge of the equations refer to the corresponding net names in the circuit diagram. An asterisk * following the net name means that the correlation is not exact because of missing or additional terms listed on the same line. In the logic labels and equations, the components are usually listed in descending order of the estimated circuit delay.

A. Logic Equations for 9B10B Encoder

1) Equations for Individual Bit Encoding

Encoded Bit a

The ‘a’ column has bold entries in the Tables 1 and 2 for the 31 vectors listed in Table 3a. The a-bit encoding equation is derived from the coding labels of Table 3a.

$$
\begin{aligned}
a = A \oplus \{ & (E \oplus F' \bullet F \oplus G' \bullet G \oplus H + E \oplus I' \bullet F \oplus G \bullet H \oplus I' + F \oplus H \bullet G \oplus I + F \oplus G \bullet H \oplus I) \bullet && \text{(n0)} \\
& A \oplus B' \bullet B \oplus C' \bullet C \oplus D' \bullet D \oplus E' + && \text{(Pn1*)} \\
& A \oplus B' \bullet B \oplus C' \bullet C \oplus D' \bullet D \oplus E \bullet E \oplus F \bullet F \oplus G' \bullet G \oplus H' \bullet H \oplus I' + && \text{(n2)} \\
& A \oplus B' \bullet B \oplus C' \bullet C \oplus D' \bullet D \oplus H' \bullet H \oplus I \bullet E \oplus F \bullet F \oplus G' + && \text{(n3)} \\
& A \oplus B' \bullet B \oplus C' \bullet C \oplus D' \bullet D \oplus E' \bullet F \oplus G' \bullet G \oplus H' \bullet I' + && \text{(n4)} \\
& E \oplus F' \bullet A \bullet B \bullet C \bullet D \bullet G' \bullet H' \bullet I' + A \bullet B' \bullet C \bullet D' \bullet E \bullet F' \bullet G \bullet H' \bullet I \bullet K' \} && \text{(Pn5*)}
\end{aligned}
$$

a-bit Encoding

Name	ABCDEFGHI K	a	S	Name	ABCDEFGHI K	a	Coding Label
D128	000000010 x	1	1	D383	111111101 x	0	
*D384	000000011 x	1	1	*D127	111111100 x	0	
D32	000001000 x	1	1	D479	111110111 x	0	
D64	000000100 x	1	1	D447	111111011 x	0	(E⊕F'•F⊕G'•G⊕H +
D96	000001100 x	1	1	D415	111110011 x	0	E⊕I'•F⊕G•H⊕I' +
D384	000000011 x	1	1	D127	111111100 x	0	F⊕H•G⊕I + F⊕G•H⊕I) •
D160	000001010 x	1	1	D351	111110101 x	0	A⊕B'•B⊕C'•C⊕D'•D⊕E'
D192	000000110 x	1	1	D319	111111001 x	0	
D288	000001001 x	1	1	D223	111110110 x	0	
D320	000000101 x	1	1	D191	111111010 x	0	
D16	000010000 x	1	1	D495	111101111 x	0	A⊕B'•B⊕C'•C⊕D'•D⊕E • E⊕F•F⊕G'•G⊕H'•H⊕I'
D0	000000000 x	1	1	D255	111111110 x	0	A⊕B'•B⊕C'•C⊕D'•D⊕E' •
D224	000001110 x	1	1	D31	111110000 x	0	F⊕G'•G⊕H'•I'
D352	000001101 x	1	1	D159	111110010 x	0	A⊕B'•B⊕C'•C⊕D'•D⊕H'•
D272	000010001 x	1	1	D239	111101110 x	0	H⊕I•E⊕F•F⊕G'
D15	111100000 x	0	1	D63	111111000 x	0	E⊕F'•A•B•C•D•G'•H'•I'
				D341	101010101 0	0	A•B'•C•D'•E•F'•G•H'•I•K'

Table 3a

Encoded Bit b

The 'b' column has bold entries in the Tables 1 and 2 for the 15 vectors listed in Table 3b.

b-bit Encoding

Name	ABCDEFGHI K	b	S	Name	ABCDEFGHI K	b	Coding Label
D0	000000000 x	1	1	D511	111111111 x	0	
D480	000001111 x	1	1	D31	111110000 x	0	A⊕B'•B⊕C'•C⊕D'•D⊕E' •
*D0	000000000 x	1	1	*D511	111111111 x	0	G⊕H'•H⊕I'•(E⊕F'+F⊕G')
D448	000000111 x	1	1	D63	111111000 x	0	
D48	000011000 x	1	1	D239	111101110 x	0	(E⊕G•G⊕H'•F•I'+E⊕G'•F'•H'•
D80	000010100 x	1	1	D15	111100000 x	0	I')•A⊕B'•B⊕C'•C⊕D'•D⊕E
D416	000001011 x	1	1	D159	111110010 x	0	A⊕B'•B⊕C'•C⊕D'•D⊕E' • B⊕I•E⊕F•G'•H
D432	000011011 x	1	1	D464	000010111 x	1	F⊕G•A'•B'•C'•D'•E•H•I
				D495	111101111 x	0	A•B•C•D•E'•F•G•H•I

Table 3b

$$b = B\oplus\{A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus E' \bullet G\oplus H'\bullet H\oplus I' \bullet (E\oplus F'+F\oplus G') + \qquad (n6)$$
$$(E\oplus G\bullet G\oplus H'\bullet F\bullet I'+E\oplus G'\bullet F'\bullet H'\bullet I') \bullet A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus E + \qquad (Pn8^*;Pn9^*)$$
$$A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus E' \bullet B\oplus I\bullet E\oplus F\bullet G'\bullet H + \qquad (Pn9^*)$$
$$F\oplus G\bullet A'\bullet B'\bullet C'\bullet D'\bullet E\bullet H\bullet I + A\bullet B\bullet C\bullet D\bullet E'\bullet F\bullet G\bullet H\bullet I\} \qquad (Pn9^*;Pn8^*)$$

Encoded Bit c

The ‘c’ column has bold entries in the Tables 1 and 2 for the 31 vectors listed in Table 3c.

c-bit Encoding

Name	ABCDEFGHI K	c	S	Name	ABCDEFGHI K	c	Coding Label
D96	**00000**1100 x	1	1	D415	**11111**0011 x	0	
D384	**00000**0011 x	1	1	D127	**11111**1100 x	0	(F⊕H•G⊕I + F⊕G•H⊕I) •
D160	**00000**1010 x	1	1	D351	**11111**0101 x	0	(E⊕G•G⊕I’)’ •
D192	**00000**0110 x	1	1	D319	**11111**1001 x	0	A⊕B’•B⊕C’•C⊕D’•D⊕E’
D288	**00000**1001 x	1	1	D223	**11111**0110 x	0	
D272	*00001***0***001* x	1	1	D207	*11110***0***110* x	0	
D80	*000010***100** x	1	1	D111	*111101***100** x	0	(E⊕G•G⊕H’•H⊕I•F’+
D144	*000010***010** x	1	1	D175	*111101***010** x	0	E⊕F•G⊕H•I’+E⊕F’•G’•H•I+
D432	*000011***011** x	1	1	D399	*111100***011** x	0	E⊕G’•F•H’•I+F•G•H•I’) •
D368	*00001*1**1**1**01** x	1	1	D303	*11110*10**01** x	0	A⊕B’•B⊕C’•C⊕D’•D⊕E
D240	*00001***1110** x	1	1	D239	*11110***1110** x	0	
D95	**111110***1*0*0* x	0	1	D287	**111110***00*1* x	0	G⊕I•E•F’•H’ • A•B•C•D
D0	**0000000**0*0* x	1	1	D256	**0000000**0*1* x	1	(G’•H’+ G•I) •
D320	**0000001***0*1 x	1	1	D448	**0000001***1*1 x	1	A’•B’•C’•D’ • E’•F’
				D495	**111101111** x	0	(E’•F’•H’ + F•H) •
				D511	**111111111** x	0	A•B•C•D • G•I
				D335	**111**00**101** x	0	

Table 3c

$$c=C\oplus\{(F\oplus H\bullet G\oplus I+F\oplus G\bullet H\oplus I) \bullet (E\oplus G\bullet G\oplus I')' \bullet A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus E' + \qquad (Pn12^*)$$
$$(E\oplus G\bullet G\oplus H'\bullet H\oplus I\bullet F' + E\oplus F\bullet G\oplus H\bullet I' + E\oplus F'\bullet G'\bullet H\bullet I + E\oplus G'\bullet F\bullet H'\bullet I + \qquad (n10)$$
$$F\bullet G\bullet H\bullet I') \bullet A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus E + \qquad (Pn12^*;Pn11^*)$$
$$G\oplus I\bullet E\bullet F'\bullet H' \bullet A\bullet B\bullet C\bullet D + \qquad (Pn13^*)$$
$$(E'\bullet F'\bullet H' + F\bullet H) \bullet A\bullet B\bullet C\bullet D \bullet G\bullet I + (G'\bullet H'+ G\bullet I) \bullet A'\bullet B'\bullet C'\bullet D' \bullet E'\bullet F'\} \qquad (Pn13^*)$$

Encoded Bit d

The 'd' column has bold entries in the Tables 1 and 2 for the 19 vectors listed in Table 3d.

d-bit Encoding

Name	ABCDEFGHI K	d	S	Name	ABCDEFGHI K	d	Coding Label
D64	**000000**10**0** x	1	1	D447	**111111**01**1** x	0	A⊕B'•B⊕C'•C⊕D'•D⊕E' • E⊕F' • (F⊕G'•G⊕H'•H⊕I + B⊕I'•G⊕H)
D128	**000000**01**0** x	1	1	D383	**111111**10**1** x	0	
D256	**000000**001 x	1	1	D255	**111111**110 x	0	
D80	000010100 x	1	1	D287	111110001 x	0	A⊕B'•B⊕C'•C⊕D'•D⊕G • G⊕I•E•F'•H'
D16	**0000**10**000** x	1	1	D32	**0000**01**000** x	1	E⊕F•A'•B'•C'•D'•G'•H'•I'
D341	101010101 0	1					A•B'•C•D'•E•F'•G•H'•I•K'
D191	**11111**10**1**0 x	0	1	D479	**11111**01**1**1 x	0	F⊕G•G⊕I' • A•B•C•D • E•H
				D111	**11110**1100 x	0	(F⊕H•G⊕I+F⊕G•H⊕I)• A•B•C•D • E'
				D399	**11110**0011 x	0	
				D175	**11110**1010 x	0	
				D207	**11110**0110 x	0	
				D303	**11110**1001 x	0	
				D335	**11110**0101 x	0	

Table 3d

$$d = D\oplus\{(F\oplus H\bullet G\oplus I + F\oplus G\bullet H\oplus I)\bullet A\bullet B\bullet C\bullet D\bullet E' + \quad (Pn17^*)$$
$$(F\oplus G'\bullet G\oplus H'\bullet H\oplus I + B\oplus I'\bullet G\oplus H)\bullet A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus E'\bullet E\oplus F' + \quad (n14;n15)$$
$$A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus G\bullet G\oplus I\bullet E\bullet F'\bullet H' + A\bullet B'\bullet C\bullet D'\bullet E\bullet F'\bullet G\bullet H'\bullet I\bullet K' + \quad (Pn17^*)$$
$$F\oplus G\bullet G\oplus I'\bullet A\bullet B\bullet C\bullet D\bullet E\bullet H + E\oplus F\bullet A'\bullet B'\bullet C'\bullet D'\bullet G'\bullet H'\bullet I'\} \quad (Pn16^*)$$

Encoded Bit e

The 'e' column has bold entries in the Tables 1 and 2 for the 45 vectors listed in Table 3e.

$$e = E\oplus\{(A\oplus B'\bullet B\oplus C\bullet C\oplus D'\bullet D\oplus E' + A\oplus B'\bullet B\oplus I'\bullet C\oplus D + A\oplus B\bullet C\oplus D'\bullet D\oplus E')\bullet \quad (n18)$$
$$E\oplus F'\bullet F\oplus G'\bullet G\oplus H'\bullet H\oplus I'\bullet K' + \quad (Pn19)$$
$$(A\oplus B\bullet G\oplus H\bullet I' + A\oplus B\bullet G'\bullet H'\bullet I)\bullet C'\bullet D'\bullet E'\bullet F' + \quad (n22;n23)$$
$$(A\oplus B\bullet C\oplus D + A'\bullet B'\bullet C\bullet D)\bullet E'\bullet F'\bullet G'\bullet H'\bullet I' + \quad (Pn24^*)$$
$$(H\oplus I\bullet F + F'\bullet H'\bullet I')\bullet A'\bullet B'\bullet C'\bullet D'\bullet E\bullet G + \quad (n20;Pn21)$$
$$(G\oplus H\bullet I' + G'\bullet H'\bullet I)\bullet C\oplus D\bullet A'\bullet B'\bullet E'\bullet F' + (F+G+H)\bullet A\bullet B\bullet C\bullet D\bullet E + \quad (Pn25^*)$$
$$A\bullet B'\bullet C\bullet D'\bullet E\bullet F'\bullet G\bullet H'\bullet I\bullet K'\} \quad (Pn25^*)$$

e-bit Encoding

Name	ABCDEFGHI K	e	S	Name	ABCDEFGHI K	e	Coding Label
				D63	111111000 x	0	
				D95	111110100 x	0	
				D127	111111100 x	0	
				D159	111110010 x	0	
				D191	111111010 x	0	
				D223	111110110 x	0	
				D255	111111110 x	0	(F+G+H) • A•B•C•D • E
				D319	111111001 x	0	
				D351	111110101 x	0	
				D383	111111101 x	0	
				D415	111110011 x	0	
				D447	111111011 x	0	
				D479	111110111 x	0	
				D511	111111111 x	0	
				D80	000010100 x	0	(H⊕I•F + F’•H’•I’) •
				D240	000011110 x	0	A’•B’•C’•D’ • E•G
				D368	000011101 x	0	
D1	100000000 x	1	1	D510	011111111 x	0	(A⊕B’•B⊕C•C⊕D’•D⊕E’ +
D2	010000000 x	1	1	D509	101111111 x	0	A⊕B’•B⊕I’•C⊕D +
D4	001000000 x	1	1	D507	110111111 x	0	A⊕B•C⊕D’•D⊕E’) •
D8	000100000 x	1	1	D503	111011111 x	0	E⊕F’•F⊕G’•G⊕H’•H⊕I’•K’
D3	110000000 x	1	1	D508	001111111 0	0	
D5	101000000 x	1					
D6	011000000 x	1					(A⊕B•C⊕D + A’•B’•C•D) •
D9	100100000 x	1					E’•F’•G’•H’•I’
D10	010100000 x	1					
D12	001100000 x	1					
D65	100000100 x	1					
D66	010000100 x	1					(A⊕B•G⊕H•I’+A⊕B•G’•H’•I) •
D129	100000010 x	1	0	D257	100000001 x	1	C’•D’•E’•F’
D130	010000010 x	1	0	D258	010000001 x	1	
D68	001000100 x	1					
D72	000100100 x	1					(G⊕H • I’ + G’•H’•I) •
D132	001000010 x	1	0	D260	001000001 x	1	C⊕D • A’•B’•E’•F’
D136	000100010 x	1	0	D264	000100001 x	1	
				D341	1010101010	0	A•B’•C•D’•E•F’•G•H’•I•K’

Table 3e

Encoded Bit f

The 'f' column has bold entries in the Tables 1 and 2 for the 19 vectors listed in Table 3f.

f-bit Encoding

Name	ABCDEFGHI K	f	S	Name	ABCDEFGHI K	f	Coding Label
D15	*111100**0**00* x	1	1	D432	*000011**0**11* x	0	A⊕B'•B⊕C'•C⊕D'•D⊕E • E⊕F'•B⊕I•H⊕I'•G'
D65	1000**00**100 x	1					(A⊕B•C'•D' + C⊕D•A'•B') • (G•H'•I' + G'•H•I' + G'•H'•I) • E'•F'
D66	0100**00**100 x	1					
D68	0010**00**100 x	1					
D72	0001**00**100 x	1					
D129	1000**00**010 x	1					
D130	0100**00**010 x	1					
D132	0010**00**010 x	1					
D136	0001**00**010 x	1					
D257	1000**00**001 x	1					
D258	0100**00**001 x	1					
D260	0010**00**001 x	1					
D264	0001**00**001 x	1					
				D383	111111101 x	0	A•B•C•D•E•F•G•H'•I
				D503	1110**11111** x	0	(A⊕B•C•D + C⊕D•A•B) • E•F•G•H•I
				D507	1101**11111** x	0	
				D509	1011**11111** x	0	
				D510	0111**11111** x	0	

Table 3f

f = F⊕{(A⊕B•C'•D' + C⊕D•A'•B') • (G•H'•I' + G'•H•I' + G'•H'•I) • E'•F' + *(Pn31)*
(A⊕B•C•D + C⊕D•A•B) • E • F•G•H•I + *(Pn32*)*
A⊕B'•B⊕C'•C⊕D'•D⊕E • E⊕F'•B⊕I•H⊕I'•G' + *(Pn32*)*
A•B•C•D•E•F•G•H'•I} *(Pn32*)*

Encoded Bit g

The 'g' column has bold entries in the Tables 1 and 2 for the 22 vectors listed in Table 3g. The bit value K' has been added redundantly to the net Pn33 of FIG. 15B to allow circuit sharing with net Pn50 for j-bit encoding.

g = G⊕{(A⊕B'•B⊕I'•C⊕D + C⊕D'•D⊕E'•K') • *(Pn33*)*
E⊕F'•F⊕G'•G⊕H'•H⊕I' + *(Pn33*)*
A⊕B'•B⊕C'•C⊕D'•D⊕G'•G⊕I'•E•F'•H + *(Pn34*)*
A⊕B'•B⊕C'•C⊕D'•D⊕E' • B⊕I•E⊕F•G'•H + *(Pn34*)*
(A⊕B•C⊕D + A⊕B'•C•D • <u>K'</u>) • E'•F'•G'•H'•I'} *(Pn34*;Pn33*)*

g-bit Encoding

Name	ABCDEFGHI K	g	S	Name	ABCDEFGHI K	g	Coding Label
D1	10**0000000** x	1	1	D510	01**1111111** x	0	(A⊕B'•B⊕I'•C⊕D + C⊕D'•D⊕E'•K') • E⊕F'•F⊕G'•G⊕H'•H⊕I'
D2	01**0000000** x	1	1	D509	10**1111111** x	0	
D0	00**0000000** x	1	1	D511	11**1111111** x	0	
D3	11**0000000** x	1	1	D508	00**1111111** 0	0	
D4	**001**0**00000** x	1	1	D507	**110**1**11111** x	0	
D8	**000**1**00000** x	1	1	D503	**111**0**11111** x	0	
D144	*0000*10*0*1*0* x	1	1	D479	*1111*10*1*1*1* x	0	A⊕B'•B⊕C'•C⊕D' • D⊕G'•G⊕I' •E•F'•H
D159	*11111*0**01***0* x	1		D416	*00000*1**01***1* x	1	A⊕B'•B⊕C'•C⊕D'•D⊕E' • B⊕I•E⊕F•G'•H
D5	1010**00000** x	1					(A⊕B•C⊕D + A⊕B'•C•D) • E'•F'•G'•H'•I'
D6	0110**00000** x	1					
D9	1001**00000** x	1	0	D12	00**1100000** x	1	
D10	0101**00000** x	1	0	D15	11**1100000** x	1	

Table 3g

Encoded Bit h

The 'h' column has bold entries in the Tables 1and 2 for the 20 vectors listed in Table 3h.

h-bit Encoding

Name	ABCDEFGHI K	h	S	Name	ABCDEFGHI K	h	Coding Label
D19	1100**10000** x	1					(A⊕B'•B⊕C•C⊕D' + A⊕B•C⊕D + A⊕B•C'•D' + C⊕D•A'•B') • E•F'•G'•H'•I'
D28	0011**10000** x	1					
D21	1010**10000** x	1					
D22	0110**10000** x	1					
D25	1001**10000** x	1					
D26	0101**10000** x	1					
D17	10**0010000** x	1					
D18	01**0010000** x	1					
D20	**001**0**10000** x	1					
D24	**000**1**10000** x	1					
D33	1000**01000** x	1	1	D36	0010**01000** x	1	(A⊕B•C'•D'+C⊕D•A'•B') • E'•F•G'•H'•I'
D34	0100**01000** x	1	1	D40	0001**01000** x	1	
D256	*000000001* x	1	1	D255	*111111110* x	0	(D⊕E'•E⊕F'•F⊕G'•G⊕H' • H⊕I + E⊕G • F•H•I) • A⊕B'•B⊕C'•C⊕D'
D432	*0000*11**011** x	0	1	D495	*1111*01**111** x	0	
D480	*0000*0**1111** x	0	1	D447	*1111*1**1011** x	0	

Table 3h

$$h = H\oplus\{(A\oplus B'\bullet B\oplus C\bullet C\oplus D' + A\oplus B\bullet C\oplus D + A\oplus B\bullet C'\bullet D' + C\oplus D\bullet A'\bullet B')\bullet \qquad (n35)$$
$$E\bullet F'\bullet G'\bullet H'\bullet I' + (A\oplus B\bullet C'\bullet D' + C\oplus D\bullet A'\bullet B')\bullet E'\bullet F\bullet G'\bullet H'\bullet I' + \qquad (Pn36^*;Pn37)$$
$$D\oplus E'\bullet E\oplus F'\bullet F\oplus G'\bullet G\oplus H'\bullet H\oplus I + E\oplus G\bullet F\bullet H\bullet I)\bullet A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\} \qquad (n14;Pn38)$$

Encoded Bit i

The ‘i’ column has bold entries in the Tables 1and 2 for the 32 vectors listed in Table 3i.

i-bit Encoding

Name	ABCDEFGHI K	i	S	Name	ABCDEFGHI K	i	Coding Label
D1	10**0000000** x	1	1	D510	01**1111111** x	0	(A⊕B•C⊕D’•D⊕E’ +
D2	01**0000000** x	1	1	D509	10**1111111** x	0	A⊕B’•B⊕I’•C⊕D) •
D4	**00**1**000000** x	1	1	D507	**11**0**111111** x	0	E⊕F’•F⊕G’•G⊕H’•H⊕I’
D8	**00**01**00000** x	1	1	D503	**11**10**11111** x	0	
D7	1110**00000** x	1					
D11	1101**00000** x	1					(A⊕B•C•D+C⊕D•A•B) •
D13	1011**00000** x	1					E’•F’•G’•H’•I’
D14	0111**00000** x	1					
D16	000010**000** x	1					
D17	100010**000** x	1					
D18	010010**000** x	1					
D20	001010**000** x	1					
D24	000110**000** x	1					(B’•C’•D’+A’•C’•D’+A’•B’•D’+
D32	000001**000** x	1					A’•B’•C’) • E⊕F • G’•H’•I’
D33	100001**000** x	1					
D34	010001**000** x	1					
D36	001001**000** x	1					
D40	000101**000** x	1					
D63	*111111000* x	1	1	D448	*000000111* x	0	(F⊕G•G⊕H’•H⊕I’ +
D128	*0000000*1*0* x	1	1	D511	*1111111*1*1* x	0	F⊕G’•G•⊕I’•H) • E⊕F’•
							A⊕B’•B⊕C’•C⊕D’•D⊕E’
D31	*1111***10***000* x	1	1	D464	*0000***10***111* x	0	A⊕B’•B⊕C’•C⊕D’ •
			1				B⊕I•G⊕H’•H⊕I’•E•F’
D48	**0000***11*0*00* x	1	1	D416	**0000***0*10*11* x	0	E⊕I•H⊕I’•A’•B’•C’•D’•F•G’
D64	**00***0000*1*00* x	1	1	D508	**00***1111*1*11* 0	0	C⊕D’•D⊕E’•E⊕F’•
							D⊕H’•H⊕I’ • A’•B’•G•K’

Table 3i

$$
\begin{array}{ll}
i = I\oplus\{(A\oplus B\bullet C\oplus D'\bullet D\oplus E'+A\oplus B'\bullet B\oplus I'\bullet C\oplus D) \bullet E\oplus F'\bullet F\oplus G'\bullet G\oplus H'\bullet H\oplus I') + & \text{(Pn41*)} \\
(A\oplus B\bullet C\bullet D + C\oplus D\bullet A\bullet B) \bullet E'\bullet F'\bullet G'\bullet H'\bullet I' + & \text{(Pn42*;Pn43*)} \\
(B'\bullet C'\bullet D' + A'\bullet C'\bullet D' + A'\bullet B'\bullet D' + A'\bullet B'\bullet C') \bullet E\oplus F \bullet G'\bullet H'\bullet I' + & \text{(Pn39;n40)} \\
A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet B\oplus I \bullet G\oplus H'\bullet H\oplus I' \bullet E\bullet F' + & \text{(Pn41*)} \\
C\oplus D'\bullet D\oplus E'\bullet E\oplus F' \bullet D\oplus H'\bullet H\oplus I' \bullet A'\bullet B'\bullet G + & \text{(Pn42*)} \\
(F\oplus G\bullet G\oplus H'\bullet H\oplus I' + F\oplus G'\bullet G\bullet\oplus I'\bullet H)\bullet A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus E'\bullet E\oplus F' + & \text{(Pn43*)} \\
E\oplus I\bullet H\oplus I'\bullet A'\bullet B'\bullet C'\bullet D'\bullet F\bullet G'\bullet K'\} & \text{(Pn42*)}
\end{array}
$$

Encoded Bit j

The ‘j’ column has bold entries foe all 116 vectors of Table 2 listed and rearranged in Table 3j.

As illustrated at the end of Table 1M, all 12 control characters with a value of j=0 for the primary vector have a value of I=1 or GH=00 and all 6 control characters with j=1 have I=0 and (G+H)=1. With K=1, only the 18 valid control vectors must be presented at the input to the encoder. Therefore, the set of 6 control characters listed in Table 3j can be uniquely identified by the bit pattern $(G+H)\bullet I'\bullet K$.

$$
\begin{array}{ll}
j = (A\oplus B\bullet C\oplus D'\bullet E' + A\oplus B'\bullet C\oplus D\bullet E' + A\oplus B\bullet C\oplus D\bullet E+A\oplus B'\bullet C\oplus D'\bullet E)\bullet & \text{(n44)} \\
F\bullet G'\bullet H'\bullet I'\bullet K' + (C\oplus D\bullet A\bullet B + C\bullet D\bullet K') \bullet E\bullet F\bullet G\bullet H\bullet I + & \text{(Pn45*;Pn50)} \\
(A\oplus B\bullet C'\bullet D' + C\oplus D\bullet A'\bullet B') \bullet (H\oplus I\bullet G'+G\bullet H'\bullet I') \bullet E'\bullet F'+ & \text{(Pn46)} \\
(F\oplus G\bullet H\oplus I + F\oplus G'\bullet H\oplus I' + F\bullet G\bullet H\bullet I') \bullet A\bullet B\bullet C\bullet D \bullet E' + & \text{(Pn47)} \\
(F\oplus G\bullet H\oplus I'+F\oplus G'\bullet H\oplus I)\bullet A'\bullet B'\bullet C'\bullet D'\bullet E + A\oplus B'\bullet B\oplus C'\bullet C\oplus D'\bullet D\oplus E' + & \text{(Pn49;Pn51*)} \\
A\bullet B'\bullet C\bullet D'\bullet E\bullet F'\bullet G\bullet H'\bullet I\bullet K'+ (G+H)\bullet I'\bullet K + F'\bullet G'\bullet H'\bullet I' & \text{(Pn51*)}
\end{array}
$$

j-bit Encoding

Name	ABCDEFGHI K	S	Name	ABCDEFGHI K	Coding Label
D33	100001000 x	1	D46	011101000 0	
D34	010001000 x	1	D45	101101000 0	
D36	001001000 x	1	D43	110101000 0	(A⊕B•C⊕D'•E' + A⊕B'•C⊕D•E' +
D39	111001000 0	1	D40	000101000 x	A⊕B•C⊕D•E + A⊕B'•C⊕D'•E) •
D57	100111000 0	1	D54	011011000 0	F•G'•H'•I' • K'
D58	010111000 0	1	D53	101011000 0	
D60	001111000 0	1	D51	110011000 0	
*D63	111111000 x	1	*D48	000011000 x	
D80	000010100 x	1	D432	000011011 x	
D48	000011000 x	1	D464	000010111 x	(F⊕G•H⊕I'+F⊕G'•H⊕I) •
D144	000010010 x	1	D368	000011101 x	A'•B'•C'•D' • E
D240	000011110 x	1	D272	000010001 x	
D0	000000000 x	0	D16	000010000 x	
D1	100000000 x	0	D17	100010000 x	
D2	010000000 x	0	D18	010010000 x	
D3	110000000 x	0	D19	110010000 x	
D4	001000000 x	0	D20	001010000 x	
D5	101000000 x	0	D21	101010000 x	
D6	011000000 x	0	D22	011010000 x	
D7	111000000 x	0	D23	111010000 x	F'•G'•H'•I'
D8	000100000 x	0	D24	000110000 x	
D9	100100000 x	0	D25	100110000 x	
D10	010100000 x	0	D26	010110000 x	
D11	110100000 x	0	D27	110110000 x	
D12	001100000 x	0	D28	001110000 x	
D13	101100000 x	0	D29	101110000 x	
D14	011100000 x	0	D30	011110000 x	
D15	111100000 x	0	D31	111110000 x	
D175	111101010 x	0	*D15	111100000 x	
D303	111101001 x	0	D111	111101100 x	(F⊕G•H⊕I + F⊕G'•H⊕I' +
D207	111100110 x	0	D399	111100011 x	F•G•H•I') • A•B•C•D • E'
D335	111100101 x	0	D495	111101111 x	
D239	111101110 x				
D341	101010101 0				A•B'•C•D'•E•F'•G•H'•I•K'

Table 3j.1

j-bit Encoding

Name	ABCDEFGHI K	S	Name	ABCDEFGHI K	Coding Label
D129	1000**00**010 x	0	D257	1000**00**001 x	
D130	0100**00**010 x	0	D258	0100**00**001 x	
D132	0010**00**010 x	0	D260	0010**00**001 x	
D136	0001**00**010 x	0	D264	0001**00**001 x	(A⊕B•C'•D'+C⊕D•A'•B') •
D65	1000**00**100 x				(H⊕I•G'+G•H'•I') • E'•F'
D66	0100**00**100 x				
D68	0010**00**100 x				
D72	0001**00**100 x				
*D31	*11111***0000** *x*	1	**D0*	*00000***0000** *x*	
D63	*11111***1000** *x*	1	*D32*	*00000***1000** *x*	
D95	*11111***0100** *x*	1	*D64*	*00000***0100** *x*	
D127	*11111***1100** *x*	1	*D96*	*00000***1100** *x*	
D159	*11111***0010** *x*	1	*D128*	*00000***0010** *x*	
D191	*11111***1010** *x*	1	*D160*	*00000***1010** *x*	
D223	*11111***0110** *x*	1	*D192*	*00000***0110** *x*	
D255	*11111***1110** *x*	1	*D224*	*00000***1110** *x*	A⊕B'•B⊕C'•C⊕D'•D⊕E'
D287	*11111***0001** *x*	1	*D256*	*00000***0001** *x*	
D319	*11111***1001** *x*	1	*D288*	*00000***1001** *x*	
D351	*11111***0101** *x*	1	*D320*	*00000***0101** *x*	
D383	*11111***1101** *x*	1	*D352*	*00000***1101** *x*	
D415	*11111***0011** *x*	1	*D384*	*00000***0011** *x*	
D447	*11111***1011** *x*	1	*D416*	*00000***1011** *x*	
D479	*11111***0111** *x*	1	*D448*	*00000***0111** *x*	
D511	*11111***1111** *x*	1	*D480*	*00000***1111** *x*	
D503	**11***10***11111** x	1	D507	**11***01***11111** x	
D508	00**1111111** 0				(C⊕D•A•B+C•D•K') •
D509	10**1111111** x				E•F•G•H•I
D510	01**1111111** x				
*D511	11**1111111** x				
K77	10110010**0 1**				
K105	10010110**0 1**				
K170	01010101**0 1**				(G+H)•I'•K
K201	10010011**0 1**				
K209	10001011**0 1**				
K216	00011011**0 1**				

Table 3j.2

2) Equations for the Required Disparity for Encoding DR

a) Positive Required Disparity for Encoding: PDR

A total of 187 vectors listed in the Table 1 require a positive entry disparity (PDR). They are listed and sorted in Table 4.

Positive Required Disparity PDR

Name	ABCDEFGHI K	DB	Name	ABCDEFGHI K	DB	Coding Label
K39[o]	111001**000** **1**	2	K53[o]	101011**000** **1**	2	G’•H’•K
K43[o]	110101**000** **1**	2	K54[o]	011011**000** **1**	2	
K45[o]	101101**000** **1**	2	K57[o]	100111**000** **1**	2	
K46[o]	011101**000** **1**	2	K58[o]	010111**000** **1**	2	
K51[o]	110011**000** **1**	2	K60[o]	001111**000** **1**	2	
D496	**00001**1111 x	0				{F⊕G’•H⊕I’•(F+H)+F⊕G•H⊕I} • A’•B’•C’•D’•E
D112	**00001**1100 x	4				
D400	**00001**0011 x	4				
D176	**00001**1010 x	4				
D208	**00001**0110 x	4				
D304	**00001**1001 x	4				
D336	**00001**0101 x	4				
D78	011**00100** x	2				(A⊕B•C•D+C⊕D•A•B) • (G•H’•I’+G’•H•I’+G’•H’•I) • E’•F’•K’
D77	101**00100** **0**	2				
D75	110**00100** x	2				
D71	111**00100** x	2				
D142	011**00010** x	2				
D141	101**00010** x	2				
D139	110**00010** x	2				
D135	111**00010** x	2				
D270	011**00001** x	2				
D269	101**00001** x	2				
D267	110**00001** x	2				
D263	111**00001** x	2				

Table 4A

[o] Optional control vector for 16B18B code, not valid for contiguous 9B10B vectors.

The validity of the expression G’•H’•K can be verified from the last 18 rows of Table 1M where all control characters are listed.

Positive Required Disparity PDR

Name	ABCDEFGHI K	DB	Name	ABCDEFGHI K	DB	Coding Label
D49	100**11000** x	4	D449	100**00111** x	2	
D50	010**11000** x	4	D450	010**00111** x	2	
D52	001**11000** x	4	D452	001**00111** x	2	
D56	000**11000** x	4	D456	000**00111** x	2	
D81	100**10100** x	4	D417	100**01011** x	2	
D82	010**10100** x	4	D418	010**01011** x	2	
D84	001**10100** x	4	D420	001**01011** x	2	
D88	000**10100** x	4	D424	000**01011** x	2	
D145	100**10010** x	4	D353	100**01101** x	2	
D146	010**10010** x	4	D354	010**01101** x	2	
D148	001**10010** x	4	D356	001**01101** x	2	
D152	000**10010** x	4	D360	000**01101** x	2	
D273	100**10001** x	4	D225	100**01110** x	2	
D274	010**10001** x	4	D226	010**01110** x	2	
D276	001**10001** x	4	D228	001**01110** x	2	
D280	000**10001** x	4	D232	000**01110** x	2	
D97	100**01100** x	4	D401	100**10011** x	2	
D98	010**01100** x	4	D402	010**10011** x	2	
D100	001**01100** x	4	D404	001**10011** x	2	(A⊕B•C’•D’+C⊕D•A’•B’) •
D104	000**01100** x	4	D408	000**10011** x	2	(G⊕H•H⊕I’+E⊕F+F⊕G) •
D385	100**00011** x	4	D113	100**11100** x	2	(E⊕F’+G⊕H+H⊕I) •
D386	010**00011** x	4	D114	010**11100** x	2	(E⊕F+F⊕H+H⊕I) • K’
D388	001**00011** x	4	D116	001**11100** x	2	
D392	000**00011** x	4	D120	000**11100** x	2	
D161	100**01010** x	4	D337	100**10101** x	2	
D162	010**01010** x	4	D338	010**10101** x	2	
D164	001**01010** x	4	D340	001**10101** x	2	
D168	000**01010** x	4	D344	000**10101** x	2	
D193	100**00110** x	4	D305	100**11001** x	2	
D194	010**00110** x	4	D306	010**11001** x	2	
D196	001**00110** x	4	D308	001**11001** x	2	
D200	000**00110** x	4	D312	000**11001** x	2	
D289	100**01001** x	4	**D**209	100**10110 0**	2	
D290	010**01001** x	4	D210	010**10110** x	2	
D292	001**01001** x	4	D212	001**10110** x	2	
D296	000**01001** x	4	**D**216	000**10110 0**	2	
D321	100**00101** x	4	D177	100**11010** x	2	
D322	010**00101** x	4	D178	010**11010** x	2	
D324	001**00101** x	4	D180	001**11010** x	2	
D328	000**00101** x	4	D184	000**11010** x	2	

Table 4B

Positive Required Disparity PDR

Name	ABCDEFGHI K	DB	Name	ABCDEFGHI K	DB	Coding Label
D85	1010**10100** x	2	D69	1010**00100** x	4	
D86	0110**10100** x	2	D70	0110**00100** x	4	
D89	1001**10100** x	2	D73	1001**00100** x	4	
D90	0101**10100** x	2	D74	0101**00100** x	4	
D83	1100**10100** x	2	D67	1100**00100** x	4	
D92	0011**10100** x	2	D76	0011**00100** x	4	
D101	1010**01100** x	2	D197	1010**00110** x	2	
D102	0110**01100** x	2	D198	0110**00110** x	2	
D105	1001**01100 0**	2	D201	1001**00110 0**	2	
D106	0101**01100** x	2	D202	0101**00110** x	2	
D99	1100**01100** x	2	D195	1100**00110** x	2	
D108	0011**01100** x	2	D204	0011**00110** x	2	
D149	1010**10010** x	2	D325	1010**00101** x	2	
D150	0110**10010** x	2	D326	0110**00101** x	2	
D153	1001**10010** x	2	D329	1001**00101** x	2	
D154	0101**10010** x	2	D330	0101**00101** x	2	
D147	1100**10010** x	2	D323	1100**00101** x	2	
D156	0011**10010** x	2	D332	0011**00101** x	2	
D277	1010**10001** x	2	D37	1010**01000** x	4	{F⊕G•(H’+I’)•E’•K’ +
D278	0110**10001** x	2	D38	0110**01000** x	4	E⊕F•G•H’•I’•K’ +
D281	1001**10001** x	2	D41	1001**01000** x	4	H⊕I•E•F’•G’ +
D282	0101**10001** x	2	D42	0101**01000** x	4	(H+I)•E’•F’•G’} •
D275	1100**10001** x	2	D35	1100**01000** x	4	(A⊕B’•B⊕C•C⊕D’+A⊕B•C⊕D)
D284	0011**10001** x	2	D44	0011**01000** x	4	
D133	1010**00010** x	4	D165	1010**01010** x	2	
D134	0110**00010** x	4	D166	0110**01010** x	2	
D137	1001**00010** x	4	D169	1001**01010** x	2	
D138	0101**00010** x	4	D170	0101**01010 0**	2	
D131	1100**00010** x	4	D163	1100**01010** x	2	
D140	0011**00010** x	4	D172	0011**01010** x	2	
D261	1010**00001** x	4	D293	1010**01001** x	2	
D262	0110**00001** x	4	D294	0110**01001** x	2	
D265	1001**00001** x	4	D297	1001**01001** x	2	
D266	0101**00001** x	4	D298	0101**01001** x	2	
D259	1100**00001** x	4	D291	1100**01001** x	2	
D268	0011**00001** x	4	D300	0011**01001** x	2	
D389	1010**00011** x	2				
D390	0110**00011** x	2				
D393	1001**00011** x	2				
D394	0101**00011** x	2				
D387	1100**00011** x	2				
D396	0011**00011** x	2				

Table 4C

EFGHI	Coding Label
00000	$(G\oplus H'+H\oplus I)\bullet E\oplus F'\bullet F\oplus G'$
00001	
00010	
11111	
11110	
11101	
01**000**	$E\oplus F\bullet G\oplus H'\bullet H\oplus I'$
10**000**	
01**111**	
10**111**	
001**00**	$E\oplus F'\bullet F\oplus H'\bullet H\oplus I'$
110**11**	

Table 5

The Table 4B includes a block of 80 vectors with ABCDK = $A\oplus B\bullet C'\bullet D'\bullet K'+C\oplus D\bullet A'\bullet B'\bullet K'$ grouped into ten dual quartets with five complementary trailing bits EFGHI, which represent 20 of the 32 5-bit combinations. The 12 *missing* vectors are listed in Table 5. The trailing 5 bits of the vectors which are *not* members of the set can be described with the logic expression:

$$\{(G\oplus H'+H\oplus I)\bullet E\oplus F'\bullet F\oplus G'\} + (E\oplus F\bullet G\oplus H'\bullet H\oplus I') + (E\oplus F'\bullet F\oplus H'\bullet H\oplus I')$$

So the trailing 5 bits of the members of the set can be described by the complement of the above expression:

$$(G\oplus H\bullet H\oplus I'+E\oplus F+F\oplus G) \bullet (E\oplus F'+G\oplus H+H\oplus I) \bullet (E\oplus F+F\oplus H+H\oplus I)$$

EFGHI K	Coding Label
00010 x	$(H+I)\bullet E'\bullet F'\bullet G'$
00001 x	
00011 x	
00110 **0**	$F\oplus G\bullet(H'+I')\bullet E'\bullet K'$
00101 x	
00100 x	
01000 x	
01001 x	
01010 **0**	
10010 x	$H\oplus I\bullet E\bullet F'\bullet G'$
10001 x	
10**100** x	$E\oplus F\bullet G\bullet H'\bullet I'\bullet K'$
01**100** **0**	

Table 6

The trailing five bits of a block of 78 vectors in Table 4C with ABCD = $A\oplus B'\bullet B\oplus C\bullet C\oplus D'+A\oplus B\bullet C\oplus D$ grouped into 13 sextets are listed in Table 6. The trailing 5 bits can be identified by the logic expression:

$$F\oplus G\bullet(H'+I')\bullet E'\bullet K'+E\oplus F\bullet G\bullet H'\bullet I'\bullet K'+H\oplus I\bullet E\bullet F'\bullet G'+(H+I)\bullet E'\bullet F'\bullet G'$$

$$PDR = (A\oplus B\bullet C'\bullet D'+C\oplus D\bullet A'\bullet B') \bullet (G\oplus H\bullet H\oplus I'+E\oplus F+F\oplus G) \bullet \qquad (Pn67^*)$$
$$(E\oplus F'+G\oplus H+H\oplus I) \bullet (E\oplus F+F\oplus H+H\oplus I) \bullet K' + \qquad (n66;Pn67^*)$$
$$\{F\oplus G\bullet(H'+I')\bullet E'+E\oplus F\bullet G\bullet H'\bullet I'+H\oplus I\bullet E\bullet F'\bullet G'+(H+I)\bullet E'\bullet F'\bullet G'\} \bullet \qquad (n68)$$
$$(A\oplus B'\bullet B\oplus C\bullet C\oplus D'+A\oplus B\bullet C\oplus D) \bullet K' + \qquad (Pn69^*)$$
$$(A\oplus B\bullet C\bullet D+C\oplus D\bullet A\bullet B) \bullet (G\bullet H'\bullet I'+G'\bullet H\bullet I'+G'\bullet H'\bullet I) \bullet E'\bullet F'\bullet K' + \qquad (Pn70^*)$$
$$\{F\oplus G'\bullet H\oplus I'\bullet(F+H) + F\oplus G\bullet H\oplus I\} \bullet A'\bullet B'\bullet C'\bullet D'\bullet E + G'\bullet H'\bullet K \qquad (n71;n72;Pn73^*)$$

b) Negative Required Disparity for Encoding: NDR

A total of 111 vectors listed in the Table 1 require a negative entry disparity (NDR). They are listed and sorted in Table 7.

Negative Required Disparity NDR

Name	ABCDEFGHI K	DB	Name	ABCDEFGHI K	DB	Coding Label
D437	1010**11011** x	2	D501	1010**11111** x	4	
D438	0110**11011** x	2	D502	0110**11111** x	4	
D441	1001**11011** x	2	D505	1001**11111** x	4	
D442	0101**11011** x	2	D506	0101**11111** x	4	
D435	1100**11011** x	2	D499	1100**11111** x	4	
D444	0011**11011** x	2	C508	0011**11111** **1**	4	
D245	1010**11110** x	2	D469	1010**10111** x	2	
D246	0110**11110** x	2	D470	0110**10111** x	2	(A⊕B’•B⊕C•C⊕D’+A⊕B•C⊕D) • (E⊕F•G•H•I+H⊕I•E•F•G+ E•F•H•I) • (A’•B’•C•D•E•F•G•H•I•K’)’
D249	1001**11110** x	2	D473	1001**10111** x	2	
D250	0101**11110** x	2	D474	0101**10111** x	2	
D243	1100**11110** x	2	D467	1100**10111** x	2	
D252	0011**11110** x	2	D476	0011**10111** x	2	
D373	1010**11101** x	2	D485	1010**01111** x	2	
D374	0110**11101** x	2	D486	0110**01111** x	2	
D377	1001**11101** x	2	D489	1001**01111** x	2	
D378	0101**11101** x	2	D490	0101**01111** x	2	
D371	1100**11101** x	2	D483	1100**01111** x	2	
D380	0011**11101** x	2	D492	0011**01111** x	2	
D497	1000**11111** x	2				
D498	0100**11111** x	2				(A⊕B•C’•D’+C⊕D•A’•B’) • E•F•G•H•I
D500	0010**11111** x	2				
D504	0001**11111** x	2				
D47	**11110**1000 x	0				
D79	**11110**0100 x	0				
D431	**11110**1011 x	4				(F⊕G•H⊕I’+H⊕I•F’•G’+F•G•H’•I) • A•B•C•D•E’
D463	**11110**0111 x	4				
D143	**11110**0010 x	0				
D271	**11110**0001 x	0				
D367	**11110**1101 x	4				

Table 7A

The expression (A’•B’•C•D•E•F•G•H•I•K’)’ in the leading coding label of Table 7A prevents the disparity independent vector D508 from activating NDR. It is a necessary appendix to E•F•H•I but is added as an inhibitor to the entire first group of 36 vectors of the Table 7A to reduce the number of required levels for the logic circuit implementation.

Negative Required Disparity NDR

Name	ABCDEFGHI K	DB	Name	ABCDEFGHI K	DB	Coding Label
D55	111**11000** x	0	D119	111**11100** x	2	
D59	110**11000** x	0	D123	110**11100** x	2	
D61	101**11000** x	0	D125	101**11100** x	2	
D62	011**11000** x	0	D126	011**11100** x	2	
D311	111**11001** x	2	D375	111**11101** x	4	
D315	110**11001** x	2	D379	110**11101** x	4	
D317	101**11001** x	2	D381	101**11101** x	4	
D318	011**11001** x	2	D382	011**11101** x	4	
D183	111**11010** x	2	D247	111**11110** x	4	
D187	110**11010** x	2	D251	110**11110** x	4	
D189	101**11010** x	2	D253	101**11110** x	4	
D190	011**11010** x	2	D254	011**11110** x	4	
D439	111**11011** x	4				
D443	110**11011** x	4				
D445	101**11011** x	4				{(E+F+G•E'+F'+G')•H•I +
D446	011**11011** x	4				E⊕F•H⊕I•G +
D455	111**00111** x	2	D407	111**10011** x	2	(G'+H'+I')•E•F} •
D459	110**00111** x	2	D411	110**10011** x	2	(A⊕B•C•D+C⊕D•A•B)
D461	101**00111** x	2	D413	101**10011** x	2	
D462	011**00111** x	2	D414	011**10011** x	2	
D423	111**01011** x	2	D471	111**10111** x	4	
D427	110**01011** x	2	D475	110**10111** x	4	
D429	101**01011** x	2	D477	101**10111** x	4	
D430	011**01011** x	2	D478	011**10111** x	4	
D487	111**01111** x	4	*D439	111**11011** x	4	
D491	110**01111** x	4	*D443	110**11011** x	4	
D493	101**01111** x	4	*D445	101**11011** x	4	
D494	011**01111** x	4	*D446	011**11011** x	4	
D215	111**10110** x	2	D343	111**10101** x	2	
D219	110**10110** x	2	D347	110**10101** x	2	
D221	101**10110** x	2	D349	101**10101** x	2	
D222	011**10110** x	2	D350	011**10101** x	2	
D231	111**01110** x	2	D359	111**01101** x	2	
D235	110**01110** x	2	D363	110**01101** x	2	
D237	101**01110** x	2	D365	101**01101** x	2	
D238	011**01110** x	2	D366	011**01101** x	2	

Table 7B

EFGHI	Coding Label
11000	(G'+H'+I')•E•F
11001	
11010	
11011	
11100	
11101	
11110	
001**11**	(E+F+G • E'+F'+G')•H•I
010**11**	
011**11**	
100**11**	
101**11**	
110**11***	
10**1**01	E⊕F•H⊕I•G
01**1**01	
10**1**10	
01**1**10	

Table 8

The Table 7B represents a block of 64 vectors with the leading 4 bits as follows
ABCD = A⊕B•C•D+C⊕D•A•B,
grouped into 16 quartets with five matching trailing bits EFGHI as listed in the Table 8 with one group (11011) listed redundantly twice. The trailing bits can be identified by the logic expression:

(E+F+G • E'+F'+G') • H•I + E⊕F•H⊕I•G +
(G'+H'+I') • E•F

NDR = {(E+F+G) • (E'+F'+G') • H•I + E⊕F•H⊕I•G + (G'+H'+I')•E•F} • (Pn74;n75)
(A⊕B•C•D + C⊕D•A•B) + (Pn76*)
(F⊕G•H⊕I' + H⊕I•F'•G' + F•G•H'•I) • A•B•C•D•E' + (n77;Pn78*)
(E⊕F•G•H•I + H⊕I•E•F•G + E•F•H•I) •
(A⊕B'•B⊕C•C⊕D' + A⊕B•C⊕D) • (A'•B'•C•D•E•F•G•H•I•K')' + (Pn79*)
(A⊕B•C'•D' + C⊕D•A'•B') • E•F•G•H•I (Pn80*)

3) Equation for Complementation of the Primary Vector (CMPLP10)

The running disparity at the vector boundaries is constrained to the four values plus or minus one or three. If the required entry disparity PDR or NDR does not match the polarity of running disparity RD, the alternate vector must be used. The alternate vector is generated by complementation of the primary vector. The positive or negative running disparity in front of a byte is referred to as PRDF or NRDF, respectively.

CMPLP10 = PDR•NRDF + NDR•PRDF

The signals PRDF and NRDF are applied preferably separately upstream to each logic cone, instead of to the complete PDR and NDR functions, to eliminate one level of gating. Note that the equality NRDF=PRDF' holds.

4) Equations for the Running Disparity RD (FIG. 13)

FIG. 13 is a state transition diagram for the running disparity RD based an the block disparities DB of the encoded vectors. The vector complementation circuit ensures that the block polarities of vectors conform with the constraints of FIG13. The running disparity can be represented by two flip-flops which pass the value along from vector to vector. The trailing values become the front values of the next encoding cycle. The output of a first flip-flop FFP indicates a positive (PRDF) or negative (NRDF) polarity and the output of a second flip-flop FFA indicates an arithmetic value of one (RD1) or three (RD3).

The two flip-flops can assume arbitrary initial values and disparity violations may be generated initially. At least three unbalanced vectors must be transmitted before payload data transmission is allowed to start. Additional requirements may have to be met before the receiver disparity monitor is in the ready state.

The conditions for complementing these two flip-flops can be derived from FIG13.

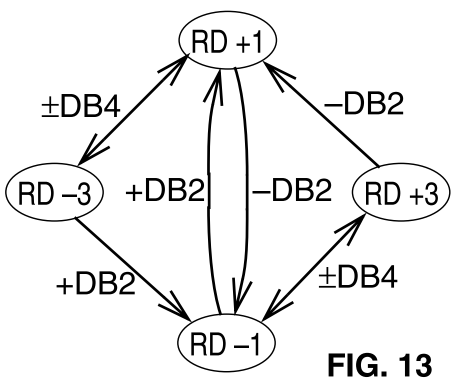

FIG. 13

CMPLFFP = DB2•RD1 + DB4
CMPLFFA = DB2•RD3 + DB4

The block disparity DB2 in the above equation can have a value of ±2 and DB4 can have a value of ±4. RD1 may be RD+1 or RD−1 and RD3 may be RD+3 or RD−3. The polarities of the above parameters can be ignored for purposes of the above two disparity equations because the complementation function CMPLP10 enforces compliance.

a) Block Disparity of Four for Encoding: DB4

EFGHI	Coding Label
11**0**00	E⊕F'•F⊕H• H⊕I'•G'
00**0**11	
10**0**10	E⊕F•H⊕I•G'
01**0**10	
10**0**01	
01**0**01	
10**100**	E⊕F•G•H'•I'
01**100**	
00110	H⊕I•E'•F'•G
00101	

Table 9A

The Tables 4 and 7 include 70 and 29 vectors, respectively, with a block disparity of four.

The Table 9A lists the trailing 5 bits of 10 quartets in the left column of Table 4B. The leading four bits of all these 10 quartets can be defined by A⊕B•C'•D' + C⊕D•A'•B'.

EFGHI K	Coding Label
00010 x	H⊕I•E'•F'•G'
00001 x	
00100 x	F⊕G•E'•H'•I'
01000 x	
11111 y	E•F•G•H•I

Table 9B

The Table 9B lists the trailing 5 bits of 4 sextets of Table 4C and one sextet from Table 7A which includes one vector (C508) with K=1. The leading four bits of all these five sextets can be defined by A⊕B'•B⊕C•C⊕D'+A⊕B•C⊕D. The value of y in the K column is one for C508 and zero for D508. The data vector D508 has zero disparity and is excluded by the expression (A'•B'•C•D•E•F•G•H•I•K')'.

EFGHI	Coding Label
11101	H⊕I•E•F•G
11110	
10111	F⊕G•E•H•I
11011	
01111	E'•F•G•H•I

Table 9C

The Table 9C lists the trailing 5 bits of 5 quartets of Table 7B. The leading four bits of all these 4 quartets can be defined by A⊕B•C•D+C⊕D•A•B.

The 6 vectors of Table 4A with DB=4 are defined by the equation (F⊕H•G⊕I+F⊕G•H⊕I) • A'•B'•C'•D'•E.

The vectors D367, D431, and D463 of Table 7A are defined by A•B•C•D•E'•I • (F•G•H'+F•G'•H+F'•G•H).

$$
\begin{aligned}
DB4 = {} & (E\oplus F'\bullet F\oplus H\bullet H\oplus I'\bullet G' + E\oplus F\bullet H\oplus I\bullet G' + E\oplus F\bullet G\bullet H'\bullet I' + H\oplus I\bullet E'\bullet F'\bullet G) \bullet && (n81)\\
& (A\oplus B\bullet C'\bullet D' + C\oplus D\bullet A'\bullet B') && (n65;Pn82)\\
& (H\oplus I\bullet E'\bullet F'\bullet G' + F\oplus G\bullet E'\bullet H'\bullet I' + E\bullet F\bullet G\bullet H\bullet I) \bullet && (n83)\\
& (A\oplus B'\bullet B\oplus C\bullet C\oplus D' + A\oplus B\bullet C\oplus D) \bullet (A'\bullet B'\bullet C\bullet D\bullet E\bullet F\bullet G\bullet H\bullet I\bullet K')' + && (Pn84^*)\\
& (H\oplus I\bullet E\bullet F\bullet G+F\oplus G\bullet E\bullet H\bullet I+E'\bullet F\bullet G\bullet H\bullet I) \bullet (A\oplus B\bullet C\bullet D + C\oplus D\bullet A\bullet B) + && (Pn85)\\
& (F\oplus H\bullet G\oplus I + F\oplus G\bullet H\oplus I) \bullet A'\bullet B'\bullet C'\bullet D'\bullet E + && (Pn86^*)\\
& (F\bullet G\bullet H' + F\bullet G'\bullet H + F'\bullet G\bullet H) \bullet A\bullet B\bullet C\bullet D\bullet E'\bullet I && (n87)
\end{aligned}
$$

b) Block Disparity of Two for Encoding: DB2

EFGHI K	Coding Label
10110 0	E⊕F•H⊕I•G•K'
01110 x	
10101 x	
01101 x	
01011 x	E⊕F•G'•H•I
10011 x	
11001 x	H⊕I•E•F•G'
11010 x	
00111 x	E⊕F'•H⊕I'• (F+H)•G
11100 x	
11111 x	

Table 10A

A total of 116 vectors listed in the Table 4 and 74 vectors listed in Table 7 have a block disparity of two.

The expression G'•H'•K is taken directly form the top of Table 4A. It represents 10 optional control vectors for 16B18B code, which are not valid for contiguous 9B10B vectors.

The Table 10A lists the trailing 5 bits of 10 quartets of Table 4B and one quartet from Table 7A. The leading four bits of these 11 quartets can be defined by A⊕B•C'•D' + C⊕D•A'•B'

EFGHI K	Coding Label
1010**0** x	E⊕F•G⊕H•I'•K'
0110**0 0**	
1001**0** x	
0101**0 0**	
00110 **0**	E⊕F'•H⊕I•G•K'
00101 x	
11110 x	
11101 x	
10**001** x	E⊕F•G⊕H'•I
01**001** x	
10**111** x	
01**111** x	
11**011** x	E⊕F'•G'•H•I
00**011** x	

Table 10B

The Table 10B lists the trailing five bits of a 9 sextets from Table 4C and 5 sextets from Table 7A. The leading four bits of these 14 sextets can be defined by A⊕B'•B⊕C•C⊕D'+A⊕B•C⊕D.

EFGHI K	Coding Label
11**0**01 x	E⊕F'•H⊕I•G'
11**0**10 x	
00**0**01 x	
00**0**10 x	
10**1**10 x	E⊕F•H⊕I•G
01**1**10 x	
10**1**01 x	
01**1**01 x	
01**011** x	E⊕F•G'•H•I
10**011** x	
00**100 0**	E⊕F'•G•H'•I'•K'
11**100** x	
00**111** x	E'•F'•G•H•I

Table 10C

The Table 10C lists the trailing five bits of 3 quartets from Table 4A and 10 quartets from Table 7B. The leading four bits of all these 14 quartets can be defined by
A⊕B•C•D+C⊕D•A•B.

DB2 = (E⊕F'•H⊕I•G' + E⊕F•H⊕I•G + E⊕F'•G•H'•I'•K' + E⊕F•G'•H•I + *(n92)*
E'•F'•G•H•I) • (A⊕B•C•D+C⊕D•A•B) + *(n94*; Pn93*;Pn97)*
{E⊕F'•H⊕I'•(F+H)•G + E⊕F•H⊕I•G + E⊕F•G'•H•I + H⊕I•E•F•G'} • *(n88)*
(A⊕B•C'•D' + C⊕D•A'•B') • K' + *(n65*;Pn89*;Pn95*)*
(E⊕F'•H⊕I•G + E⊕F•G⊕H•I' + E⊕F•G⊕H'•I + E⊕F'•G'•H•I) • *(n90)*
(A⊕B'•B⊕C•C⊕D' + A⊕B•C⊕D) • K' + G'•H'•K *(Pn91*;Pn96*;n94*;n98*)*

B. Logic Equations for 10B9B Decoding

It is a feature of this code that only balanced and disparity independent vectors are subject to individual bit changes and the complementation of entire vectors for disparity control is limited to primary vectors for which the source bits ABCDEFGHI are identical to the encoded bits abcdefghi. Consequently, bit decoding and complementation can be executed independently of each other in parallel.

1) Individual Bit Decoding

The bit decoding tables can be developed from the bit encoding Tables 3a, 3b, 3c, 3d, 3e, 3f, 3g, 3h, and 3i by substitution of the bits abcdefghi for ABCDEFGHI and a separate table for the control bit K. Some of the tables show both complementary bit sets and identical bit sets in the left and the right column; they are illustrated in italic and bold face type, respectively.

The j-bit has a value of one for all vectors which require individual bit modifications or full vector complementation for decoding and consequently, the j-position is eliminated from the Tables 11A through 11I. In the circuits, the j-bit value is added near the end of each logic cone which ostensible adds one logic level, but this gating level is required for the complementation of entire vectors anyway and the two functions can be implemented with an AOI21 gate with a circuit delay and area which are comparable to typical primitive logic gates.

The logic equations for X1 are developed below. X1 is the command to complement an individual bit x where x stands for any one of the leading 9 encoded bits. The respective decoded bits X are generated by a circuit implementation of the equation as shown on the right side of FIG. 17C.

$$X = (X1 \bullet j) \oplus x$$

Two circuit simplification methods are available, but if two bit positions of a set of vectors are ignored, all four possible combinations must be examined for correct operation:

1. The decoding equations can be simplified if we allow arbitrary bit changes for the decoding of invalid vectors. Appropriate invalid vectors can be added to the vectors defining a logic expression. In the following, these redundant vectors are not shown, but the terms of logic expressions which can be eliminated by their inclusion are *over-lined*. Vectors with a leading or trailing run of five are easily recognized as invalid.
2. The value of a bit position before decoding of that bit can be ignored because for this code, the same bit position of a vector which is complementary in that position and equal in all other positions is an alternate or an invalid vector. Alternate vectors are complemented for decoding, as an example, D16=**1**001100011 has the first bit complemented to 0, but the entire vector **0**001100011 (D231A) is complemented for decoding. However, for decoding classes which are applicable to several bits, the redundant bit is usually included to enable circuit sharing but *underlined* in the logic equations to indicate that it could be left out perhaps to reduce delay in a critical path.

The table labels include all terms, but the equations do not include the terms which are not included in the circuits.

Decoded Bit A

The ‘a’ column has bold entries in the Tables 1 and 2 for the 31 vectors listed in Table 11A. The A-bit decoding equation is derived from the coding labels of Table 11A.

A-bit Decoding

Name	abcdefghi	A	S	Name	abcdefghi	A	Coding Label
D96	101001100	0	1	D415	010100011	1	(f⊕i•g⊕h + f⊕g•h⊕i) • (d⊕g•g⊕i’)’ • <u>a⊕b</u>•b⊕c•c⊕d•e’
D384	101000011	0	1	D127	010101100	1	
D160	101001010	0	1	D351	010100101	1	
D192	101000110	0	1	D319	010101001	1	
D288	101001001	0	1	D223	010100110	1	
D320	101000101	0	1	D191	011001010	1	<u>a⊕b</u>•b⊕g•f⊕g•g⊕h•h⊕i•c•d’•e’
D16	100110001	0	1	D255	011001100	1	<u>a⊕b</u>•b⊕c’•c⊕d•d⊕e’ • e⊕f•f⊕g’•g⊕i • h’
D32	100101001	0	1	D447	011001001	1	<u>a⊕b</u>•b⊕c’•c⊕d • e’•i • (f•g’•h’ + f’•g•h’ + f’•g’•h)
D64	100100101	0	1	D383	011000101	1	
D128	100100011	0	1	D479	011000011	1	
D224	100001110	0	1	D495	000101101	1	<u>a⊕d</u>•h⊕i • b’•c’ • e’•f•g
D352	100001101	0	1	D239	000101110	1	
D0	111000100	0	1	D272	101010001	0	b⊕e•e⊕g•g⊕i • <u>a</u>•c•d’•f’•h’
				D31	001110001	1	(e⊕f • g’•h’•i + h⊕i • e’•f’•g + e’•f•g•h’•i’) • <u>a</u>’•b’•c•d
				D63	001101001	1	
				D159	001100110	1	
				D341	001100101	1	
				D15	001101100	1	

Table 11A

A1 = (f⊕i•g⊕h + f⊕g•h⊕i) • (d⊕g•g⊕i’)’ • <u>a⊕b</u>•b⊕c•c⊕d • e’ + (Pn0*)
<u>a⊕b</u>•b⊕c’•c⊕d • d⊕e’•e⊕f•f⊕g’•g⊕i • h’ + (Pn0*)
(e⊕f • g’•h’•i + h⊕i • e’•f’•g + e’•f•g•h’•i’) • <u>a</u>’•b’•c•d + h⊕i • b’•c’•e’•f•g + (n2)
<u>a⊕b</u>•b⊕c’•c⊕d • e’•i • (f•g’•h’ + f’•g•h’ + f’•g’•h) + (Pn1*)
b⊕g•f⊕g•g⊕h•h⊕i • c•d’•e’ + b⊕e•e⊕g•g⊕i • c•d’•f’•h’ (Pn0*;Pn1*)

Decoded Bit B

The ‘b’ column has bold entries in the Tables 1 and 2 for the 15 vectors listed in Table 11B.

B-bit Decoding

Name	abcdefghi	B	S	Name	abcdefghi	B	Coding Label
D48	010110001	0	1	D464	010010110	0	d⊕g•g⊕h’•h⊕i • a’•b•c’•e•f’
D80	011100100	0	1	D432	011010001	0	d⊕e•e⊕g•g⊕i • a’•b•c•f’•h’
D0	111000100	0	1	D448	011000110	0	a⊕h • b•c•d’•e’ • f’•g•i’
				D15	001101100	1	(f⊕h • e’•g•i’ + e⊕f • g’•h’•i) • a’•b’•c•d
				D159	001100110	1	
				D31	001110001	1	
				D63	001101001	1	
				D511	100101010	1	a•b’•c’•d • e’•f • g’•h•i’
D416	010001110	0	1	D239	000101110	1	b⊕d•h⊕i • a’•c’ • e’•f•g
D480	010001101	0	1	D495	000101101	1	

Table 11B

B1 = (f⊕h • e’•g•i’ + e⊕f • g’•h’•i) • a’•b’•c•d + a⊕h • b•c•d’•e’ • f’•g•i’ + *(Pn5*)*
d⊕g•g⊕h’•h⊕i • a’•c’•e•f’ + d⊕e•e⊕g•g⊕i • a’•c•f’•h’ + *(Pn3*)*
h⊕i • a’•c’•e’•f•g + a•b’•c’•d • e’•f • g’•h•i’ *(Pn3*;Pn5*)*

Decoded Bit C

The ‘c’ column has bold entries in the Tables 1 and 2 for the 31 vectors listed in Table 11C.

C-bit Decoding

Name	abcdefghi	C	S	Name	abcdefghi	C	Coding Label
D96	101001100	0	1	D415	010100011	1	(f⊕i•g⊕h+f⊕g•h⊕i) • (d⊕g•g⊕i’)’ • a⊕b•b⊕c•c⊕d • e’
D384	101000011	0	1	D127	010101100	1	
D160	101001010	0	1	D351	010100101	1	
D192	101000110	0	1	D319	010101001	1	
D288	101001001	0	1	D223	010100110	1	
D80	011100100	0	1	D287	110010001	1	a⊕c•d⊕e•e⊕g•g⊕i • b•f’•h’
D432	011010001	0	1	D95	110100100	1	
D240	001001110	0	1	D495	000101101	1	c⊕d•h⊕i • a’•b’•e’•f•g
D368	001001101	0	1	D239	000101110	1	
D144	001010110	0	1	D272	101010001	0	a⊕h•g⊕h’•h⊕i • b’•c•d’•e•f’
D0	111000100	0					a⊕h • b•c•d’•e’•f’•g•i’
D448	011000110	0					
D256	001100011	0	1	D320	101000101	0	a⊕d•a⊕h•g⊕h • b’•c • e’•f’•i
				D111	110001100	1	(f⊕i•g⊕h + f⊕g•h⊕i) • a•b•c’•d’•e’
				D399	110000011	1	
				D175	110001010	1	
				D207	110000110	1	
				D303	110001001	1	
				D335	110000101	1	
				D511	100101010	1	a•b’•c’•d•e’•f•g’•h•i’

Table 11C

C1 = (f⊕i•g⊕h + f⊕g• h⊕i) • (d⊕g• g⊕i’)’ • a⊕b•b⊕c•c⊕d • e’ + (Pn6*)
a⊕d•a⊕h•g⊕h • b’•e’•f’•i + (Pn6*)
(f⊕i•g⊕h + f⊕g•h⊕i) • a•b•c’•d’•e’ + h⊕i • a’•b’ • e’•f•g + (Pn7)
d⊕e•e⊕g•g⊕i • b•f’•h’ + a⊕h•g⊕h’•h⊕i • b’•d’•e • f’ + (Pn9*)
a⊕h • b•c•d’•e’ • f’•g•i’ + a•b’•c’•d • e’•f • g’•h•i’ (n8*)

Decoded Bit D

The ‘d’ column has bold entries in the Tables 1 and 2 for the 19 vectors listed in Table 11D.

D-bit Decoding

Name	abcdefghi	D	S	Name	abcdefghi	D	Coding Label
D32	100101001	0	1	D447	011001001	1	(a⊕b•b⊕c’•c⊕d) • e’•i • (f•g’•h’+f’•g•h’+ f’•g’•h)
D64	100100101	0	1	D383	011000101	1	
D128	100100011	0	1	D479	011000011	1	
D16	100110001	0	1	D255	011001100	1	a⊕b•b⊕c’•c⊕d•d⊕e’ • e⊕f•f⊕g’•g⊕i • h’
D80	011100100	0	1	D256	001100011	0	b⊕g’•g⊕h•h⊕i’ • a’•c•d•e’•f’
D341	001100101	0	1	D191	011001010	1	b⊕d•d⊕g’•f⊕g•g⊕h•h⊕i•a’•c•e’
				D287	110010001	1	a•b•c’•d’•e•f’•g’•h’•i
				D111	110001100	1	(f⊕i•g⊕h + f⊕g•h⊕i) • a•b•c’•d’•e’
				D399	110000011	1	
				D175	110001010	1	
				D207	110000110	1	
				D303	110001001	1	
				D335	110000101	1	

Table 11D

D1 = a⊕b•b⊕c’•c⊕d • e’•i • (f•g’•h’ + f’•g•h’ + f’•g’•h) + (Pn10*)
a⊕b•b⊕c’•c⊕d•d⊕e’ • e⊕f•f⊕g’•g⊕i • h’ + (Pn10*)
(f⊕i•g⊕h + f⊕g•h⊕i) • a•b•c’•d’ • e’ + (Pn11*;Pn10*)
b⊕g•f⊕g•g⊕h•h⊕i • a’•c•e’ + (Pn10*)
b⊕g’•g⊕h•h⊕i’ • a’•c•e’ • f’ + a•b•c’•d’ • e•f’•g’ • h’•i (Pn11*)

Decoded Bit E

The ‘e’ column has bold entries in the Tables 1 and 2 for the 45 vectors listed in Table 11E.

E-bit Decoding

Name	abcdefghi	E	S	Name	abcdefghi	E	Coding Label
D1	100010101	0	1	D510	011100010	1	(a⊕b•c⊕d'•d⊕e + a⊕b'•b⊕e•c⊕d) • e⊕g'•g⊕h•h⊕i • f'
D2	010010101	0	1	D509	101100010	1	
D4	001010101	0	1	D507	110100010	1	
D8	000110101	0	1	D503	111000010	1	
D80	011100100	1	1	D508	001101010	1	a⊕c•b⊕g'•f⊕g•g⊕h • d•e'•i'
D95	110100100	1	1	D511	100101010	1	
D3	110010100	0					(a⊕d•b⊕c + a⊕b•c⊕d) • e•f'•g•h'•i'
D12	001110100	0					
D5	101010100	0					
D6	011010100	0					
D9	100110100	0					
D10	010110100	0					
D65	100011100	0					(a⊕b • c'•d' + c⊕d • a'•b') • (g•h'•i' + g'•h•i' + g'•h'•i) • e•f
D66	010011100	0					
D68	001011100	0					
D72	000111100	0					
D129	100011010	0					
D130	010011010	0					
D132	001011010	0					
D136	000111010	0					
D257	100011001	0					
D258	010011001	0					
D260	001011001	0					
D264	000111001	0					
				D127	010101100	1	(f⊕i•g⊕h + f⊕g•h⊕i) • (d⊕g • h•i')' • c⊕d • a'•b•e'
				D415	010100011	1	
				D319	010101001	1	
				D351	010100101	1	
				D223	010100110	1	
				D255	011001100	1	
				D479	011000011	1	
				D383	011000101	1	
				D447	011001001	1	
				D191	011001010	1	
				D63	001101001	1	(d⊕f•h⊕i • g + d•f•g'•h'•i) • a'•b'•c•e'
				D159	001100110	1	
				D341	001100101	1	
				D240	00$\bar{1}$001110	1	
				D368	00$\bar{1}$001101	1	

Table 11E

$$E1 = (a\oplus b\bullet c\oplus d'\bullet d\oplus \underline{e} + a\oplus b'\bullet b\oplus \underline{e}\bullet c\oplus d) \bullet \underline{e}\oplus g'\bullet g\oplus h\bullet h\oplus i \bullet f' + \qquad \text{(Pn12*)}$$
$$(a\oplus d\bullet b\oplus c + a\oplus b\bullet c\oplus d) \bullet \underline{e}\bullet f'\bullet g\bullet h'\bullet i' + \qquad \text{(Pn12*;Pn15*)}$$
$$(a\oplus b \bullet c'\bullet d' + c\oplus d \bullet a'\bullet b') \bullet (g\bullet h'\bullet i' + g'\bullet h\bullet i' + g'\bullet h'\bullet i) \bullet \underline{e}\bullet f + \qquad \text{(Pn13*)}$$
$$(f\oplus i\bullet g\oplus h + f\oplus g\bullet h\oplus i) \bullet (d\oplus g \bullet h\bullet i')' \bullet c\oplus d \bullet a'\bullet b + \qquad \text{(Pn16*)}$$
$$a\oplus c\bullet b\oplus g'\bullet f\oplus g\bullet g\oplus h \bullet d\bullet i' + \qquad \text{(Pn13*)}$$
$$(d\oplus f\bullet h\oplus i \bullet g + d\bullet f\bullet g'\bullet h'\bullet i) \bullet a'\bullet b'\bullet c \qquad \text{(n15*)}$$

Decoded Bit F

The ‘f’ column has bold entries in the Tables 1 and 2 for the 19 vectors listed in Table 11F.

F-bit Decoding

Name	abcdefghi	F	Name	abcdefghi	F	Coding Label
D65	1000**11**100	0				(a⊕b • c’•d’+c⊕d • a’•b’) • (g•h’•i’ + g’•h•i’ + g’•h’•i) • e•f̲
D66	0100**11**100	0				
D68	0010**11**100	0				
D72	0001**11**100	0				
D129	1000**11**010	0				
D130	0100**11**010	0				
D132	0010**11**010	0				
D136	0001**11**010	0				
D257	1000**11**001	0				
D258	0100**11**001	0				
D260	0010**11**001	0				
D264	0001**11**001	0				
D15	001101100	0				a’•b’•c•d • e’•f̲•g•h’•i’
			D510	0111**00010**	1	(a⊕b•c•d+c⊕d•a•b) • e’•f̲’•g’•h•i’
			D509	1011**00010**	1	
			D507	1101**00010**	1	
			D503	1110**00010**	1	
			D432	**0110**1**0**0**01**	1	e̲⊕g•a’•b•c•d’•f̲’•h’•i
			D383	**0110**0**0**1**01**	1	

Table 11F

$$F1 = (a\oplus b \bullet c'\bullet d'+c\oplus d \bullet a'\bullet b') \bullet (g\bullet h'\bullet i' + g'\bullet h\bullet i' + g'\bullet h'\bullet i) \bullet e\bullet \underline{f} + \qquad \text{(Pn18*)}$$
$$(a\oplus b\bullet c\bullet d+c\oplus d\bullet a\bullet b) \bullet e'\bullet g'\bullet h\bullet i' + \qquad \text{(Pn19)}$$
$$e\oplus g \bullet a'\bullet b\bullet c\bullet d'\bullet h'\bullet i + a'\bullet b'\bullet c\bullet d \bullet e'\bullet \underline{f}\bullet g\bullet h'\bullet i' \qquad \text{(n17)}$$

Decoded Bit G

The ‘g’ column has bold entries in the Tables 1 and 2 for the 22 vectors listed in Table 11G.

G-bit Decoding

Name	abcdefghi	G	S	Name	abcdefghi	G	Coding Label
D1	100010101	0	1	D510	011100010	1	(a⊕b•c⊕d’•d⊕e + a⊕b’•b⊕e•c⊕d) • e⊕g’•g⊕h•h⊕i • f’
D2	010010101	0	1	D509	101100010	1	
D4	001010101	0	1	D507	110100010	1	
D8	000110101	0	1	D503	111000010	1	
D144	001010110	0	1	D416	010001110	0	b⊕c•c⊕f•e⊕f • a’•d’•g•h•i’
D159	001100110	0	1	D479	011000011	1	b⊕d•d⊕g’•g⊕i • a’•c•e’•f’•h
D0	111000100	0	0				a⊕b’•b⊕d•d⊕f’ • c•e’•g•h’•i’
D15	001101100	0	0				
				D511	100101010	1	a⊕c • b’•d•e’•f•g’•h•i’
				D508	001101010	1	
D3	110010100	0					(a⊕d•b⊕c + a⊕b•c⊕d) • e•f’•g•h’•i’
D12	001110100	0					
D5	101010100	0					
D6	011010100	0					
D9	100110100	0					
D10	010110100	0					

Table 11G

G1= (a⊕b•c⊕d’•d⊕e + a⊕b’•b⊕e•c⊕d) • e⊕g’•g⊕h•h⊕i • f’ + *(Pn50*)*
a⊕b’•b⊕d•d⊕f’•c•e’•h’•i’ + a⊕c • b’•d•e’•f•g’•h•i’ *(Pn50*;n51*)*
(a⊕d•b⊕c + a⊕b•c⊕d) • e•f’•g•h’•i’ + *(Pn50*;n51*)*
b⊕c•c⊕f•e⊕f • a’•d’•h•i’ + b⊕d•d⊕g’•g⊕i • a’•c•e’•f’•h *(Pn52*)*

Decoded Bit H

The ‘h’ column has bold entries in the Tables 1and 2 for the 20 vectors listed in Table 11H.

H-bit Decoding

Name	abcdefghi	H	S	Name	abcdefghi	H	Coding Label
D256	**00**_11_**0**_001_**1**	0	1	D495	**00**_0_**10**_110_**1**	1	$c\oplus f\bullet f\oplus g'\bullet \underline{g\oplus h} \bullet a'\bullet b'\bullet d\bullet e'\bullet i$
D19	1100**10010**	0					$(a\oplus d\bullet b\oplus c+a\oplus b\bullet c\oplus d) \bullet e\bullet f'\bullet g'\bullet \underline{h}\bullet i'$
D28	0011**10010**	0					
D21	1010**10010**	0					
D22	0110**10010**	0					
D25	1001**10010**	0					
D26	0101**10010**	0					
D17	10**00**10**011**	0					$(a\oplus b \bullet c'\bullet d'+c\oplus d \bullet a'\bullet b') \bullet e\oplus f \bullet g'\bullet \underline{h}\bullet i$
D18	01**00**10**011**	0					
D20	**00**1010**011**	0					
D24	**00**0110**011**	0					
D33	10**00**01**$\bar{0}$11**	0					
D34	01**00**01**$\bar{0}$11**	0					
D36	**00**1001**$\bar{0}$11**	0					
D40	**00**0101**$\bar{0}$11**	0					
				D255	**011001**100	1	$g\oplus i \bullet a'\bullet b\bullet c\bullet d' \bullet e'\bullet f\bullet \underline{h'}$
				D447	**011001**001	1	
				D432	**01**1**0**100**01**	1	$c\oplus f\bullet e\oplus f\bullet f\oplus g' \bullet a'\bullet b\bullet d'\bullet \underline{h'}\bullet i$
				D480	**01**0**0**011**01**	1	

Table 11H

$$H1 = (a\oplus d\bullet b\oplus c + a\oplus b\bullet c\oplus d) \bullet e\bullet f'\bullet g'\bullet i' + \qquad (Pn53^*;n55^*)$$

$$(a\oplus b \bullet c'\bullet d' + c\oplus d \bullet a'\bullet b') \bullet e\oplus f \bullet g'\bullet \underline{h}\bullet i + \qquad (Pn53^*;Pn54^*)$$

$$c\oplus f\bullet e\oplus f\bullet f\oplus g' \bullet a'\bullet b\bullet d'\bullet i + c\oplus f\bullet f\oplus g' \bullet a'\bullet b'\bullet d\bullet e'\bullet i + \qquad (Pn54^*;Pn53^*)$$

$$g\oplus i \bullet a'\bullet b\bullet c\bullet d'\bullet e'\bullet f \qquad (n55^*)$$

Decoded Bit I

The ‘i’ column has bold entries in the Tables 1 and 2 for the 32 vectors listed in Table 11I.

I-bit Decoding

Name	abcdefghi	I	S	Name	abcdefghi	I	Coding Label
D1	10001**0**101	0	1	D510	01110**0**010	1	(a⊕b•c⊕d’•d⊕e +
D2	01001**0**101	0	1	D509	10110**0**010	1	a⊕b’•b⊕e•c⊕d) •
D4	00101**0**101	0	1	D507	11010**0**010	1	e⊕g’•g⊕h•h⊕i • f’
D8	00011**0**101	0	1	D503	11100**0**010	1	
D48	**010**110001	0	1	D416	**010**001110	1	d⊕e’•e⊕f•f⊕g’•g⊕h’•h⊕i • a’•b•c’
D7	1110**00001**	0					
D11	1101**00001**	0					(c⊕d • a•b + a⊕b•c•d) •
D13	1011**00001**	0					e’•f’•g’•h’•i
D14	0111**00001**	0					
D16	1**00**110**001**	0					
D32	1**00**101**001**	0					a⊕c•e⊕f • b’•d•g’•h’•i
D31	0**01**110**001**	0					
D63	0**01**101**001**	0					
D64	**100100**101	0					g⊕h • a•b’•c’•d • e’•f’•i
D128	**100100**011	0					
D17	100010**011**	0					
D18	010010**011**	0					
D20	001010**011**	0					
D24	000110**011**	0					(a⊕b • c’•d’ + c⊕d • a’•b’) •
D33	100001**011**	0					e⊕f • g’•h•i
D34	010001**011**	0					
D36	001001**011**	0					
D40	000101**011**	0					
				D464	**010**0**10110**	1	c⊕e •a’•b•d’•f’ • g•h•i’
				D448	**011**0**00110**	1	
				D511	1**00101010**	1	a⊕c • b’•d • e’•f•g’•h•i’
				D508	0**01101010**	1	

Table11I

I1 = (a⊕b•c⊕d’•d⊕e + a⊕b’•b⊕e•c⊕d) • e⊕g’•g⊕h•h⊕i • f’ + (Pn56*)
d⊕e’•e⊕f•f⊕g’•g⊕h’ • a’•b•c’ + (Pn59*)
(a⊕b • c’•d’ + c⊕d • a’•b’) • e⊕f • g’•h•i + (Pn56*)
(c⊕d • a•b + a⊕b • c•d) • e’•f’•g’•h’ + a⊕c•e⊕f • b’•d•g’•h’•i + (n57)
g⊕h•a•b’•c’•d • e’•f’ + c⊕e•a’•b•d’•f’ • g•h•i’ + a⊕c • b’•d•e’•f•g’•h•i’ (n58)

Control Bit K

The primary and alternate versions of 18 control vectors at the trailing end of Table 1M are listed in Table 11K. In the absence of errors, a 10-bit vector aligned with the vector boundaries can be identified as the control character C508 by a run length of 7 in bits c through i because of code constraints. For some applications it may be advisable to check all 10 bits for improved error immunity. The equation for K-bit decoding below is derived from the coding labels of Table 11K.

K-bit Decoding

Name	abcdefghij	K	S	Name	abcdefghij	K	Coding Label
K201	**1001**0**01101**	1					(d⊕e•a+a'•d•e) • b'•c'•f'•g•h•i'•j
K209	**100**01**01101**	1					
K216	0**001**1**01101**	1					
K77	**101**1**100**0**1001**	1					c⊕f • a•b'•d•e'•g•h'•i'•j
K105	**100**1**01**1**1001**	1					
K170	*0101010101*	1	1	K341	*1010101010*	1	a⊕b•b⊕c•c⊕d•d⊕e• e⊕f•f⊕g•g⊕h•h⊕i•i⊕j
C508P	$\overline{00}1111111\overline{0}$	1	1	C508A	$\overline{11}0000000\overline{1}$	1	c⊕d'•d⊕e'•e⊕f'• f⊕g'•g⊕h'•h⊕i'
K39P°	$11100\overline{1}0000$	1	1	K39A*	$00011\overline{0}1111$	1	(a⊕b'•c⊕d•b⊕e•e⊕g' + a⊕b•c⊕d'•d⊕e•e⊕g' + a⊕d•b⊕c•e⊕g + a⊕b•c⊕d•e⊕g) • $\overline{f\oplus g}$•g⊕h'•h⊕i'•i⊕j'
K43P°	$11010\overline{1}0000$	1	1	K43A*	$00101\overline{0}1111$	1	
K45P°	$10110\overline{1}0000$	1	1	K45A*	$01001\overline{0}1111$	1	
K46P°	$01110\overline{1}0000$	1	1	K46A*	$10001\overline{0}1111$	1	
K51P°	$11001\overline{1}0000$	1	1	K51A*	$00110\overline{0}1111$	1	
K60P°	$00111\overline{1}0000$	1	1	K60A*	$11000\overline{0}1111$	1	
K53P°	$10101\overline{1}0000$	1	1	K53A*	$01010\overline{0}1111$	1	
K54P°	$01101\overline{1}0000$	1	1	K54A*	$10010\overline{0}1111$	1	
K57P°	$10011\overline{1}0000$	1	1	K57A*	$01100\overline{0}1111$	1	
K58P°	$01011\overline{1}0000$	1	1	K58A*	$10100\overline{0}1111$	1	

Table 11K

° Optional control vector for 16B18B code, not valid for contiguous 9B10B vectors.

K = (a⊕b'•c⊕d•b⊕e•e⊕g' + a⊕b•c⊕d'•d⊕e•e⊕g' + a⊕d•b⊕c•e⊕g + *(n72*)*
a⊕b•c⊕d•e⊕g) • g⊕h'•h⊕i'•i⊕j' + *(n72*;Pn60*)*
(d⊕e•a+a'•d•e)•b'•c'•f'•g•h•i'•j + c⊕d'•d⊕e'•e⊕f'•f⊕g'•g⊕h'•h⊕i' + *(NK*;Pn62*)*
a⊕b•b⊕c•c⊕d•d⊕e•e⊕f•f⊕g•g⊕h•h⊕i•i⊕j + c⊕f • a•b'•d•e'•g•h'•i'•j + *(n61)*

2) Full Vector Complementation

It is helpful to remember that for this code all alternate vectors have a j-bit value of one and the only vectors with j=1 which are not alternate vectors are the 116 balanced, disparity independent vectors BM4c'4t'6t'J of FIG. 10 listed in Tables 2A and 2B. The equation for the complementation of alternate vectors can thus be expressed by:

CMPL10 = j • (BM4c'4t'6t')'

An expression in terms of bit values for BM4c’4t’6t’ can be derived from the trellis of FIG. 10. The left side of Table 12 lists the bit patterns of FIG. 10 from node 0b to the nodes 4u, 4b, and 4m, and the right side lists the bit patterns from nodes 4u, 4b, and 4m to node M. The number of vectors represented is 4•5 + 6•10 + 4•9 = 116.

116 Balanced, Disparity independent Vectors of FIG. 10 with j=1

Nodes	abcd	Coding Label	Nodes	e f g h i	Coding Label
				10**000**	
	1110			01**000**	e⊕f•g’•h’•i’ + g⊕h•e’•f’•i’ +
0-4u	**11**01	a⊕b•c•d+c⊕d•a•b	4u-M	**00**10**0**	h⊕i•e’•f’•g’
	10**11**			**00**01**0**	
	01**11**			**00**010*	
				00001	
				1100**0**	
				0011**0**	
	1100			1010**0**	
	0011			0110**0**	
0-4b	1010	a⊕d•b⊕c +	4b-M	1001**0**	e⊕f’•f⊕g•g⊕h’•i’ + e⊕f•g⊕h•i’ +
	0110	a⊕b•c⊕d		0101**0**	e⊕f•g’•h’•i + g⊕h•e’•f’•i
	1001			10**001**	
	0101			01**001**	
				00101	
				00011	
				11100	
				10011	
	10**00**			**1**1010	
0-4m	01**00**	a⊕b•c’•d’ +	4m-M	**1**1001	(f⊕i•g⊕h + f⊕g•h⊕i)•e +
	0010	c⊕d•a’•b’		**1**0110	(g’•h•i + g•h’•i + g•h•i’) • e’•f
	0001			**1**0101	
				01110	
				01101	
				01011	

Table 12

CMPL10 = j • {(e⊕f’•f⊕g•g⊕h’•i’ + e⊕f•g⊕h•i’ + e⊕f•g’•h’•i + g⊕h•e’•f’•i) • *(n80*)*
(a⊕d•b⊕c + a⊕b•c⊕d) + *(n80)*
[(f⊕i•g⊕h + f⊕g•h⊕i)•e + (g’•h•i + g•h’•i + g•h•i’) • e’•f)] • *(n82*;n83*)*
(a⊕b•c’•d’ + c⊕d•a’•b’) + *(n82;n83)*
(e⊕f•g’•h’•i’ + g⊕h•e’•f’•i’ + h⊕i•e’•f’•g’) • (a⊕b•c•d + c⊕d•a•b)}’ *(n81)*

On the upper right side in the circuit diagram of FIG. 17C, the part of the equation for CMPL10 within the brackets { } is referred to by the net name PBM4cn4tn6tn which references the trellis of FIG. 10 up to node M.

3) Invalid Characters

Since there are 828 valid vectors in the code (with all optional control vectors included), there are 196 invalid vectors. They are listed in Table 13. The first two rows represent 124 vectors with a leading or trailing run of five. The letter x indicates arbitrary values for the bit positions involved. Each of the top two rows represents 64 vectors but only 124 vectors together because of overlap. The third row is a complementary vector pair with a disparity of four not included in FIGS. 2C.1 or 2C.2. This is followed by 10 complementary vector pairs with a disparity of two and a leading run of four not included in FIG. 2B, and a complementary set of 25 vector pairs with disparity of six and not ending or starting with a run of five. The overlined bit positions are redundant because the opposite value would generate a leading or trailing run of five.

$$INV = (b\oplus c'\bullet c\oplus d'\bullet d\oplus e'\bullet e\oplus f' + a\oplus b'\bullet b\oplus c'\bullet c\oplus f'\bullet d\oplus e + a\oplus d'\bullet d\oplus e'\bullet e\oplus f'\bullet b\oplus c) \qquad (n63)$$
$$\bullet\ (f\oplus g'\bullet g\oplus h'\bullet i\oplus j + f\oplus i'\bullet i\oplus j'\bullet g\oplus h) + \qquad (Pn64^*)$$
$$(a\oplus b'\bullet b\oplus c'\bullet a\oplus j'\bullet d\oplus e + a\oplus d'\bullet d\oplus e'\bullet a\oplus j'\bullet b\oplus c + b\oplus c'\bullet c\oplus d'\bullet d\oplus e'\bullet d\oplus g') \bullet \qquad (n65)$$
$$g\oplus h'\bullet h\oplus i'\bullet i\oplus j' + \qquad (Pn66^*)$$
$$(f\oplus g'\bullet g\oplus h'\bullet h\oplus i\bullet i\oplus j' + f\oplus i\bullet g\oplus h + f\oplus g\bullet h\oplus i)\bullet a\oplus b'\bullet b\oplus c'\bullet c\oplus d'\bullet a\oplus j + \qquad (n70)$$
$$(f\oplus g\bullet h\oplus i'\bullet i\oplus j + c\oplus f\bullet f\oplus g'\bullet h\oplus i) \bullet a\oplus b'\bullet b\oplus c'\bullet c\oplus d'\bullet a\oplus j' + \qquad (Pn67;n68)$$
$$a\oplus b'\bullet b\oplus c'\bullet c\oplus d'\bullet d\oplus e' + f\oplus g'\bullet g\oplus h'\bullet h\oplus i'\bullet i\oplus j' \qquad (Pn71^*)$$

For concatenated 10B vectors, the optional control vectors identified by the expression INVK below must also be included in the set of invalid vectors.

$$INVK = (a\oplus b'\bullet c\oplus d\bullet b\oplus e\bullet e\oplus g' + a\oplus b\bullet c\oplus d'\bullet d\oplus e\bullet e\oplus g' +$$
$$a\oplus d\bullet b\oplus c\bullet e\oplus g + a\oplus b\bullet c\oplus d\bullet e\oplus g) \bullet g\oplus h'\bullet h\oplus i'\bullet i\oplus j' \qquad (Pn60)$$

4) Disparity Checks on Decoding

Disparity checks serve a variety of purposes with different implementations depending on the application. As an example, long distance, multi-hop carrier type applications require a simple in line quality monitoring system as described by Sharland et al. (Ref.9) for the case of a 7B8B code. Computer links use such checks to help in the isolation of failing link components and to supplement higher level error checking schemes in the goal of weeding out all flawed frames or packets.

Some important applications of this code will not be helped much by disparity monitoring and not implement it. As an example, a computer bus as described in Ref. 8 requires separate extensive error checking and correction facilities with low latency. Disparity errors often show up with some delay after one or more disparity independent coding blocks have passed.

Some applications may implement simplified monitoring circuits which miss a small fraction of disparity violations, or they may tolerate some double counts, or they may want to deactivate monitoring until a reliable running disparity value is reestablished after an error indication. Some expressions which can be used as building blocks for any such monitoring process are defined below.

196 Invalid Vectors

abcdefghij	abcdefghij	Coding Label
11111xxxxx	00000xxxxx	a⊕b'•b⊕c'•c⊕d'•d⊕e' + f⊕g'•g⊕h'•h⊕i'•i⊕j'
xxxxx11111	xxxxx00000	
1111$\overline{0}$01110	0000$\overline{1}$00011	(f⊕g'•g⊕h'•h⊕i•i⊕j' + f⊕i•g⊕h + f⊕g•h⊕i)• a⊕b'•b⊕c'•c⊕d'•a⊕j
1111$\overline{0}$11000	0000$\overline{1}$00111	
1111$\overline{0}$00110	0000$\overline{1}$11001	
1111$\overline{0}$10100	0000$\overline{1}$01011	
1111$\overline{0}$01100	0000$\overline{1}$10011	
1111$\overline{0}$10010	0000$\overline{1}$01101	
1111$\overline{0}$01010	0000$\overline{1}$10101	
1111$\overline{0}$10001	0000$\overline{1}$01110	(f⊕g•h⊕i'•i⊕j + c⊕f•f⊕g'•h⊕i) • a⊕b'•b⊕c'•c⊕d'•a⊕j'
1111$\overline{0}$01001	0000$\overline{1}$10110	
1111$\overline{0}$00101	0000$\overline{1}$11010	
1111$\overline{0}$00011	0000$\overline{1}$11100	
11110$\overline{0}$1111	00001$\overline{1}$0000	(a⊕b'•b⊕c'•a⊕j'•d⊕e + a⊕d'•d⊕e'•a⊕j'•b⊕c + b⊕c'•c⊕d'•d⊕e'•d⊕g') • g⊕h'•h⊕i'•i⊕j'
11101$\overline{0}$1111	00010$\overline{1}$0000	
11011$\overline{0}$1111	00100$\overline{1}$0000	
10111$\overline{0}$1111	01000$\overline{1}$0000	
$\overline{0}$1111$\overline{0}$1111	$\overline{1}$0000$\overline{1}$0000	
1111$\overline{0}$11110	0000$\overline{1}$00001	(b⊕c'•c⊕d'•d⊕e'•e⊕f' + a⊕b'•b⊕c'•c⊕f'•d⊕e + a⊕d'•d⊕e'•e⊕f'•b⊕c)•(f⊕g'•g⊕h'•i⊕j + f⊕i'•i⊕j'•g⊕h)
1111$\overline{0}$11101	0000$\overline{1}$00010	
1111$\overline{0}$11011	0000$\overline{1}$00100	
1111$\overline{0}$10111	0000$\overline{1}$01000	
111$\overline{0}$111110	000$\overline{1}$000001	
111$\overline{0}$111101	000$\overline{1}$000010	
111$\overline{0}$111011	000$\overline{1}$000100	
111$\overline{0}$110111	000$\overline{1}$001000	
11$\overline{0}$1111110	00$\overline{1}$0000001	
11$\overline{0}$1111101	00$\overline{1}$0000010	
11$\overline{0}$1111011	00$\overline{1}$0000100	
11$\overline{0}$1110111	00$\overline{1}$0001000	
1$\overline{0}$11111110	0$\overline{1}$00000001	
1$\overline{0}$11111101	0$\overline{1}$00000010	
1$\overline{0}$11111011	0$\overline{1}$00000100	
1$\overline{0}$11110111	0$\overline{1}$00001000	
$\overline{0}$111111110	$\overline{1}$000000001	
$\overline{0}$111111101	$\overline{1}$000000010	
$\overline{0}$111111011	$\overline{1}$000000100	
$\overline{0}$111110111	$\overline{1}$000001000	

Table 13

For some applications, the disparity circuits are less latency sensitive than the rest of the decoding circuits because system performance is not affected by modest delay in the error detection and perhaps more than one clock cycle is acceptable for the execution of these functions. Therefore, they can be generated by logic synthesis programs rather than a hand-crafted design and no circuit design for disparity monitoring is shown in this report.

At a receiver, the vector sequences can be monitored to see whether they still conform to the rules imposed by the encoder. A single or odd number of errors in transmission will always cause a violation of the disparity rules without necessarily generating an invalid vector as described above. In a mixture of balanced vectors, and vectors with a block disparity of ±2 or ±4, the running disparity in the absence of errors is constrained to values of ±1 and ±3 at the vector boundaries. A transmission error is not always immediately detectable by just adding and subtracting the cumulative block disparities to see whether the actual running disparity of the received vector sequence meets the above constraints. The following rules assume that the error, if any, occurred before the two-vector blocks under consideration. If an error is present in the block itself, a duplicate error indication may occur later because the value of the original running disparity following an error is uncertain. The rules apply to any mixture of vectors in the sequence such as 6B, 8B, 10B, or other vectors with compatible disparity characteristics.

An error is flagged if the required polarity of the entry disparity of a received coded block does not match the polarity of the running disparity at the start of that block.

5) Equations for Required Disparity on Decoding (DR)

a) Positive Required Disparity PDR

Any valid or invalid vector in FIG. 1(L) ending in nodes 10m, 10t, 10q, 10s, or 10n and the 9 balanced vectors of FIG. 2A.2(R) require a positive entry disparity. These vectors can be grouped and defined as follows:

- 3 or more zeros in the 5 leading bit positions combined with 3 or more zeros in the 5 trailing positions.
- 4 or more zeros in the 5 leading bit positions combined with 2 or more zeros in the 5 trailing positions.
- 2 or more zeros in the 5 leading bit positions combined with 4 or more zeros in the 5 trailing positions.
- 5 or more zeros in the 6 leading bit positions or 4 leading zeros

$$
\begin{aligned}
PDR = {} & \{a'\bullet b'\bullet(c'+d'+e') + d'\bullet e'\bullet(a'+b'+c') + (a'+b')\bullet(d'+e')\bullet c'\} \bullet \\
& \quad \{f'\bullet g'\bullet(h'+i'+j') + i'\bullet j'\bullet(f'+g'+h') + (f'+g')\bullet(i'+j')\bullet h'\} + \\
& \{a'\bullet b'\bullet c'\bullet(d'+e') + c'\bullet d'\bullet e'\bullet(a'+b') + a'\bullet b'\bullet d'\bullet e'\} \bullet \\
& \quad \{f'\bullet(g'+h'+i'+j') + g'\bullet(h'+i'+j') + h'\bullet(i'+j')+i'\bullet j'\} + \\
& \{a'\bullet(b'+c'+d'+e') + b'\bullet(c'+d'+e') + c'\bullet(d'+e')+d'\bullet e'\} \bullet \\
& \quad \{f'\bullet g'\bullet h'\bullet(i'+j') + h'\bullet i'\bullet j'\bullet(f'+g') + f'\bullet g'\bullet i'\bullet j'\} + \\
& a'\bullet b'\bullet e'\bullet f'\bullet(c'+d') + c'\bullet d'\bullet e'\bullet f'\bullet(a'+b') + a'\bullet b'\bullet c'\bullet d'
\end{aligned}
$$

b) Negative Required Disparity NDR

The equation for the negative required disparity NDR is the same as for PDR but with complementary bit values.

$$
\begin{aligned}
NDR = &\{a \bullet b \bullet (c+d+e)+d \bullet e \bullet (a+b+c)+(a+b) \bullet (d+e) \bullet c\} \bullet \\
&\{f \bullet g \bullet (h+i+j)+i \bullet j \bullet (f+g+h)+(f+g) \bullet (i+j) \bullet h\} + \\
&\{a \bullet b \bullet c \bullet (d+e)+c \bullet d \bullet e \bullet (a+b)+a \bullet b \bullet d \bullet e\} \bullet \{f \bullet (g+h+i+j)+g \bullet (h+i+j)+h \bullet (i+j)+i \bullet j\} + \\
&\{a \bullet (b+c+d+e)+b \bullet (c+d+e)+c \bullet (d+e)+d \bullet e\} \bullet \{f \bullet g \bullet h \bullet (i+j)+h \bullet i \bullet j \bullet (f+g)+f \bullet g \bullet i \bullet j\} + \\
&a \bullet b \bullet e \bullet f \bullet (c+d) + c \bullet d \bullet e \bullet f \bullet (a+b)+a \bullet b \bullet c \bullet d
\end{aligned}
$$

6) Equations for Running Disparity on Decoding (RD)

The running disparity is determined by the characteristics of the most recent one or two disparity dependent vectors. Quicker recovery of the running disparity is possible by looking at the three most recent disparity dependent vectors, but the added complexity is probably not worthwhile for most applications. Disparity independent blocks are ignored. From the state diagram of FIG. 13 in section III.A.4 above, it is evident that after a block disparity of 4 (DB4), the polarity (PRD/NRD) is known, but not the arithmetic value (RD1/RD3). It also shows that the arithmetic value is RD1 after any block with a disparity of 2 (DB2). The running disparity is at +1 after DB2 of either polarity followed by PDB2 with a positive disparity or after PDB2 followed by one of 9 disparity dependent balanced vectors PDB0 with a positive required entry disparity RD (D47A, D55A, D59A, D61A, D62A, D79A, D143A, D271A, D496). The running disparity is at –1 after DB2 of either polarity followed by NDB2 with a negative disparity or after NDB2 followed by one of 9 disparity dependent balanced vectors NDB0 with a negative required entry disparity (D47, D55, D59, D61, D62, D79, D143, D271, D496A). The primary version of these vectors is illustrated in the trellises of FIGS. 5A, 5B, and 5C.

The Table 14 illustrates how the running disparity can be initially established or reestablished after an error and is used to extract the equations below for the polarity and the arithmetic value of the running disparity. The following acronyms are used:

PRD = Positive Running Disparity   NRD = Negative Running Disparity
PDB4 = Positive Block Disparity of 4   NDB4 = Negative Block Disparity of 4
PDB2 = Positive Block Disparity of 2   NDB2 = Negative Block Disparity of 2
RD1, RD3 = Arithmetic value of running disparity is equal 1 or 3, respectively
PDB0 = D47A, D55A, D59A, D61A, D62A, D79A, D143A, D271A, D496
NDB0 = D47, D55, D59, D61, D62, D79, D143, D271, D496A

The appended letter L(ast) refers to the next preceding disparity dependent block.

$$
\begin{aligned}
PRD &= PDB4 + PDB2 \bullet (PDB2L + NDB2L) + PDB0 \bullet PDB2L \\
NRD &= NDB4 + NDB2 \bullet (PDB2L + NDB2L) + NDB0 \bullet NDB2L \\
RD1 &= PDB2 + NDB2 + (PDB4 + NDB4) \bullet RD3L \\
RD3 &= (PDB4 + NDB4) \bullet RD1L
\end{aligned}
$$

Running Disparities PRD, NRD, RD1, RD3

PRD	NRD	RD1	RD3	PDB4	NDB4	PDB2	NDB2	PDB0	NDB0	PDB2L	NDB2L	RD1L	RD3L
1				1									
1						1				1	1		
1								1		1			
	1				1								
	1						1			1	1		
	1								1		1		
		1				1	1						
		1		1	1								1
			1	1	1							1	

Table 14

7) Equations for Block Disparity (DB)

Invalid vectors which simplify the equations are included and such vectors with more than seven ones or zeros are lumped together with vectors of a disparity of four.

a) Positive Block Disparity of Four PDB4

All vectors of this set contain at least seven ones and end with nodes 10x, 10h, 10v, or 10c in the trellis of FIG. 1(L). Invalid vectors with fewer than 3 ones in the leading or trailing 5 bit positions are not included. The vectors belong to one of the following two groups:

- 4 or 5 ones in the 5 leading bit positions combined with 3 or more ones in the 5 trailing 4 positions.
- 3 or more ones in the 5 leading bit positions combined with 4 or 5 ones in the 5 trailing positions.

$$PDB4=\{(a+b)\bullet c\bullet d\bullet e + (d+e)\bullet a\bullet b\bullet c + a\bullet b\bullet d\bullet e\} \bullet \{(h+i+j)\bullet f\bullet g + (f+g+h)\bullet i\bullet j + (f+g)\bullet (i+j)\bullet h\} + \{(c+d+e)\bullet a\bullet b + (a+b+c)\bullet d\bullet e + (a+b)\bullet (d+e)\bullet c\} \bullet \{(i+j)\bullet f\bullet g\bullet h + (f+g)\bullet h\bullet i\bullet j + f\bullet g\bullet i\bullet j\}$$

b) Negative Block Disparity of Four NDB4

The equation for the negative block disparity NDB4 is the same as for PDB4 but with complementary bit values.

$$NDB4 = \{(a'+b')\bullet c'\bullet d'\bullet e' + (d'+e')\bullet a'\bullet b'\bullet c' + a'\bullet b'\bullet d'\bullet e'\} \bullet \{(h'+i'+j')\bullet f'\bullet g' + (f'+g'+h')\bullet i'\bullet j' + (f'+g')\bullet (i'+j')\bullet h'\} + \{(c'+d'+e')\bullet a'\bullet b' + (a'+b'+c')\bullet d'\bullet e' + (a'+b')\bullet (d'+e')\bullet c'\} \bullet \{(i'+j')\bullet f'\bullet g'\bullet h' + (f'+g')\bullet h'\bullet i'\bullet j' + f'\bullet g'\bullet i'\bullet j'\}$$

c) Positive Block Disparity of Two PDB2

This set includes all vectors with exactly 6 ones ending with node 10u in FIG. 1(L). Some invalid vectors with 5 leading or trailing ones are included with the assumption that they originated from valid vectors with only 4 ones in the respective 5 bit positions.

- 3 ones in the 5 leading bit positions combined with 3 ones in the 5 trailing bit positions.
- 2 ones in the 5 leading bit positions combined with 4 or 5 ones in the trailing 5 positions.
- 4 or 5 ones in the 5 leading bit positions combined with 2 ones in the trailing 5 positions.

a b c d e	Coding Label
11100	$a\oplus e \bullet b\oplus d \bullet c$
00111	
10110	$a\oplus b \bullet d\oplus e \bullet c$
01110	
10101	
01101	
10011	$a\oplus b \bullet c' \bullet d \bullet e$
01011	
11010	$d\oplus e \bullet a \bullet b \bullet c'$
11001	

Table 15

The Table 15 lists all ten 5-bit leading sequences with 3 ones and 2 zeros, suitably ordered for identification by labels. For the trailing 5 bits with 3 ones, 'abcde' is substituted by 'fghij'.

The coding labels for 2 ones and 3 zeros in the leading 5 positions are the same except that all bit positions not associated with an exclusive OR function have complementary values.

Four or five ones in the trailing 5 positions can be expressed by:
$(f+g) \bullet h \bullet i \bullet j + (i+j) \bullet f \bullet g \bullet h + f \bullet g \bullet i \bullet j$

Four or five ones in the leading 5 positions can be expressed by:
$(a+b) \bullet c \bullet d \bullet e + (d+e) \bullet a \bullet b \bullet c + a \bullet b \bullet d \bullet e$

The resulting equation for PDB2 follows:

$$\begin{aligned} PDB2 = &\{(a\oplus e \bullet b\oplus d + a\oplus b \bullet d\oplus e) \bullet c + (a\oplus b \bullet d \bullet e + d\oplus e \bullet a \bullet b) \bullet c'\} \bullet \\ &\quad \{(f\oplus j \bullet g\oplus i + f\oplus g \bullet i\oplus j) \bullet h + (f\oplus g \bullet i \bullet j + i\oplus j \bullet f \bullet g) \bullet h'\} + \\ &\{(a\oplus e \bullet b\oplus d + a\oplus b \bullet d\oplus e) \bullet c' + (a\oplus b \bullet d' \bullet e' + d\oplus e \bullet a' \bullet b') \bullet c\} \bullet \\ &\quad \{(f+g) \bullet h \bullet i \bullet j + (i+j) \bullet f \bullet g \bullet h + f \bullet g \bullet i \bullet j\} + \\ &\{(a+b) \bullet c \bullet d \bullet e + (d+e) \bullet a \bullet b \bullet c + a \bullet b \bullet d \bullet e\} \bullet \\ &\quad \{(f\oplus j \bullet g\oplus i + f\oplus g \bullet i\oplus j) \bullet h' + (f\oplus g \bullet i' \bullet j' + i\oplus j \bullet f' \bullet g') \bullet h\} \end{aligned}$$

d) Negative Block Disparity of Two NDB2

The equation for the negative block disparity NDB2 is the same as for PDB2 but with complemented bit values other than those associated with an exclusive OR function.

$$\begin{aligned} NDB2 = &\{(a\oplus e \bullet b\oplus d + a\oplus b \bullet d\oplus e) \bullet c' + (a\oplus b \bullet d' \bullet e' + d\oplus e \bullet a' \bullet b') \bullet c\} \bullet \\ &\quad \{(f\oplus j \bullet g\oplus i + f\oplus g \bullet i\oplus j) \bullet h' + (f\oplus g \bullet i' \bullet j' + i\oplus j \bullet f' \bullet g') \bullet h\} + \\ &\{(a\oplus e \bullet b\oplus d + a\oplus b \bullet d\oplus e) \bullet c + (a\oplus b \bullet d \bullet e + d\oplus e \bullet a \bullet b) \bullet c'\} \bullet \\ &\quad \{(f'+g') \bullet h' \bullet i' \bullet j' + (i'+j') \bullet f' \bullet g' \bullet h' + f' \bullet g' \bullet i' \bullet j'\} + \\ &\{(a'+b') \bullet c' \bullet d' \bullet e' + (d'+e') \bullet a' \bullet b' \bullet c' + a' \bullet b' \bullet d' \bullet e'\} \bullet \\ &\quad \{(f\oplus j \bullet g\oplus i + f\oplus g \bullet i\oplus j) \bullet h + (f\oplus g \bullet i \bullet j + i\oplus j \bullet f \bullet g) \bullet h'\} \end{aligned}$$

e) Zero Block Disparity with a positive required front end disparity PDB0

This vector set can be derived from FIG. 2A.2(R).

$$\begin{aligned} PDB0 = &(f\oplus g \bullet h \bullet i \bullet j + i\oplus j \bullet f \bullet g \bullet h + f \bullet g \bullet h' \bullet i \bullet j) \bullet a' \bullet b' \bullet c' \bullet d' \bullet e + \\ &(a\oplus b \bullet c' \bullet d' + c\oplus d \bullet a' \bullet b') \bullet e' \bullet f' \bullet g \bullet h \bullet i \bullet j \end{aligned}$$

f) Zero Block Disparity with a negative required front end disparity NDB0

This vector set can be derived from FIG. 2A.2(L) and is the same as for PDB0 but with complemented bit values.

$$NDB0 = (f \oplus g \bullet h' \bullet i' \bullet j' + i \oplus j \bullet f' \bullet g' \bullet h' + f' \bullet g' \bullet h \bullet i' \bullet j') \bullet a \bullet b \bullet c \bullet d \bullet e' + (a \oplus b \bullet c \bullet d + c \oplus d \bullet a \bullet b) \bullet e \bullet f \bullet g' \bullet h' \bullet i' \bullet j'$$

IV. CIRCUIT IMPLEMENTATION

For the circuit implementation, it is assumed that all inputs are available in complementary form, i.e. both the +L2 and –L2 outputs of the input register latches are made available. Nevertheless, the assumption is that the –L2 outputs are slightly delayed relative to the +L2 outputs.

The circuit diagrams show only NAND, NOR, INV, XOR, XNOR, and AOI21 gates and a single OR4 gate in a non-critical path in FIG. 17A (Pn5). The use of AND and OR gates has been avoided because of their increased delays. For the NAND and NOR gates, the upper inputs of the logic symbols usually have less delay than the lower ones. The presumed critical paths are therefore routed through the top inputs. The wire routing also assumes that XNOR delays are shorter than XOR delays. The gate representations use bubble notation. A bubble indicates a lower logic level. The functions indicated by the symbols are true if the inputs and outputs are at the levels indicated. Functions suggested by net names are true when at the level indicated by the first letter, P for the upper level and N or n for the lower level.

There is some leeway in the definition of the basic logic equations and in the partitioning of the longer expressions to match the fan-in limitations of the gates. Variations in these choices leads to different ranges in circuit sharing and circuit counts. In circuit areas which are suspected to be in the upper range of circuit delay, the circuit count has occasionally been increased to reduce delay primarily by reducing the fan-in of gates in the critical path. For delay considerations, both XOR and XNOR gates have been used at the input to generate both polarities and some of those gates can be replaced by INV circuits once simulation results are available. Similarly, the circuit diagrams generally do not show complex gates to allow maximum circuit sharing; the logic processing programs will introduce complex gates automatically where appropriate.

Note that some of the logic variables of the equations are not present explicitly in the circuit diagrams. If so, they have been merged with other functions in a single gate to reduce overall circuit delay. An example is the variable PDR which is only present in the merged signal NRDFaPDR of FIG. 15C.

A. Circuit for Encoding

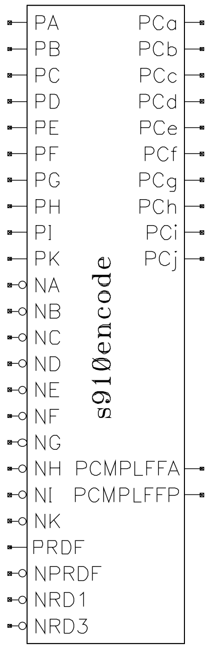

FIG. 14

1) Block Diagram (FIG.14)

The block diagram for the encoding circuit with all inputs and outputs is shown in FIG. 14.

2) Gate Level Circuit Diagram (FIGS.15A, 15B, 15C)

A gate-level circuit diagram of the encoder of FIG. 14 is shown in FIGS. 15A, 15B, and 15C which represent a single circuit with net sharing.

a) Individual Bit Complementation

Fig. 15A shows most of the encoding of the leading 5 bits (abcde), the encoding of the trailing 5 bits (fghij) is shown in FIG. 15B. The upper right side of FIG. 15C shows the last two gate levels for bit encoding. The center right side lists a number of EXCLUSIVE OR (XOR and XNOR) gates which are shared across the three encoding circuit diagrams. Some of these gates can be replaced by inverters driven from the gate of opposite polarity if they are not part of any critical timing path.

b) Full Vector Complementation Circuit

The signal CMPL10 which complements all 10 bits of a coded byte is orthogonal to the signals (Ca1, Cb1, Cc1, Cd1, Ce1, Cf1, Cg1, Ch1, Ci1) which cause complementation of individual bits. In other words, both for encoding and decoding, no individual bits are changed when a full vector is complemented and vice-versa. This feature allows the merger of both types of signals in a single OR function as shown at the upper right side of FIG. 15C, greatly simplifying the circuitry preceding the output EXCLUSIVE OR function. The upper left part of FIG. 15C shows the implementation of the equations for the complementation of entire vectors. The CMPL10 signal is not explicitly present in the circuit version shown. It is dependent on the required entry disparity and the starting running disparity RDF which is equal to the ending disparity RDT of the preceding byte. Note that the value of RDF is not required immediately at the start of the encoding interval because in the critical signal paths it is typically an input to a gate at the third of fourth level which facilitates pipe-lining of this logic path into the next cycle if required, as described in Ref. 11 for an 8B10B code.

c) Disparity Control

The bottom part of FIG. 15C shows the equation for the determination whether the polarity and/or absolute value of the running disparity at the end of the new vector has to be changed (CMPLFFP, CMPLFFA). Because these two signals typically feed a flip-flop with a multiplexer input which has a longer setup time than a regular flip-flop, extra gates have been added to reduce the number of logic levels to 6.

9B10B Bit Encoding abcde

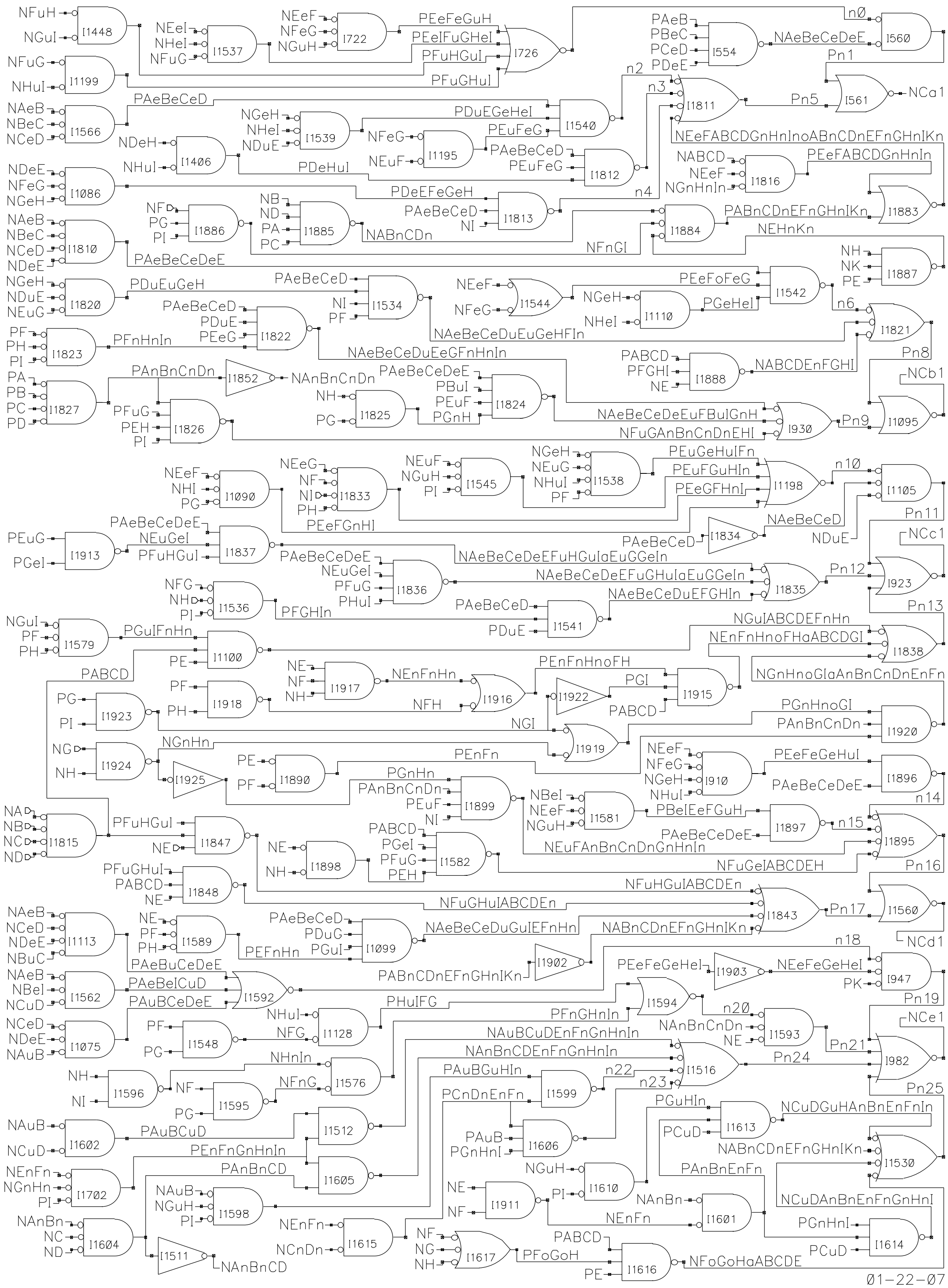

FIG. 15A

9B10B Bit Encoding fghij

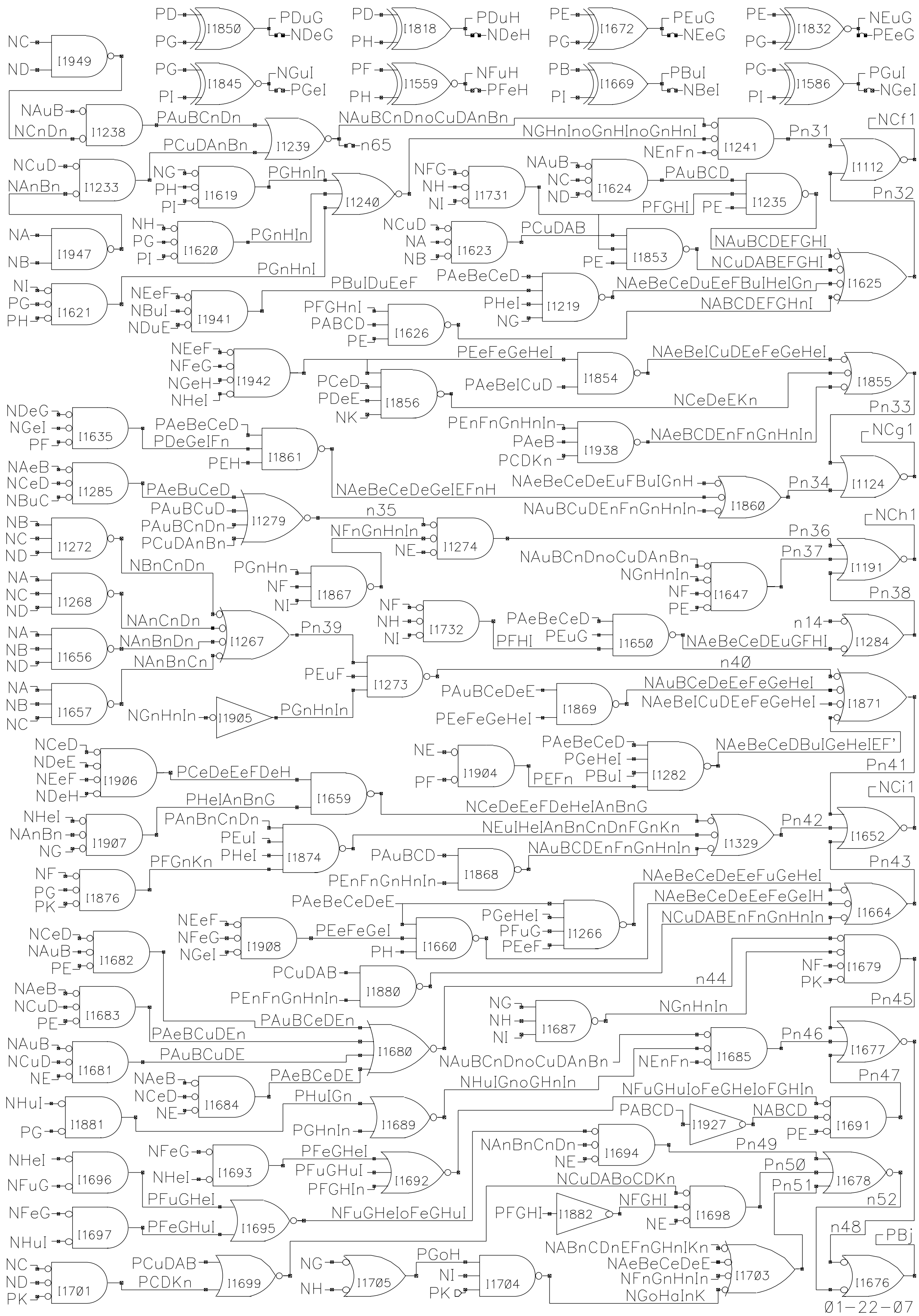

FIG. 15B

9B10B Disparity Control

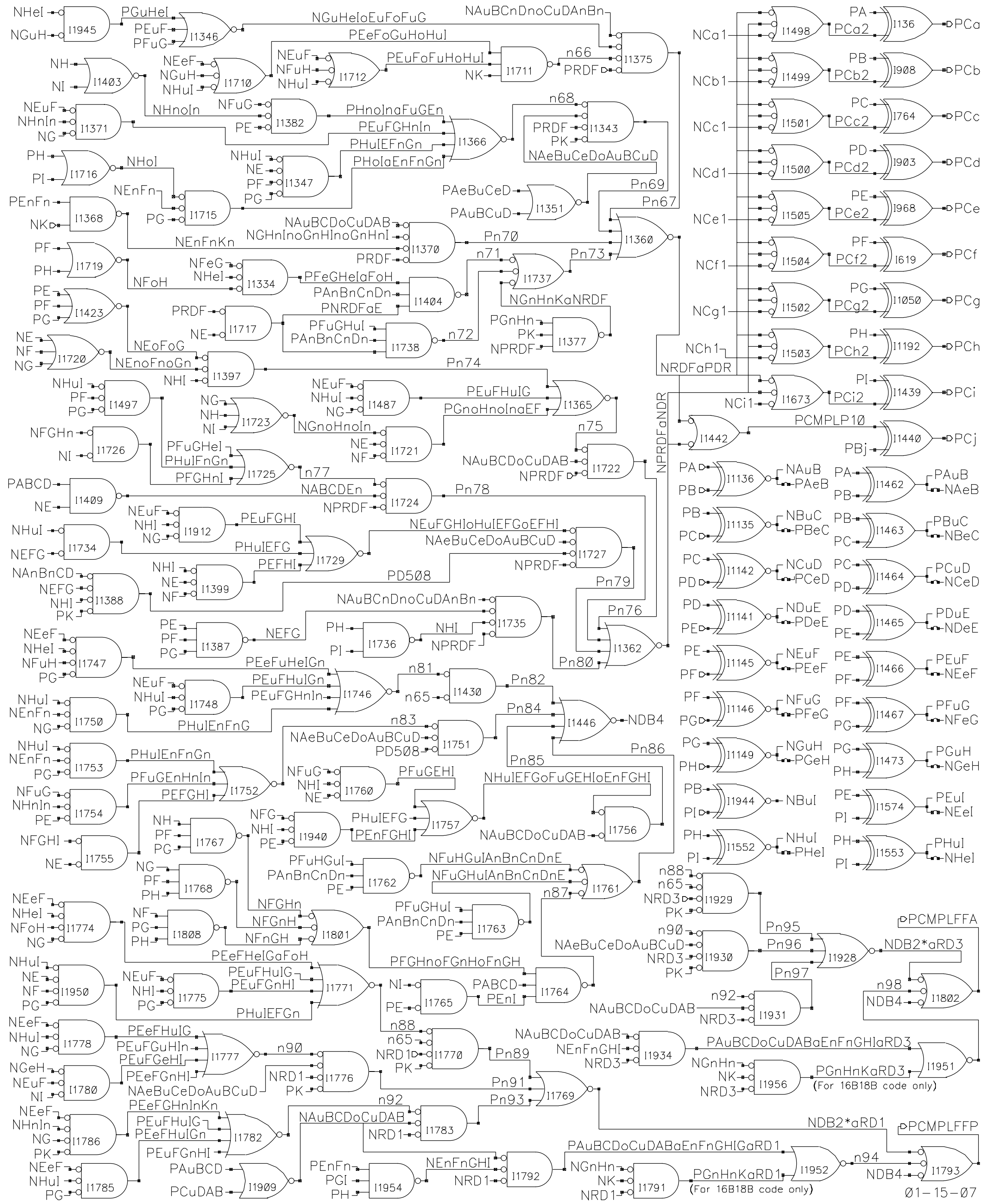

FIG. 15C

3) Gate Count, Circuit Delays and Pipe-Lining for Encoding

The encoder circuit shown comprises 352 gates and two flip-flops (not shown) to keep track of the disparity. No logic path exceeds 7 gates; all gates are of the inverting type with shorter delay except some XOR gates which for most power and loading levels have comparable or only slightly more delay than XNOR gates. The circuit area can be reduced by an estimated 5% to 10% if 8 gating levels are acceptable.

If the circuit does not meet the performance goals, the first step is to reduce the fan-in of gates in the critical paths by off loading the shorter sections of the logic cone with some additional gates. Pipe-lining can result in larger delay reductions. To this end, the fan-in for the trailing 3 logic levels has been kept low to reduce the number of parameters which must be carried forward. Minor rearrangements may be useful depending on whether one, two, or three trailing logic levels are moved into a second clock cycle which can reduce the first cycle to four logic levels.

A further delay reduction can be accomplished by itself or in combination with any of the above versions by minor circuit modifications and moving some of the leading EXCLUSIVE OR functions into the preceding clock cycle in the data source path.

B. Decoding Circuit

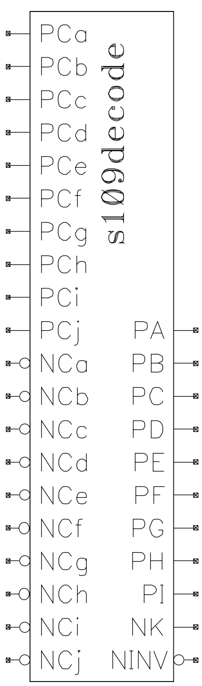

FIG. 16

1) Block Diagram (FIG. 16)

The block diagram for the decoding circuit with all inputs and outputs is shown in FIG. 16.

2) Gate Level Circuit Diagram (FIGS. 17A, 17B, 17C)

a) Individual Bit Complementation and Validity Check

A gate-level circuit diagram of the decoder of FIG. 16 is shown in FIGS. 17A, 17B, and 17C which represent a single circuit with net sharing. FIG. 17A shows the implementation of the equations for the complementation of the first six individual bits (a, b, c, d, e, f) to restore the original values (A, B, C, D, E, F). FIG. 17B shows the decoding of the individual trailing three bits (g, h, i) to restore the original values (G, H, I) and the generation of the control bit K. The validity checks are shown at the bottom.

b) Full Vector (bit ‘a’ through ‘i’) Complementation Circuit

The circuit which controls the complementation of entire 9-bit vectors at the top of the diagram of FIG. 17C generates the signal PBM4cn4tn6tn which complements at the lower level the entire vector to recover the primary version. The signal PBM4cn4tn6tn represents the 116 vectors of FIG. 10. The OAI21 gate, which is the negative polarity version of a circuit commonly referred in its positive version as AOI21, is counted as single logic level because its typical delay and area is comparable to a NAND3 or a XNOR2 gate.

10B9B Bit Decoding ABCDEF

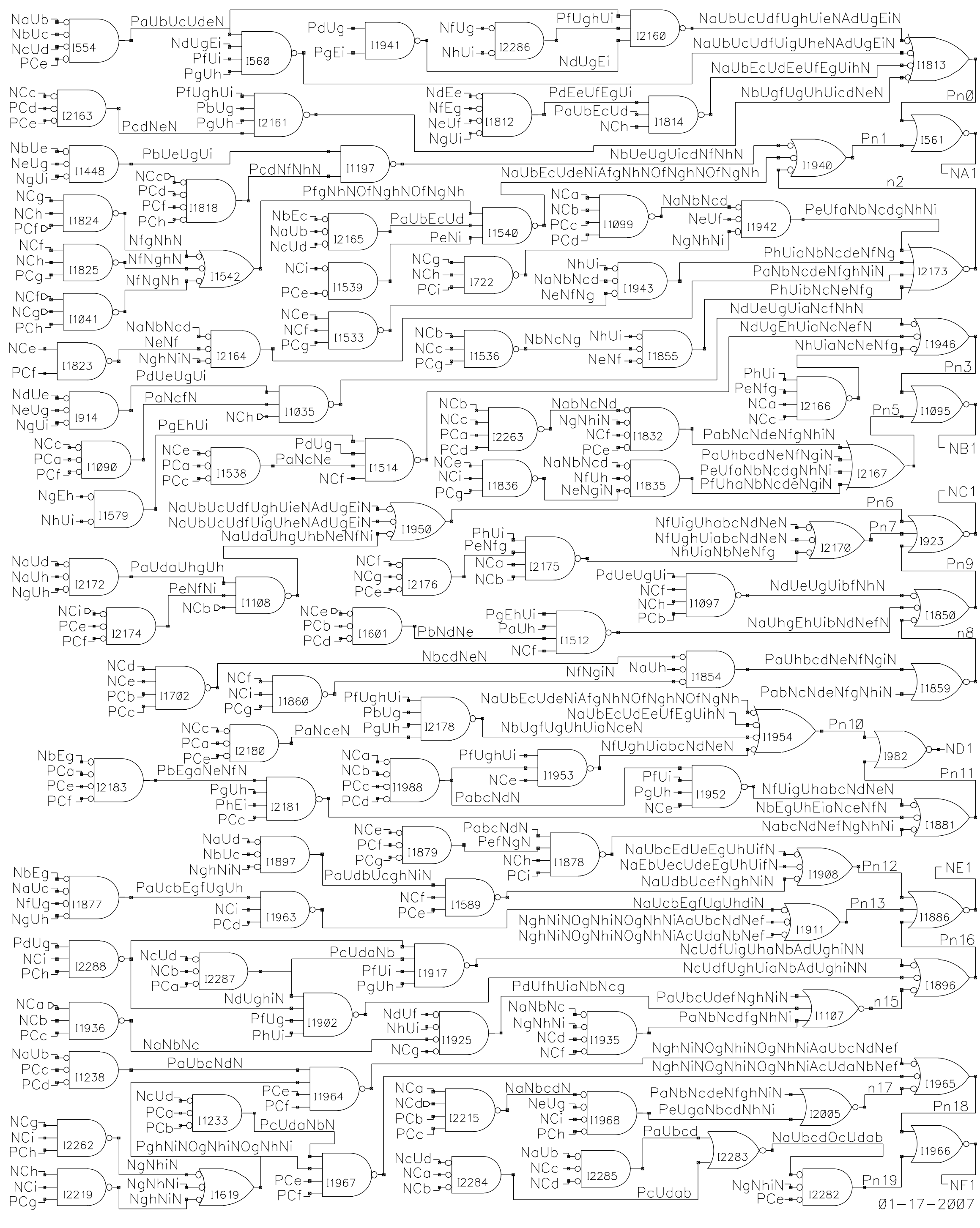

FIG. 17A

10B9B Bit Decoding GHIK, Validity

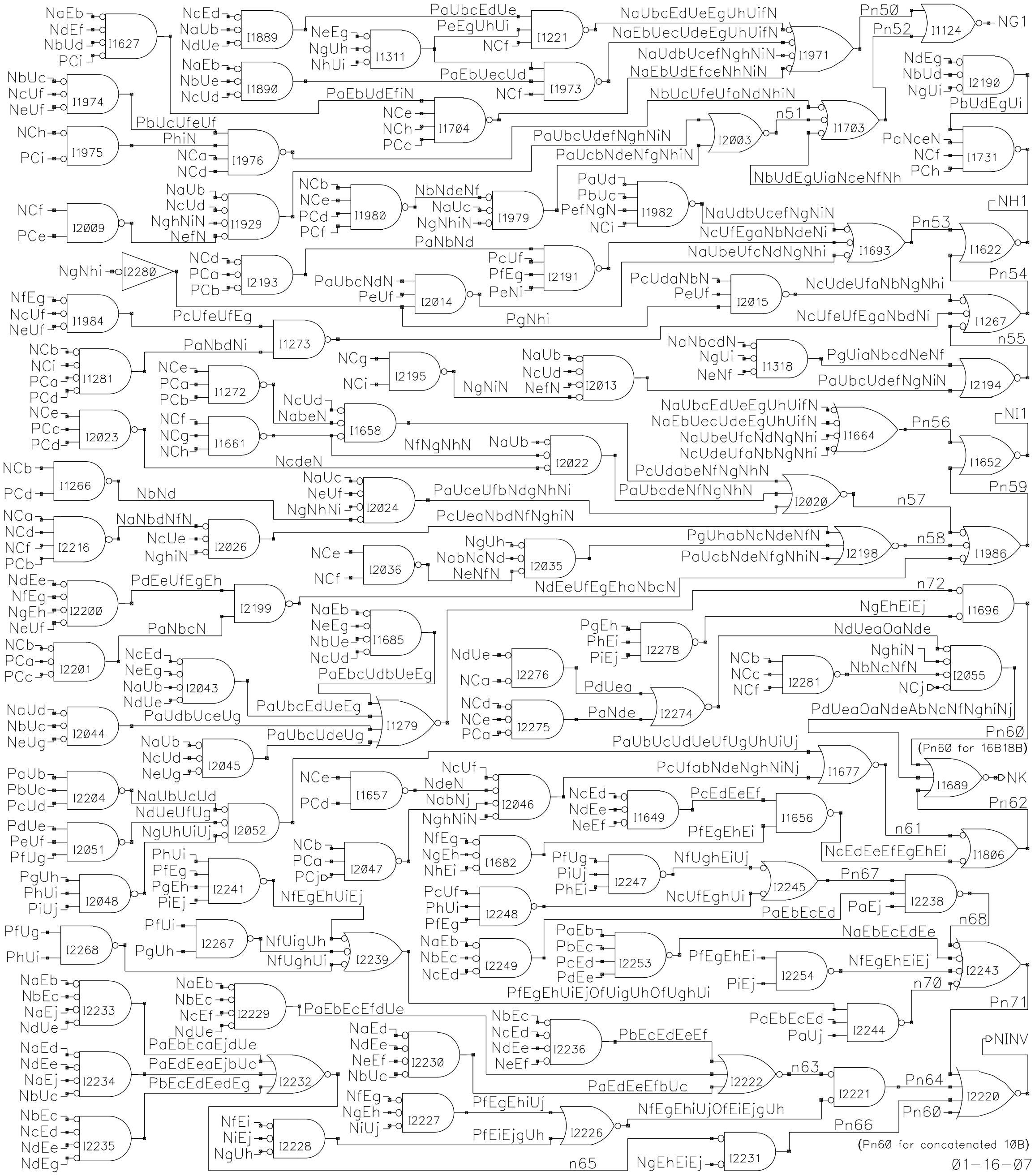

FIG. 17B

c) Error Monitoring Circuits

At the bottom of the diagram 17B is the validity check. A specific application may hold unused control vectors in reserve or declare them invalid at the circuit level. The control vectors represented by the signal Pn60 are invalid for concatenated 9B10B vectors and are then not part of the NK output but are added to the NINV output as shown. A disparity monitoring circuit has not been implemented because bus applications may not use it and for other applications, the detection of disparity errors may be allowed to take two cycles.

The circuits are less time sensitive and can be generated automatically from the equations by design tools.

10B9B Complementation, XOR Functions

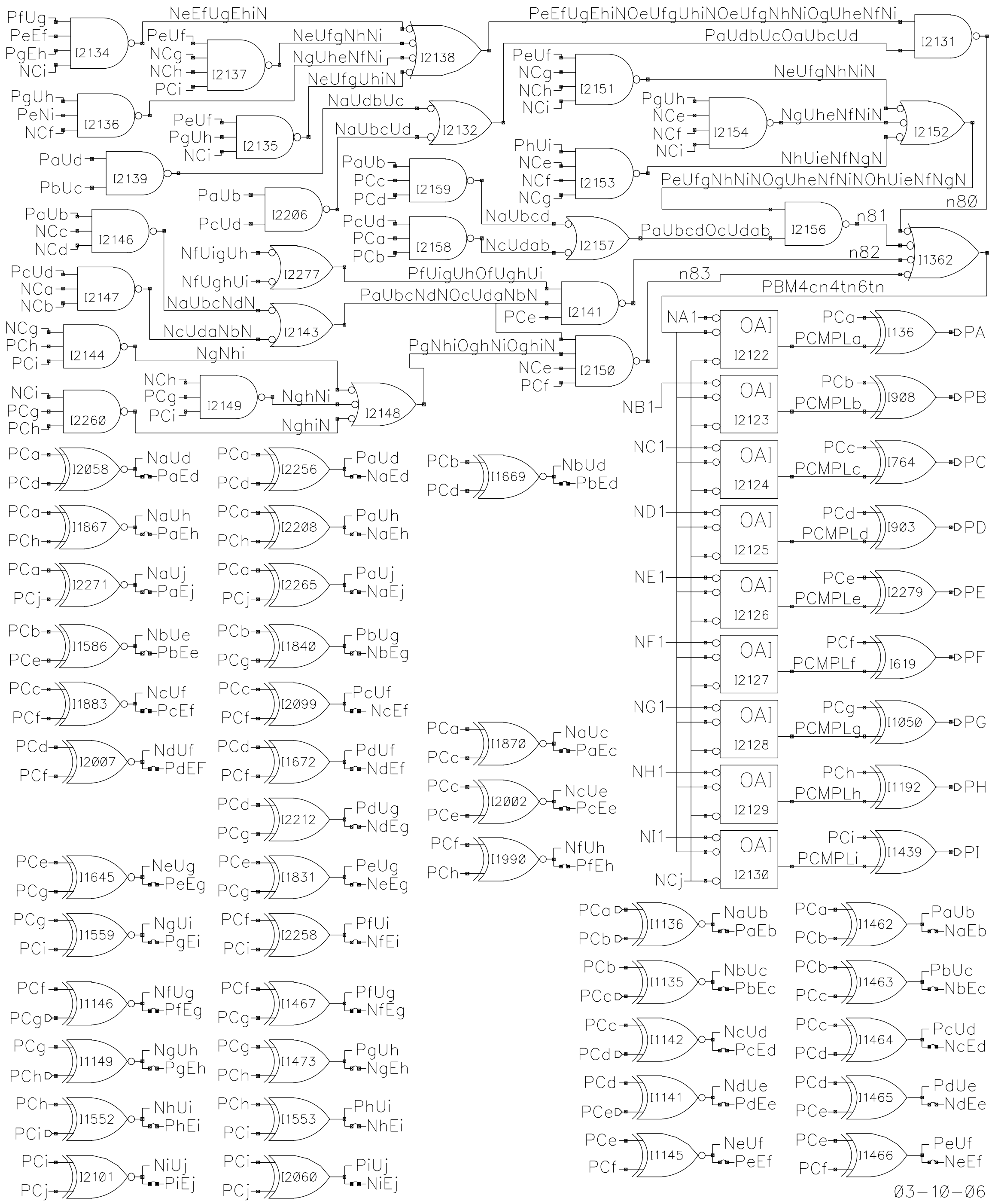

FIG. 17C

The shared EXCLUSIVE OR functions of all 3 diagrams are shown in FIG. 17C. Again, inverters can be substituted for some of these gates depending on speed requirements.

3) Gate Count, Circuit Delays and Pipe-Lining for Decoding

The decoder as shown without disparity monitoring comprises 298 gates, all of the inverting type except some XOR gates. No logic path exceeds seven levels. The path for NK and for PINV is 5 and 6 logic levels, respectively.

For fast operation, pipe-lining can be used analogous to the steps described above for the encoder. The fan-in to the third last gate of the NOR type in the bit decoding cones has been minimized at the cost of a few gates to reduce the number of latches required for pipe-lining at this point. Some of the 2-way and 3-way OR functions have been moved forward and merged with OR functions at the 4th level back from the end. This requires the duplication of some AND functions. It was learned that the circuit penalty is less than apparent, because a uniform design approach results in more matching signal polarities which enables more gate sharing. Similar modifications could be made to the encoding circuit if required.

V. CONCLUSION

The encoder and decoder circuits for the 9B10B code require seven logic levels and can operate at a rate comparable to the best implementations of the well known and widely used partitioned 8B10B code. The number of required gates is far lower than one would expect. Normalized to the number of bits encoded it is just about 2.4 times the number required for 8B10B code. The 16B18B code which uses the 9B10B and the 7B8B code requires just twice as many gates as two 8BB10B encoders and decoders operating in parallel, not counting the disparity monitoring circuits at the receiver.

Many applications are equipped with receivers incorporating Decision Feedback Equalization (DFE). This code supports the rapid recovery of DFE circuits from an error because strings of alternating ones and zeros are limited to less than two vectors in length.

The 9B10B code can be used as stand alone code or as a component of the 16B18B code of Ref. 1. It is also compatible with the 8B10B code and its 5B6B and 3B4B components. A particular attractive applications of the full code or the components is for very high speed busses to save lines in combination with Ref. 5 which shows how to avoid an increase in the line baud rate due to coding and how to eliminate clock gear boxes and extra clock domains by adding extra lines to compensate for the loss of throughput resulting from the code redundancy.

The tables, equations and circuits have only been manually checked, and no programmed computer checks have been run so far. So the presence of minor errors is likely. However, the basic coding principles are sound and detail errors can be corrected by engineers with ordinary skills using the background and methods described in this report.

References

1. WIDMER, A. X., ‘Partitioned dc Balanced (0,6) 16B/18B Transmission Code with Error Correction’, US Patent 6,198,413 B1, Mar. 6, 2001
2. WIDMER, A. X., ‘dc Balanced 7B/8B, 9B/10B, and Partitioned dc Balanced 12B/14B, 17B/20B, and 16B/18B Transmission Codes’, US Patent 6,614,369 B1, Sept. 2, 2003
3. A. X. WIDMER, ‘The ANSI Fibre Channel Transmission Code’, IBM Research Report RC 18855, 4/23/93
4. FRANASZEK, P. A. and WIDMER, A. X., ‘Byte Oriented dc balanced (0,4) 8B/10B Transmission Code’, US Patent 4,486,739, Dec. 4, 1984
5. WIDMER, A. X., ‘Transformation of Parallel Interface into Coded Format with Preservation of Baud-Rate’. US Patent 6,496,540 B1, Dec. 17, 2002
6. WIDMER, A. X., ‘Error Correction with low Latency for Bus Structures’. US Patent 6,978,416 B2, Dec. 20, 2005
7. WIDMER, A. X., ‘dc-Balanced 7B8B Transmission Line Code’, IBM Research Report No. RC 24173, March 16. 2007
8. WIDMER, A. X., ‘Circuit Implementation of a dc-balanced 8B10B-P Transmission Code with local Parity’, IBM Research Report No. RC23924, March 29, 2006
9. SHARLAND, A. J. and STEVENSON, A., ‘A simple in-service error detection scheme based on the statistical properties of line codes for optical fibre Systems’, INT. J. ELECTRONICS, 1983, VOL. 55, NO. 1, 3-33.
10. WIDMER, A. X., ‘Error Correction with Low Latency for Bus Structures’, US Patent 6,978,416 B2, Dec. 20, 2005.
11. WIDMER, A. X., ‘8B/10B Encoding and Decoding for High Speed Applications’, US Patent 6,977,599 B2, Dec. 20, 2005.

Reference File Locations:

Full Report (Frame Maker): /homes/axw/widmer/doc/coding/Code9B10B-RC
Circuit Diagrams (Cadence cteCds): define ether/homes/axw/widmer/artist/serdesg
→ ether → s910encode, s109decode