**RC23924 (W0603-175) March 29, 2006**
**Electrical Engineering**

# IBM Research Report

## Circuit Implementation of a dc-balanced 8B10B-P Transmission Code with Local Parity

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

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

# Circuit Implementation of a dc-balanced 8B10B-P Transmission Code with local Parity

**Albert X. Widmer**

*IBM T.J. Watson Research Center*
*Communication Technology*
*1101 Kitchawan Rd. Route 134, Yorktown Heights, NY 10598-0218*
*914-945-2047, widmer@us.ibm.com*

## Abstract

This report describes a hardware implementation using combinational logic for the encoding and decoding circuits and the validity check of the dc-Balanced 8B10B-P Transmission Code with Local Parity described in US Patent 5,699,062. Less than 300 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

The 8B10B-P Code serves similar applications as the 6B8B-P Code described in Ref. 3. The higher circuit complexity with increased latency delivers better coding efficiency and compatibility with the 8-bit byte format. Other error correction techniques based on the standard 8B10B codes with no local parity of Ref. 4 and 5 have been described in Ref. 1, 6, and 7. They require additional redundancy to precisely locate the error. The 8B10B-P Code uses balanced vectors and vectors with a disparity of plus or minus four as originally suggested by Martin in Ref. 8 and Widmer in Ref. 8 to generate an invalid vector on the occurrence of any odd number of bit errors within a byte.

Reference 1 and 2 do not assign any source vectors to specific encoded vectors. This could be done arbitrarily and a brute force implementation for the encoding and decoding circuits could be accomplished by table look-up. This report shows a specific vector assignment with a corresponding implementation of an encoding and a decoding circuit using only combinational logic.

## 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

### 1) Names

Please note that the capital “B” in 8B10B 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 nine to accommodate control characters, and the number 8 refers to the data vectors only.

The bits of the uncoded 8B data vectors are labelled with the upper case letters ‘ABCDEFGH’ 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’.

### 2) Trellis Diagrams

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 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:

- 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 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.

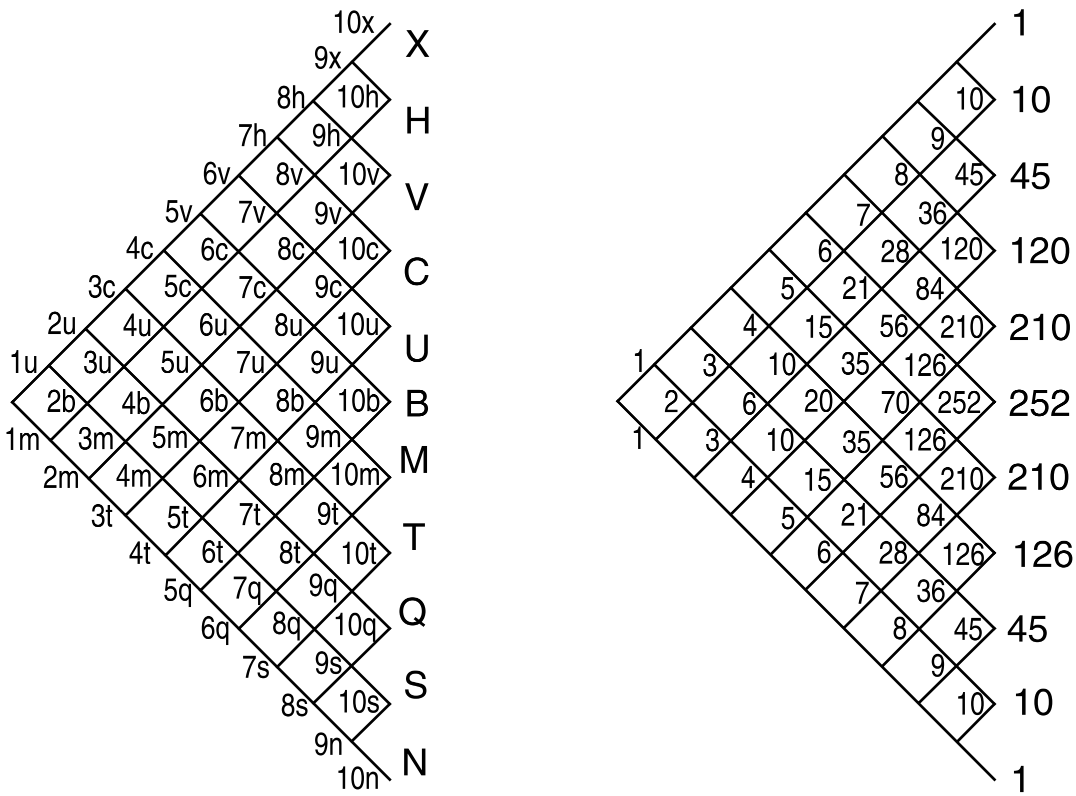


**FIG. 1. Trellis Node Notation and Number of Vectors to Nodes for up to 10 Bits**

### 3) Vector Classification

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

- The first capital letter B, P, 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.

  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 ending coordinate 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 two coded bits *i* and *j* are appended with a default value zero and complemented, if indicated by a classification name ending in *I* and/or *J*, respectively.

## II. DESCRIPTION OF 8B10B-P CODE

### A) Disparity Rules

At all 10B boundaries, the running disparity can assume one of two values D=±2. Encoded vectors in this code are either balanced and disparity independent, *balanced and disparity dependent,* or have a disparity of ±4. If the current running disparity at a byte boundary is positive (+2), only disparity independent vectors or vectors with a required positive entry disparity may be entered and complementary rules apply for a negative running disparity. About two thirds of the source vectors are translated into a single, balanced, disparity independent, encoded vector. All other 8B vectors are translated into one of a pair of complementary 10B vectors, according to the disparity rules above.

Serial transmission of the coded vectors is assumed to be in alphabetical order starting with bit ‘a’.

### B) 8B10B-P Coded Vectors

The 8B10B-P code comprises a total of 263 source vectors each translated into one of 352 coded 10B vectors illustrated by the trellis diagrams of FIGS. 2 to 14. All the other 672 10B vectors are invalid. The use and interpretation of trellis diagrams for this kind of application is explained in Ref. 2. 174 source vectors are encoded into balanced, disparity independent vectors, 29 source vectors are encoded into balanced vectors which are disparity dependent and have complementary representations, the remaining 60 source vectors are each encoded into one of a complementary pair of vectors with a disparity of 4.

The code described in Ref. 1 and 2 includes only four control characters. Because some potential applications have a preference for more control characters, three extra control characters have been added which can generate contiguous runs of five. It is the users choice to include or exclude those characters. If the use of the new control characters is carefully planned, contiguous runs of five can still be avoided. The new control characters are represented by dotted lines in the trellis diagrams below and printed in italic font in the tables.

#### 1) 174 Balanced, Disparity Independent 10B Vectors (FIGS. 2 and 3)

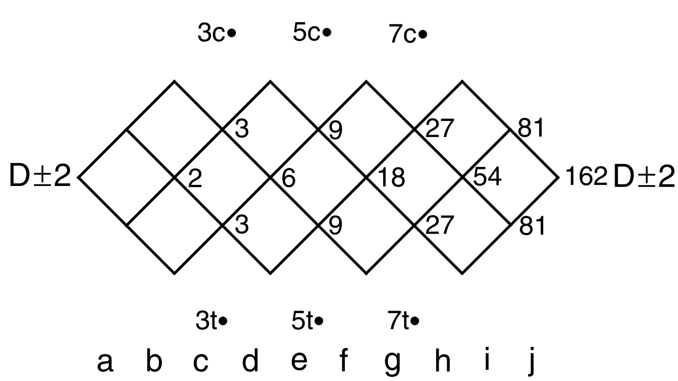



**FIG. 2**

There are 174 disparity independent balanced vectors as illustrated in FIGS. 2 and 3. Balance means that the running disparities at the start and end of the vector are identical. Disparity independence means that they can be entered in a sequence of vectors regardless of the current starting disparity which can have a value of plus two or minus two at the vector boundaries.

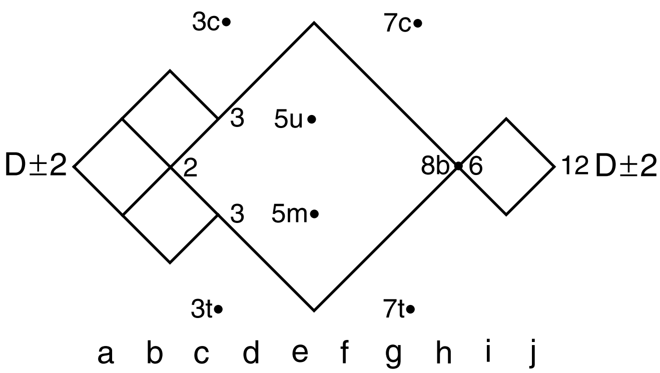


**FIG. 3**

The 174 disparity independent vectors include all balanced 10B vectors with a run length of no more than two at the leading and trailing boundaries with the exception of the three disparity dependent vectors of FIG. 4 and their complements with a trailing run of two followed by a run of one and another run of two. Using the notation described above, the trellises of FIG. 2 and 3 can be described by the expressions B3c'5c'7c'3t'5t'7t' and B3c'7c'3t'7t'8b, respectively. The latter expression includes some vectors of FIG. 2 redundantly. They can be excluded by the addition of the term 5u'5m'.

**2) 2x29 Balanced, Disparity Dependent 10B Vectors (FIGS. 4 to 6)**

FIGS. 4 to 6 show balanced trellises with a required negative starting disparity. For a positive running disparity, their complements must be used.

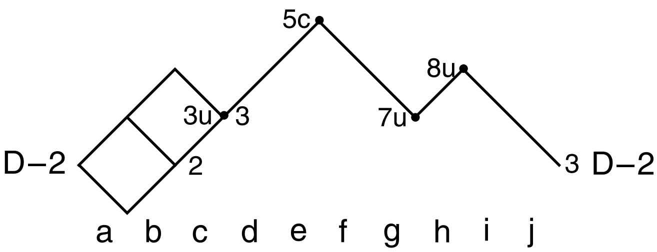


**FIG. 4**

The three vectors of FIG. 4 can be described by the expression P3u5c7u8u and their complements by P3m5t7m8m.

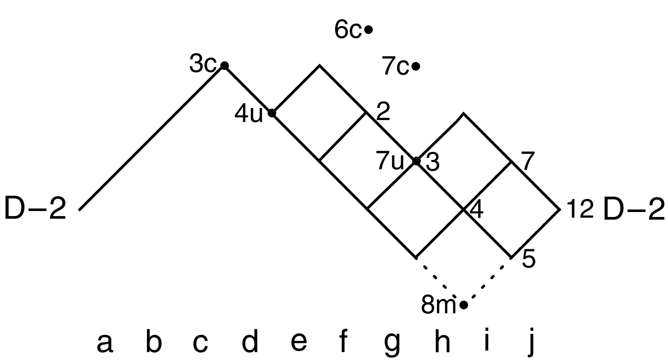


**FIG. 5**

The eleven vectors of FIG. 5 in solid lines can be described by the expression P3c4u6c'7c'8m'. The vector P3c8m through the node 8m is assigned to the alternate version K248A of an optional control character K248P (P3t8u). This polarity selection simplifies the equations for 10-bit vector complementation.

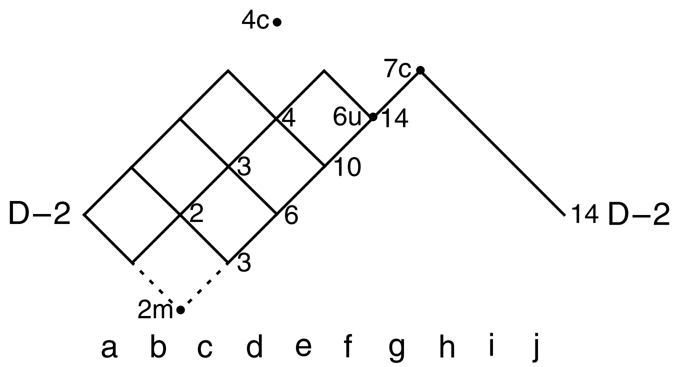



FIG. 6

The fourteen balanced vectors of FIG. 6 can be described by the expression P4c'6u7c. The vector through the node 2m is assigned to an optional primary control character K124P.

**3) 2x60 10B Vectors with a Disparity of Four (FIGS. 7 to 13)**

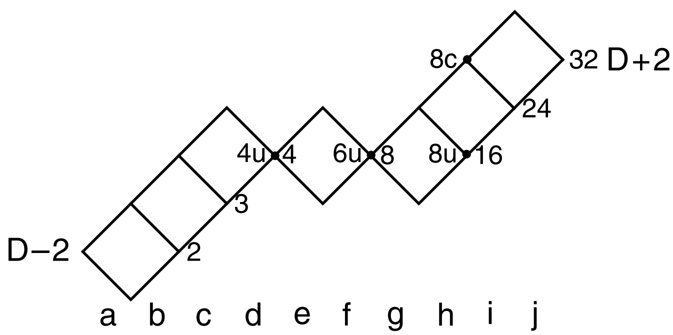



FIG. 7

The thirty-two vectors of FIG. 7 with a disparity of plus four and a negative required entry disparity can be described by the expression F4u6u. Its complement is described by the expression F4m6m.

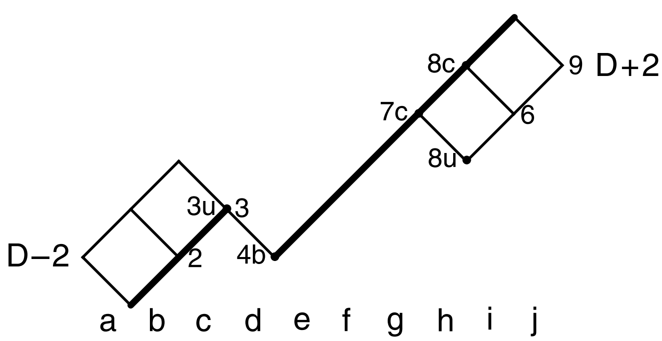



FIG. 8

The nine vectors of FIG. 8 with a disparity of plus four and a negative required entry disparity can be described by the expression F3u4b7c. Its complement is described by the expression F3m4b7t. The two fat lines represent the singular comma sequence.

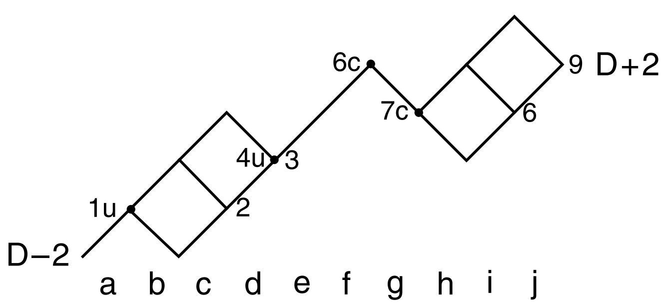



FIG. 9

The nine vectors of FIG. 9 with a disparity of plus four and a negative required entry disparity can be described by the expression F1u4u6c7c. Its complement is described by the expression F1m4m6t7t.

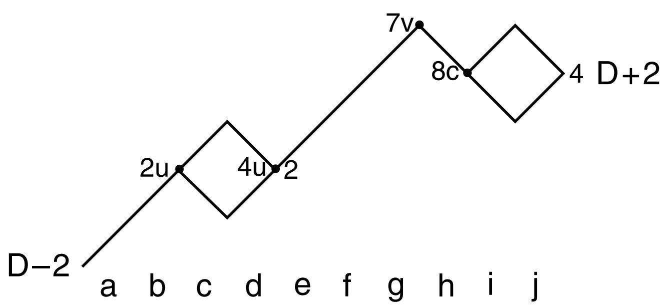


**FIG. 10**

The four vectors of FIG. 10 with a disparity of plus four and a negative required entry disparity can be described by the expression F2u4u7v8c. Its complement is described by the expression F2m4m7q8t.

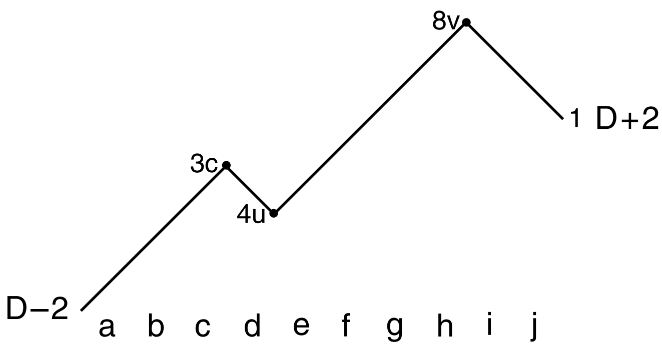


**FIG. 11**

The single vector of FIG. 11 with a disparity of plus four and a negative required entry disparity can be described by the expression F3c4u8v. Its complement is described by the expression F3t4m8q.

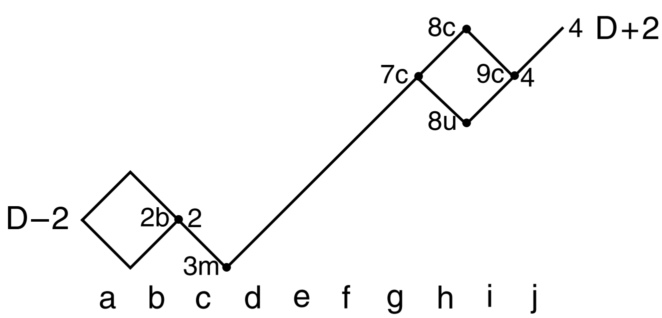


**FIG. 12**

The four vectors of FIG. 12 with a disparity of plus four and a negative required entry disparity can be described by the expression F2b3m7c9c. Its complement is described by the expression F2b3u7t9t.

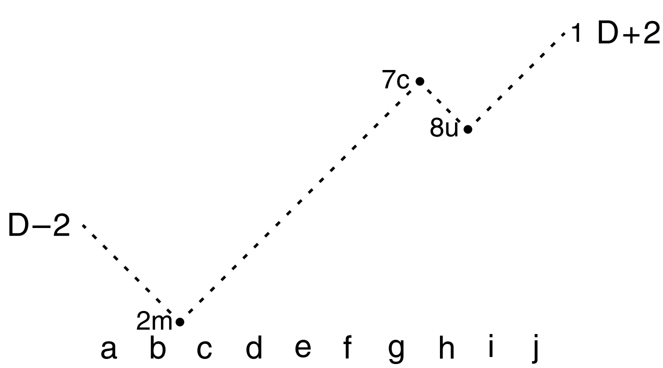


**FIG. 13**

The single vector of FIG. 13 with a disparity of plus four and a negative required entry disparity is the alternate version K131A of one of the optional control characters. It can be described by the expression F2m7c8u. Its complement is described by the expression F2u7t8m.

## C) Code Validation

The trellis diagrams of FIGS. 2 to 13 can be used to prove the validity of the code. They show the total number of available coded vectors. Since none of the vectors of each diagram is congruent with any of the vectors of all the other diagrams, there are no duplicate vectors. The combinations of any trailing and leading runs of true and complement forms with their associated running disparity at the byte boundaries comply with the run length limit of five. Similarly, the singularity of the comma character can be assured by the examination of all possible combinations of trailing and leading bit patterns.

## D) 8B10B-P Source Vector to Coded Vector Assignment, Coding Table

This chapter describes the specific assignment of source vectors to coded vectors. The execution of this step materially affects the complexity of the implementation. Preference is given to coding assignments which preserve the values of the source bits as is the case for all vectors listed in Tables 1 to 8. The Tables 1, 2, and 3 list balanced, disparity independent vectors and the Tables 4 through 8 list the primary vectors of all the disparity dependent vectors. Disparity dependent vectors have two complementary representations which are referred to as the primary (P) and the alternate (A) vector. The coding assignments have been chosen such that all disparity dependent primary vectors end with ij=00 or 01 to simplify the decoding process. The important aspect is not the specific ending, but the restriction to just two ending patterns. All source vectors which require individual bit changes for encoding are listed in Table 9. The 60 coded vectors of Table 9 can be identified as disparity independent, balanced vectors ending with ij=01.

FIGS. 14 and 15 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 8B10B-P Encoding**

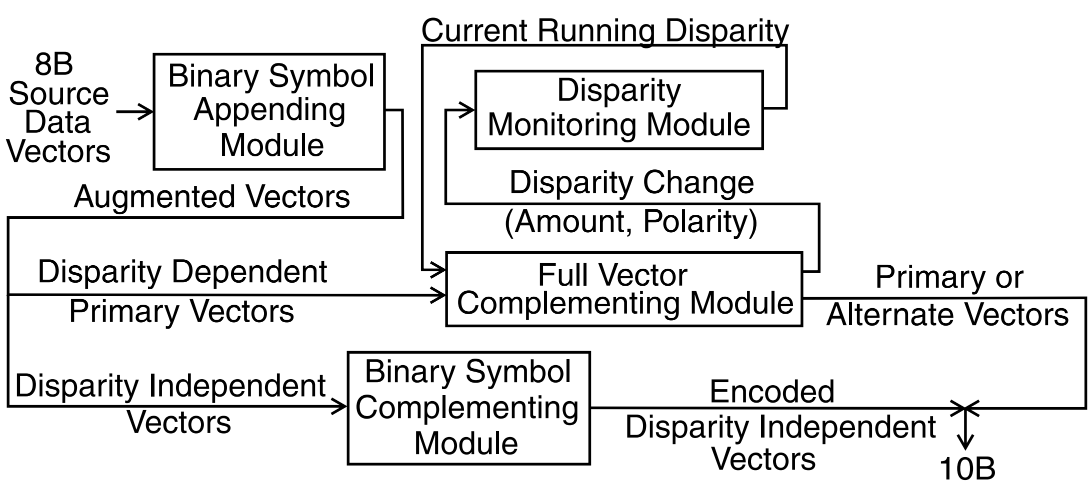



**FIG. 14**

**Conceptual View of 10B8B-P Decoding**

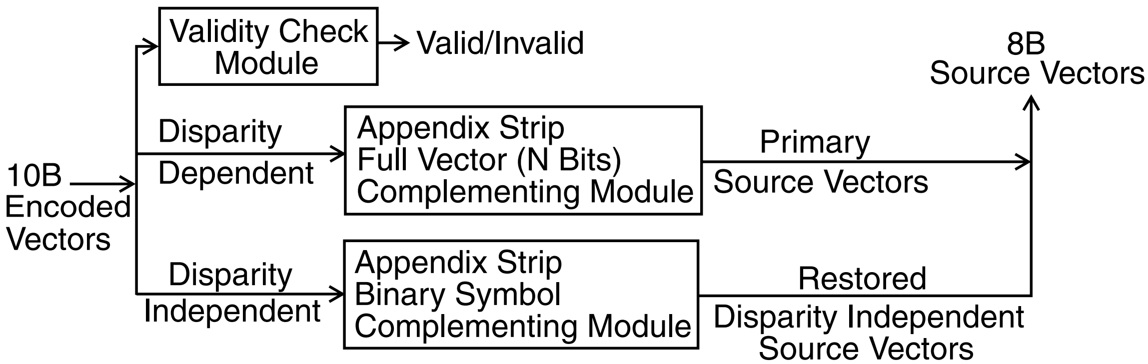



**FIG. 15**

The implementation problems to be addressed for the encoder and decoder are circuit area and delay reduction which has been solved primarily by a coding table based on the following principles:

1. All vectors with individual bit changes are relegated to a single class of vectors which is balanced and disparity independent. This makes it possible to execute individual bit changes and full vector inversion for disparity control as completely separate functions in parallel rather than serial mode in both the encoder and the decoder circuit with significant less overall circuit delay.
2. Uncoded source vectors are assigned to coded vectors such that the number of vectors with individual bit changes and the number of bit changes in a vector is minimized. In the interest of logic circuit sharing, preference is given to solutions with bit changes concentrated to fewer vectors over solutions with the same number of bit changes spread over more vectors.
3. Both the source vectors and the encoded vectors have been sorted into groups with commonalities which can be defined by simple logic equations and circuits.

The 114 disparity independent balanced vectors listed in Table 1, 2, and 3 are the subset of the vectors of FIGS. 2 and 3 which end with ij = 00, 11, or 10, but not those which end with 01. The 27 vectors of Table 1 can be identified by BU3c’5c’7c’3t’.

**Table 1. Twenty-seven Balanced Disparity independent Vectors BU3c’5c’7c’3t’**

| Name | abcdefghij | Name | abcdefghij | Name | abcdefghij |
|---|---|---|---|---|---|
| D171 | 1101010100 | D203 | 1101001100 | D241 | 1000111100 |
| D173 | 1011010100 | D205 | 1011001100 | D242 | 0100111100 |
| D174 | 0111010100 | D206 | 0111001100 | D244 | 0010111100 |
| D179 | 1100110100 | D211 | 1100101100 | D227 | 1100011100 |
| D181 | 1010110100 | D213 | 1010101100 | D229 | 1010011100 |
| D182 | 0110110100 | D214 | 0110101100 | D230 | 0110011100 |
| D185 | 1001110100 | D217 | 1001101100 | D233 | 1001011100 |
| D186 | 0101110100 | D218 | 0101101100 | D234 | 0101011100 |
| D188 | 0011110100 | D220 | 0011101100 | D236 | 0011011100 |

The 27 vectors of Table 2 can be identified by BM3c'3t'5t'3t'7t'IJ.

**Table 2. Twenty-seven Balanced Disparity independent Vectors BM3c'3t'5t'7t'IJ**

| Name | abcde f gh i j | Name | abcde f gh i j | Name | abcde f gh i j |
|---|---|---|---|---|---|
| D11 | 1101000011 | D49 | 1000110011 | D81 | 1000101011 |
| D13 | 1011000011 | D50 | 0100110011 | D82 | 0100101011 |
| D14 | 0111000011 | D52 | 0010110011 | D84 | 0010101011 |
| D19 | 1100100011 | D35 | 1100010011 | D67 | 1100001011 |
| D21 | 1010100011 | D37 | 1010010011 | D69 | 1010001011 |
| D22 | 0110100011 | D38 | 0110010011 | D70 | 0110001011 |
| D25 | 1001100011 | D41 | 1001010011 | D73 | 1001001011 |
| D26 | 0101100011 | D42 | 0101010011 | D74 | 0101001011 |
| D28 | 0011100011 | D44 | 0011010011 | D76 | 0011001011 |

The 60 vectors of Table 3 can be identified by BB3c'3t'I.

**Table 3. Sixty Balanced Disparity Independent Vectors BB3c'3t'I**

| Name | abcde f gh i j | Name | abcde f gh i j | Name | abcde f gh i j | Name | abcde f gh i j |
|---|---|---|---|---|---|---|---|
| D27 | 1101100010 | D43 | 1101010010 | D75 | 1101001010 | D139 | 1101000110 |
| D29 | 1011100010 | D45 | 1011010010 | D77 | 1011001010 | D141 | 1011000110 |
| D30 | 0111100010 | D46 | 0111010010 | D78 | 0111001010 | D142 | 0111000110 |
| D51 | 1100110010 | D99 | 1100011010 | D163 | 1100010110 | | |
| D53 | 1010110010 | D101 | 1010011010 | D165 | 1010010110 | | |
| D54 | 0110110010 | D102 | 0110011010 | D166 | 0110010110 | | |
| D57 | 1001110010 | D105 | 1001011010 | D169 | 1001010110 | | |
| D58 | 0101110010 | D106 | 0101011010 | D170 | 0101010110 | | |
| D60 | 0011110010 | D108 | 0011011010 | D172 | 0011010110 | | |
| D83 | 1100101010 | D147 | 1100100110 | D195 | 1100001110 | | |
| D85 | 1010101010 | D149 | 1010100110 | D197 | 1010001110 | | |
| D86 | 0110101010 | D150 | 0110100110 | D198 | 0110001110 | | |
| D89 | 1001101010 | D153 | 1001100110 | D201 | 1001001110 | | |
| D90 | 0101101010 | D154 | 0101100110 | D202 | 0101001110 | | |
| D92 | 0011101010 | D156 | 0011100110 | D204 | 0011001110 | | |
| D113 | 1000111010 | D177 | 1000110110 | D209 | 1000101110 | D225 | 1000011110 |
| D114 | 0100111010 | D178 | 0100110110 | D210 | 0100101110 | D226 | 0100011110 |
| D116 | 0010111010 | D180 | 0010110110 | D212 | 0010101110 | D228 | 0010011110 |

The 24 balanced vectors listed in Table 4 preserve the values of the source bits and are the subset of the vectors of FIGS.4 to 6 which do not end with ij=10 or 11. They all require a negative entry disparity. The optional control character K124P is illustrated in FIG.6.

**Table 4. Twenty-four Balanced Disparity Dependent Vectors**

| Name | abcdefghij | Coding Class | Name | abcdefghij | Coding Class |
|---|---|---|---|---|---|
| D155P | 1101100100 | PU3u5c7u | D87P | 1110101000 | PU2m'4c'6u7c |
| D157P | 1011100100 | PU3u5c7u | D91P | 1101101000 | PU2m'4c'6u7c |
| D158P | 0111100100 | PU3u5c7u | D93P | 1011101000 | PU2m'4c'6u7c |
| D151P | 1110100100 | PU3c4u7u | D94P | 0111101000 | PU2m'4c'6u7c |
| D167P | 1110010100 | PU3c4u7u | D103P | 1110011000 | PU2m'4c'6u7c |
| D199P | 1110001100 | PU3c4u7u | D107P | 1101011000 | PU2m'4c'6u7c |
| D23P | 111010000**1** | PB3c4uJ | D109P | 1011011000 | PU2m'4c'6u7c |
| D39P | 111001000**1** | PB3c4uJ | D110P | 0111011000 | PU2m'4c'6u7c |
| D71P | 111000100**1** | PB3c4uJ | D115P | 1100111000 | PU2m'4c'6u7c |
| D135P | 111000010**1** | PB3c4uJ | D117P | 1010111000 | PU2m'4c'6u7c |
| *K124P* | *0011111000* | *PUK2m7c* | D118P | 0110111000 | PU2m'4c'6u7c |
| | | | D121P | 1001111000 | PU2m'4c'6u7c |
| | | | D122P | 0101111000 | PU2m'4c'6u7c |

The five balanced vectors of FIG. 16 and Table 5 require a positive entry disparity. They are the complements of those five vectors of FIG. 5 which end with ij=10 or 11.

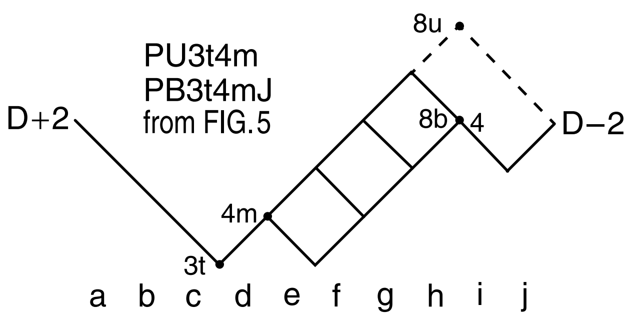



**FIG. 16**

**Table 5. Five Balanced Disparity Dependent Vectors**

| Name | abcdefghij | Coding Class |
|---|---|---|
| D120P | 000111100**1** | PB3t4mJ |
| D184P | 000111010**1** | PB3t4mJ |
| D216P | 000110110**1** | PB3t4mJ |
| D232P | 000101110**1** | PB3t4mJ |
| *K248P* | *0001111100* | *PUK3t4m* |

The complement is chosen so the source vectors translate with no changes into primary vectors and so all encoded vectors with changes in the source bits end with ij=01.

The nineteen vectors of Table 6 have disparity of plus four and the value of the source bits is preserved. They all require a negative entry disparity. One vector (D247P) ending with ij = 00 is illustrated in FIG. 11 and the others are a subset of the vectors of FIGS. 7 through 10, and 12 and end with ij = 01.

Table 6. **Nineteen Vectors with a Disparity of plus Four**

| Name | abcdefghij | Coding Class | Name | abcdefghij | Coding Class |
|---|---|---|---|---|---|
| D215P | 1110101101 | FC4u6uJ | D249P | 1001111101 | FC2b3mJ |
| D219P | 1101101101 | FC4u6uJ | D250P | 0101111101 | FC2b3mJ |
| D221P | 1011101101 | FC4u6uJ | D243P | 1100111101 | FC3u4bJ |
| D222P | 0111101101 | FC4u6uJ | D245P | 1010111101 | FC3u4bJ |
| D231P | 1110011101 | FC4u6uJ | D246P | 0110111101 | FC3u4bJ |
| D235P | 1101011101 | FC4u6uJ | D183P | 1110110101 | FC1u4u6c7cJ |
| D237P | 1011011101 | FC4u6uJ | D187P | 1101110101 | FC1u4u6c7cJ |
| D238P | 0111011101 | FC4u6uJ | D189P | 1011110101 | FC1u4u6c7cJ |
| | | | D119P | 1110111001 | FC2u4u7vJ |
| D247P | 1110111100 | FV3c4u | D123P | 1101111001 | FC2u4u7vJ |

The twenty-five vectors of Table 7 have a disparity of minus four and the value of the source bits is preserved. They are the complements of the vector of FIG. 13 or of a subset of vectors of FIGS. 7, 8, 9, and 12 and end with ij = 00.

FIG. 17 is the trellis of the top 16 vectors in Table 7 which are a complemented subset of the vectors of FIG. 7.

Table 7. **Twenty-five Vectors with a Disparity of minus Four**

| Name | abcdefghij | Coding Class | Name | abcdefghij | Coding Class |
|---|---|---|---|---|---|
| K81P | 1000101000 | FMK3m4m5m6m7m | D97P | 1000011000 | FM4m5t6m |
| K82P | 0100101000 | FMK3m4m5m6m7m | D98P | 0100011000 | FM4m5t6m |
| K84P | 0010101000 | FMK3m4m5m6m7m | D100P | 0010011000 | FM4m5t6m |
| D88P | 0001101000 | FM3t5m6m7m | D104P | 0001011000 | FM4m5t6m |
| D145P | 1000100100 | FM4m5m7t | D161P | 1000010100 | FM4m5t6m |
| D146P | 0100100100 | FM4m5m7t | D162P | 0100010100 | FM4m5t6m |
| D148P | 0010100100 | FM4m5m7t | D164P | 0010010100 | FM4m5t6m |
| D152P | 0001100100 | FM4m5m7t | D168P | 0001010100 | FM4m5t6m |
| D137P | 1001000100 | FM3m4b7t | D194P | 0100001100 | FM1m4m6t |
| D138P | 0101000100 | FM3m4b7t | D196P | 0010001100 | FM1m4m6t |
| D140P | 0011000100 | FM3m4b7t | D200P | 0001001100 | FM1m4m6t |
| D133P | 1010000100 | FM2b3u7t | *K131P* | *1100000100* | *FMK2u7t* |
| D134P | 0110000100 | FM2b3u7t | | | |

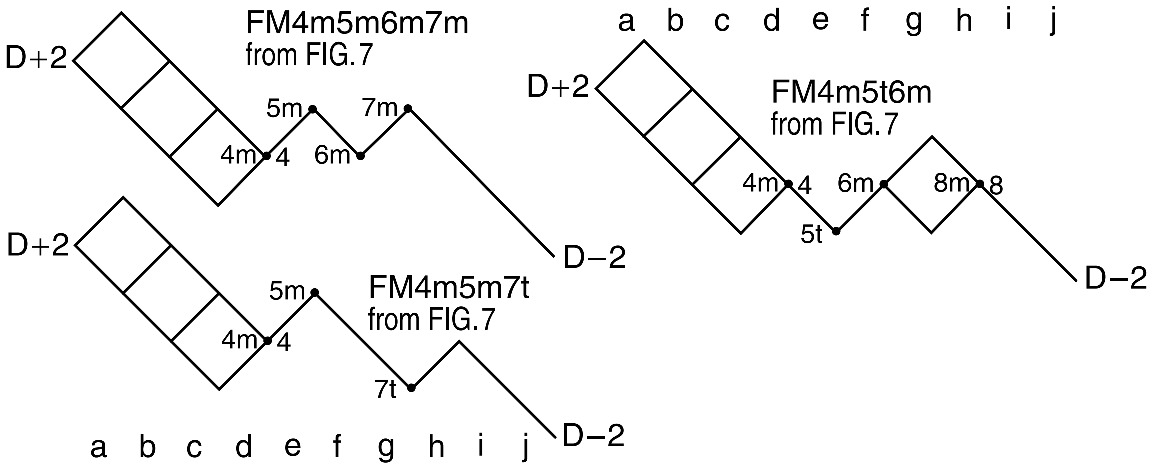



**FIG. 17**

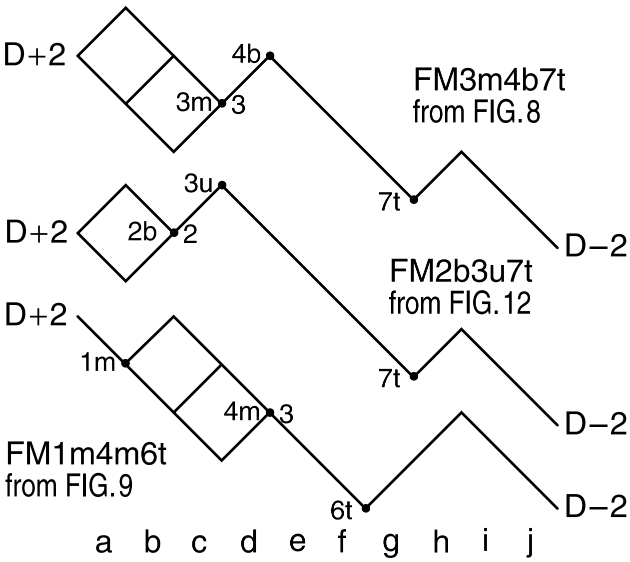



**FIG. 18**

FIG. 18 is the trellis of the bottom data vector sets which are complemented subsets of the vectors of FIGS. 8, 9, and 12, respectively. They all have a disparity of minus four and the value of the source bits is preserved. K131 is is the complement of the vector of FIG. 13.

The sixteen vectors of Table 8 have disparity of minus four and the value of the source bits is preserved. They are the complements of a subset of the vectors of FIGS. 7, 8, 9, and 10 and end with ij=01 and are illustrated with their trellises in FIG. 19.

**Table 8. Sixteen Vectors with a Disparity of minus Four**

| Name | abcdefghij | Coding Class | Name | abcdefghij | Coding Class |
|---|---|---|---|---|---|
| D17P | 1000100001 | FT4m6mJ | C9P | 1001000001 | FTK3m4bJ (Comma) |
| D18P | 0100100001 | FT4m6mJ | D10P | 0101000001 | FTK'3m4bJ |
| D20P | 0010100001 | FT4m6mJ | D12P | 0011000001 | FTK'3m4bJ |
| D24P | 0001100001 | FT4m6mJ | D66P | 0100001001 | FT1m4m6t7tJ |
| D33P | 1000010001 | FT4m6mJ | D68P | 0010001001 | FT1m4m6t7tJ |
| D34P | 0100010001 | FT4m6mJ | D72P | 0001001001 | FT1m4m6t7tJ |
| D36P | 0010010001 | FT4m6mJ | D132P | 0010000101 | FT2m4m7qJ |
| D40P | 0001010001 | FT4m6mJ | D136P | 0001000101 | FT2m4m7qJ |

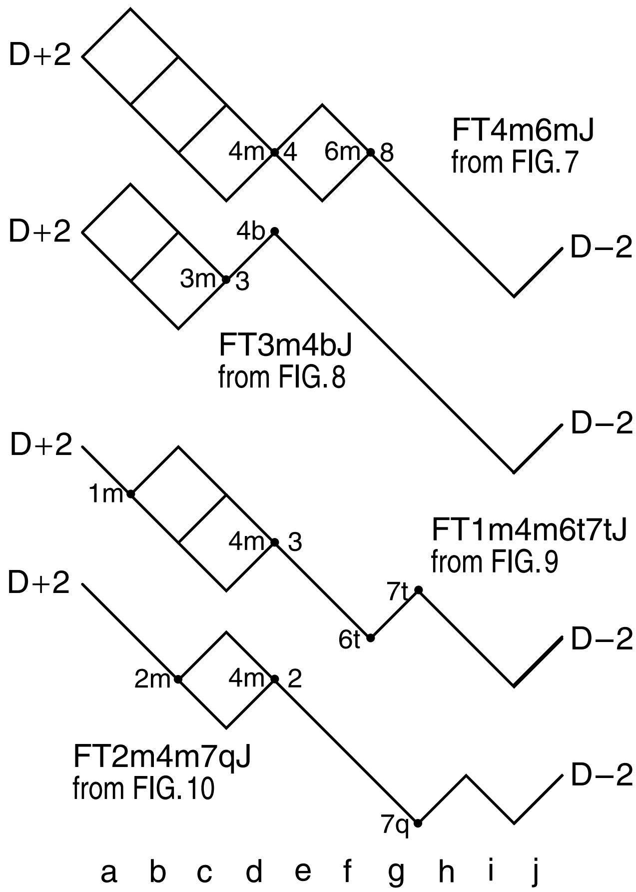



**FIG.19**

Up to this point, 203 source vectors (196 data, 7 control) have been assigned to encoded vectors as listed in the Tables 1 through 8. None of these vectors requires any change in the source bits for encoding and decoding. There remain 60 unassigned data source vectors and 60 available encoded vectors, 54 from FIG.2 and 6 from FIG.3, all balanced and disparity independent ending with ij=01. The trellis diagrams of some of the unassigned 8-bit source vectors are illustrated in FIG.20. The fat lanes indicate the bits which are complemented for encoding. Wherever possible, complementary source vector pairs are assigned to a pair of encoded vectors which are also complements and the individual bit positions complemented for encoding are identical for both coded vectors of a pair. Also, groups of several vector pairs with identical encoding rules are defined as shown for three sets of four pairs in FIG.20.

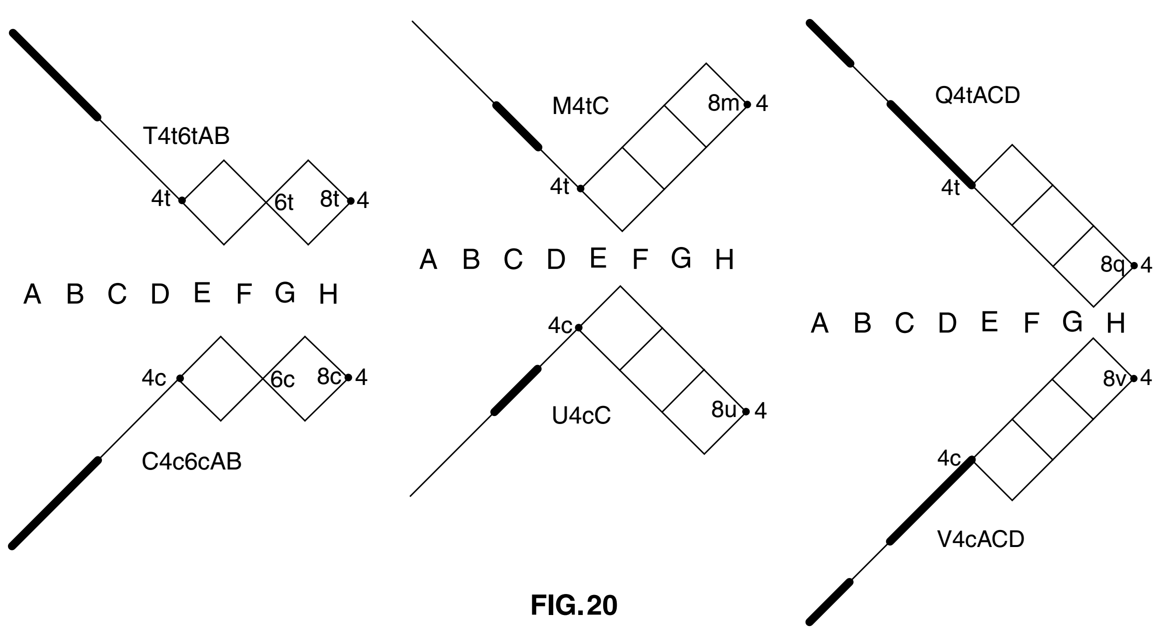



**FIG.20**

The complete set of vectors with individual source bit changes is listed in Table 9. The appended bits i and j have an assumed default value of zero. The coded bits which are different from the source bits or the default value are printed in bold type and the vectors on the right side of the table are the complements of the left side. This arrangement contributes to circuit simplification.

**Table 9. Sixty Balanced, Disparity Independent Encoded Vectors Ending with ij=01**

| Name | ABCDEFGH | Coding Class | abcdefghij | Name | ABCDEFGH | Coding Class | abcdefghij |
|---|---|---|---|---|---|---|---|
| D80 | 00001010 | BT4tABJ | 1100101001 | D175 | 11110101 | BC4cABJ | 0011010101 |
| D96 | 00000110 | BT4tABJ | 1100011001 | D159 | 11111001 | BC4cABJ | 0011100101 |
| D144 | 00001001 | BT4tABJ | 1100100101 | D111 | 11110110 | BC4cABJ | 0011011001 |
| D160 | 00000101 | BT4tABJ | 1100010101 | D95 | 11111010 | BC4cABJ | 0011101001 |
| D224 | 00000111 | BM4tCJ | 0010011101 | D31 | 11111000 | BU4cCJ | 1101100001 |
| D208 | 00001011 | BM4tCJ | 0010101101 | D47 | 11110100 | BU4cCJ | 1101010001 |
| D176 | 00001101 | BM4tCJ | 0010110101 | D79 | 11110010 | BU4cCJ | 1101001001 |
| D112 | 00001110 | BM4tCJ | 0010111001 | D143 | 11110001 | BU4cCJ | 1101000101 |
| D128 | 00000001 | BQ4tACDJ | 1011000101 | D127 | 11111110 | BV4cACDJ | 0100111001 |
| D64 | 00000010 | BQ4tACDJ | 1011001001 | D191 | 11111101 | BV4cACDJ | 0100110101 |
| D32 | 00000100 | BQ4tACDJ | 1011010001 | D223 | 11111011 | BV4cACDJ | 0100101101 |
| D16 | 00001000 | BQ4tACDJ | 1011100001 | D239 | 11110111 | BV4cACDJ | 0100011101 |
| D48 | 00001100 | BT4tADJ | 1001110001 | D207 | 11110011 | BC4cADJ | 0110001101 |
| D192 | 00000011 | BT4tBDJ | 0101001101 | D63 | 11111100 | BC4cBDJ | 1010110001 |
| D4 | 00100000 | BQ2m3mBDFJ | 0111010001 | D251 | 11011111 | BV2u3uBDFJ | 1000101101 |
| D3 | 11000000 | BT2uEFJ | 1100110001 | D252 | 00111111 | BC1m4bEFJ | 0011001101 |
| D130 | 01000001 | BT1m2b7qDEJ | 0101100101 | D125 | 10111110 | BC2b7vDEJ | 1010011001 |
| D129 | 10000001 | BT1u6tDEJ | 1001100101 | D126 | 01111110 | BC2b7vDEJ | 0110011001 |
| D0 | 00000000 | BSBCEGJ | 0110101001 | D255 | 11111111 | BHBCEGJ | 1001010101 |
| D65 | 10000010 | BT1u6tCEJ | 1010101001 | D190 | 01111101 | BC1m6c7cCEJ | 0101010101 |
| D2 | 01000000 | BQ1m2bCDHJ | 0111000101 | D253 | 10111111 | BV2bCDHJ | 1000111001 |
| D1 | 10000000 | BQ1uDFGJ | 1001011001 | D254 | 01111111 | BV2bDFGJ | 0110100101 |
| D131 | 11000001 | BMK'2u7tGJ | 1100001101 | D124 | 00111110 | BUK'2m7cGJ | 0011110001 |
| D15 | 11110000 | BB4cAGJ | 0111001001 | D240 | 00001111 | BB4tAGJ | 1000110101 |
| D193 | 10000011 | BM1u6tFJ | 1000011101 | D62 | 01111100 | BU1m6cFJ | 0111100001 |
| D7 | 11100000 | BM3cBFHJ | 1010010101 | D248 | 00011111 | BUK'3tBFHJ | 0101101001 |
| | | | | D55 | 11101100 | BU2u4u6cAJ | 0110110001 |
| | | | | D59 | 11011100 | BU2u4u6cAJ | 0101110001 |
| D5 | 10100000 | BT2b3uGHJ | 1010001101 | | | | |
| D9 | 10010000 | BTK'1u3m4bGHJ | 1001001101 | | | | |
| | | | | D61 | 10111100 | BU1u2b6cDFHJ | 1010100101 |
| | | | | D56 | 00011100 | BM3t6bAFGJ | 1001101001 |
| D8 | 00010000 | BQ3t4mBFGJ | 0101011001 | | | | |
| D6 | 01100000 | BT2b3uFHJ | 0110010101 | | | | |

The value of the K-bit is not listed with the source vectors of Table 9. For most vectors, the K-bit is redundant and can assume a value of zero if present. A zero value must be included for the seven data source vectors which are otherwise identical to a control vector. For those data vectors, the letter D in the vector name is printed in bold type.

Some assignment choices, e.g. selecting K248 rather than K7 as a control vector have been made so it is easier to recognize the alternate vectors which must be complemented.

**Summary Table**

Table 10 lists the selected vector assignments of the Tables 1 to 9 in ascending order of vector names. The alternate vectors are also shown. The six control characters are listed at the end of the Table. An ‘x’ entry in the K-column means that the K-bit has a value of zero but can be ignored for encoding. The column headed by ‘Pri DR’ lists the required entry disparity for the primary vector. The column ‘Pri DB’ lists the block disparity of the primary vector.

**Table 10. 8B10B-P Encoding with Local Parity**

| Name Primary | ABCDEFGH K | Coding Class Primary | Primary abcdef ghij | Pri DR | Pri DB | Name Alternate | Alternate abcdef ghij |
|---|---|---|---|---|---|---|---|
| D0 | 00000000 x | BSBCEGJ | 0110101001 | ± | 0 | | |
| D1 | 10000000 x | BQ1uDFGJ | 1001011001 | ± | 0 | | |
| D2 | 01000000 x | BQ1m2bCDHJ | 0111000101 | ± | 0 | | |
| D3 | 11000000 x | BT2uEFJ | 1100110001 | ± | 0 | | |
| D4 | 00100000 x | BQ2m3mBDFJ | 0111010001 | ± | 0 | | |
| D5 | 10100000 x | BT2b3uGHJ | 1010001101 | ± | 0 | | |
| D6 | 01100000 x | BT2b3uFHJ | 0110010101 | ± | 0 | | |
| D7 | 11100000 x | BM3cBFHJ | 1010010101 | ± | 0 | | |
| D8 | 00010000 x | BQ3t4mBFGJ | 0101011001 | ± | 0 | | |
| **D**9 | 10010000 0 | BTK’1u3m4bGHJ | 1001001101 | ± | 0 | | |
| D10P | 01010000 x | FT3m4bJ | 0101000001 | + | –4 | D10A | 1010111110 |
| D11 | 11010000 x | BM3c’3t’5t’7t’IJ | 1101000011 | ± | 0 | | |
| D12 | 00110000 x | FT3m4bJ | 0011000001 | + | –4 | D12A | 1100111110 |
| D13 | 10110000 x | BM3c’3t’5t’7t’IJ | 1011000011 | ± | 0 | | |
| D14 | 01110000 x | BM3c’3t’5t’7t’IJ | 0111000011 | ± | 0 | | |
| D15 | 11110000 x | BB4cAGJ | 0111001001 | ± | 0 | | |
| D16 | 00001000 x | BQ4t6tACDJ | 1011100001 | ± | 0 | | |
| D17P | 10001000 x | FT4m6mJ | 1000100001 | + | –4 | D17A | 0111011110 |
| D18P | 01001000 x | FT4m6mJ | 0100100001 | + | –4 | D18A | 1011011110 |
| D19 | 11001000 x | BM3c’3t’5t’7t’IJ | 1100100011 | ± | 0 | | |
| D20P | 00101000 x | FT4m6mJ | 0010100001 | + | –4 | D20A | 1101011110 |
| D21 | 10101000 x | BM3c’3t’5t’7t’IJ | 1010100011 | ± | 0 | | |
| D22 | 01101000 x | BM3c’3t’5t’7t’IJ | 0110100011 | ± | 0 | | |
| D23P | 11101000 x | PB3c4uJ | 1110100001 | – | 0 | D23A | 0001011110 |
| D24P | 00011000 x | FT4m6mJ | 0001100001 | + | –4 | D24A | 1110011110 |
| D25 | 10011000 x | BM3c’3t’5t’7t’IJ | 1001100011 | ± | 0 | | |

Table 10. 8B10B-P Encoding with Local Parity

| Name Primary | ABCDEFGH K | Coding Class Primary | Primary a b c d e f g h i j | Pri DR | Pri DB | Name Alternate | Alternate a b c d e f g h i j |
|---|---|---|---|---|---|---|---|
| D26 | 01011000 x | BM3c'3t'5t'7t'IJ | 0101100011 | ± | 0 | | |
| D27 | 11011000 x | BB3c'3t'I | 1101100010 | ± | 0 | | |
| D28 | 00111000 x | BM3c'3t'5t'7t'IJ | 0011100011 | ± | 0 | | |
| D29 | 10111000 x | BB3c'3t'I | 1011100010 | ± | 0 | | |
| D30 | 01111000 x | BB3c'3t'I | 0111100010 | ± | 0 | | |
| D31 | 11111000 x | BU4cCJ | 1101100001 | ± | 0 | | |
| D32 | 00000100 x | BQ4t6tACDJ | 1011010001 | ± | 0 | | |
| D33P | 10000100 x | FT4m6mJ | 1000010001 | + | –4 | D33A | 0111101110 |
| D34P | 01000100 x | FT4m6mJ | 0100010001 | + | –4 | D34A | 1011101110 |
| D35 | 11000100 x | BM3c'3t'5t'7t'IJ | 1100010011 | ± | 0 | | |
| D36P | 00100100 x | FT4m6mJ | 0010010001 | + | –4 | D36A | 1101101110 |
| D37 | 10100100 x | BM3c'3t'5t'7t'IJ | 1010010011 | ± | 0 | | |
| D38 | 01100100 x | BM3c'3t'5t'7t'IJ | 0110010011 | ± | 0 | | |
| D39P | 11100100 x | PB3c4uJ | 1110010001 | – | 0 | D39A | 0001101110 |
| D40P | 00010100 x | FT4m6mJ | 0001010001 | + | –4 | D40A | 1110101110 |
| D41 | 10010100 x | BM3c'3t'5t'7t'IJ | 1001010011 | ± | 0 | | |
| D42 | 01010100 x | BM3c'3t'5t'7t'IJ | 0101010011 | ± | 0 | | |
| D43 | 11010100 x | BB3c'3t'I | 1101010010 | ± | 0 | | |
| D44 | 00110100 x | BM3c'3t'5t'7t'IJ | 0011010011 | ± | 0 | | |
| D45 | 10110100 x | BB3c'3t'I | 1011010010 | ± | 0 | | |
| D46 | 01110100 x | BB3c'3t'I | 0111010010 | ± | 0 | | |
| D47 | 11110100 x | BU4cCJ | 1101010001 | ± | 0 | | |
| D48 | 00001100 x | BT4tADJ | 1001110001 | ± | 0 | | |
| D49 | 10001100 x | BM3c'3t'5t'7t'IJ | 1000110011 | ± | 0 | | |
| D50 | 01001100 x | BM3c'3t'5t'7t'IJ | 0100110011 | ± | 0 | | |
| D51 | 11001100 x | BB3c'3t'I | 1100110010 | ± | 0 | | |
| D52 | 00101100 x | BM3c'3t'5t'7t'IJ | 0010110011 | ± | 0 | | |
| D53 | 10101100 x | BB3c'3t'I | 1010110010 | ± | 0 | | |
| D54 | 01101100 x | BB3c'3t'I | 0110110010 | ± | 0 | | |
| D55 | 11101100 x | BU1u4uAJ | 0110110001 | ± | 0 | | |
| D56 | 00011100 x | BM3t6bAFGJ | 1001101001 | ± | 0 | | |
| D57 | 10011100 x | BB3c'3t'I | 1001110010 | ± | 0 | | |
| D58 | 01011100 x | BB3c'3t'I | 0101110010 | ± | 0 | | |
| D59 | 11011100 x | BU1u4uAJ | 0101110001 | ± | 0 | | |
| D60 | 00111100 x | BB3c'3t'I | 0011110010 | ± | 0 | | |
| D61 | 10111100 x | BU1u4uDFHJ | 1010100101 | ± | 0 | | |
| D62 | 01111100 x | BU1m6cFJ | 0111100001 | ± | 0 | | |
| D63 | 11111100 x | BC4cBDJ | 1010110001 | ± | 0 | | |
| D64 | 00000010 x | BQ6qACDJ | 1011001001 | ± | 0 | | |
| D65 | 10000010 x | BT1u6tCEJ | 1010101001 | ± | 0 | | |

**Table 10. 8B10B-P Encoding with Local Parity**

| Name Primary | ABCDEFGH K | Coding Class Primary | Primary abcdefghij | Pri DR | Pri DB | Name Alternate | Alternate abcdefghij |
|---|---|---|---|---|---|---|---|
| D66P | 01000010 x | FT1m4m6t7tJ | 0100001001 | + | –4 | D66A | 1011110110 |
| D67 | 11000010 x | BM3c'3t'5t'7t'IJ | 1100001011 | ± | 0 | | |
| D68P | 00100010 x | FT1m4m6t7tJ | 0010001001 | + | –4 | D68A | 1101110110 |
| D69 | 10100010 x | BM3c'3t'5t'7t'IJ | 1010001011 | ± | 0 | | |
| D70 | 01100010 x | BM3c'3t'5t'7t'IJ | 0110001011 | ± | 0 | | |
| D71P | 11100010 x | PB3c4uJ | 1110001001 | – | 0 | D71A | 0001110110 |
| D72P | 00010010 x | FT1m4m6t7tJ | 0001001001 | + | –4 | D72A | 1110110110 |
| D73 | 10010010 x | BM3c'3t'5t'7t'IJ | 1001001011 | ± | 0 | | |
| D74 | 01010010 x | BM3c'3t'5t'7t'IJ | 0101001011 | ± | 0 | | |
| D75 | 11010010 x | BB3c'3t'I | 1101001010 | ± | 0 | | |
| D76 | 00110010 x | BM3c'3t'5t'7t'IJ | 0011001011 | ± | 0 | | |
| D77 | 10110010 x | BB3c'3t'I | 1011001010 | ± | 0 | | |
| D78 | 01110010 x | BB3c'3t'I | 0111001010 | ± | 0 | | |
| D79 | 11110010 x | BU4cCJ | 1101001001 | ± | 0 | | |
| D80 | 00001010 x | BT4tABJ | 1100101001 | ± | 0 | | |
| **D**81 | 10001010 0 | BMK'3c'3t'5t'7t' | 1000101011 | ± | 0 | | |
| **D**82 | 01001010 0 | BMK'3c'3t'5t'7t' | 0100101011 | ± | 0 | | |
| D83 | 11001010 x | BB3c'3t'I | 1100101010 | ± | 0 | | |
| **D**84 | 00101010 0 | BMK'3c'3t'5t'7t' | 0010101011 | ± | 0 | | |
| D85 | 10101010 x | BB3c'3t'I | 1010101010 | ± | 0 | | |
| D86 | 01101010 x | BB3c'3t'I | 0110101010 | ± | 0 | | |
| D87P | 11101010 x | PU2m'4c'6u7c | 1110101000 | – | 0 | D87A | 0001010111 |
| D88P | 00011010 x | FM3t5m6m7m | 0001101000 | + | –4 | D88A | 1110010111 |
| D89 | 10011010 x | BB3c'3t'I | 1001101010 | ± | 0 | | |
| D90 | 01011010 x | BB3c'3t'I | 0101101010 | ± | 0 | | |
| D91P | 11011010 x | PU2m'4c'6u7c | 1101101000 | – | 0 | D91A | 0010010111 |
| D92 | 00111010 x | BB3c'3t'I | 0011101010 | ± | 0 | | |
| D93P | 10111010 x | PU2m'4c'6u7c | 1011101000 | – | 0 | D93A | 0100010111 |
| D94P | 01111010 x | PU2m'4c'6u7c | 0111101000 | – | 0 | D94A | 1000010111 |
| D95 | 11111010 x | BC4cABJ | 0011101001 | ± | 0 | | |
| D96 | 00000110 x | BT4tABJ | 1100011001 | ± | 0 | | |
| D97P | 10000110 x | FM4m5t6m | 1000011000 | + | –4 | D97A | 0111100111 |
| D98P | 01000110 x | FM4m5t6m | 0100011000 | + | –4 | D98A | 1011100111 |
| D99 | 11000110 x | BB3c'3t'I | 1100011010 | ± | 0 | | |
| D100P | 00100110 x | FM4m5t6m | 0010011000 | + | –4 | D100A | 1101100111 |
| D101 | 10100110 x | BB3c'3t'I | 1010011010 | ± | 0 | | |
| D102 | 01100110 x | BB3c'3t'I | 0110011010 | ± | 0 | | |
| D103P | 11100110 x | PU2m'4c'6u7c | 1110011000 | – | 0 | D103A | 0001100111 |
| D104P | 00010110 x | FM4m5t6m | 0001011000 | + | –4 | D104A | 1110100111 |
| D105 | 10010110 x | BB3c'3t'I | 1001011010 | ± | 0 | | |

**Table 10. 8B10B-P Encoding with Local Parity**

| Name Primary | ABCDEFGH K | Coding Class Primary | Primary abcdefghij | Pri DR | Pri DB | Name Alternate | Alternate abcdefghij |
|---|---|---|---|---|---|---|---|
| D106 | 01010110 x | BB3c'3t'I | 0101011010 | ± | 0 | | |
| D107P | 11010110 x | PU2m'4c'6u7c | 1101011000 | – | 0 | D107A | 0010100111 |
| D108 | 00110110 x | BB3c'3t'I | 0011011010 | ± | 0 | | |
| D109P | 10110110 x | PU2m'4c'6u7c | 1011011000 | – | 0 | D109A | 0100100111 |
| D110P | 01110110 x | PU2m'4c'6u7c | 0111011000 | – | 0 | D110A | 1000100111 |
| D111 | 11110110 x | BC4cABJ | 0011011001 | ± | 0 | | |
| D112 | 00001110 x | BM4tCJ | 0010111001 | ± | 0 | | |
| D113 | 10001110 x | BB3c'3t'I | 1000111010 | ± | 0 | | |
| D114 | 01001110 x | BB3c'3t'I | 0100111010 | ± | 0 | | |
| D115P | 11001110 x | PU2m'4c'6u7c | 1100111000 | – | 0 | D115A | 0011000111 |
| D116 | 00101110 x | BB3c'3t'I | 0010111010 | ± | 0 | | |
| D117P | 10101110 x | PU2m'4c'6u7c | 1010111000 | – | 0 | D117A | 0101000111 |
| D118P | 01101110 x | PU2m'4c'6u7c | 0110111000 | – | 0 | D118A | 1001000111 |
| D119P | 11101110 x | FC2u4u7vJ | 1110111001 | – | +4 | D119A | 0001000110 |
| D120P | 00011110 x | PB3t4mJ | 0001111001 | + | 0 | D120A | 1110000110 |
| D121P | 10011110 x | PU2m'4c'6u7c | 1001111000 | – | 0 | D121A | 0110000111 |
| D122P | 01011110 x | PU2m'4c'6u7c | 0101111000 | – | 0 | D122A | 1010000111 |
| D123P | 11011110 x | FC2u4u7vJ | 1101111001 | – | +4 | D123A | 0010000110 |
| D124 | 00111110 0 | BUK'2m7cGJ | 0011110001 | ± | 0 | | |
| D125 | 10111110 x | BC2b7vDEJ | 1010011001 | ± | 0 | | |
| D126 | 01111110 x | BC2b7vDEJ | 0110011001 | ± | 0 | | |
| D127 | 11111110 x | BV4cACDJ | 0100111001 | ± | 0 | | |
| D128 | 00000001 x | BQ6qACDJ | 1011000101 | ± | 0 | | |
| D129 | 10000001 x | BT1u6tDEJ | 1001100101 | ± | 0 | | |
| D130 | 01000001 x | BT1m2b7qDEJ | 0101100101 | ± | 0 | | |
| D131 | 11000001 0 | BMK'2u7tGJ | 1100001101 | ± | 0 | | |
| D132P | 00100001 x | FT2m4m7qJ | 0010000101 | + | –4 | D132A | 1101111010 |
| D133P | 10100001 x | FM2b3u7t | 1010000100 | + | –4 | D133A | 0101111011 |
| D134P | 01100001 x | FM2b3u7t | 0110000100 | + | –4 | D134A | 1001111011 |
| D135P | 11100001 x | PB3c4uJ | 1110000101 | – | 0 | D135A | 0001111010 |
| D136P | 00010001 x | FT2m4m7qJ | 0001000101 | + | –4 | D136A | 1110111010 |
| D137P | 10010001 x | FM3m4b7t | 1001000100 | + | –4 | D137A | 0110111011 |
| D138P | 01010001 x | FM3m4b7t | 0101000100 | + | –4 | D138A | 1010111011 |
| D139 | 11010001 x | BB3c'3t'I | 1101000110 | ± | 0 | | |
| D140P | 00110001 x | FM3m4b7t | 0011000100 | + | –4 | D140A | 1100111011 |
| D141 | 10110001 x | BB3c'3t'I | 1011000110 | ± | 0 | | |
| D142 | 01110001 x | BB3c'3t'I | 0111000110 | ± | 0 | | |
| D143 | 11110001 x | BU4cCJ | 1101000101 | ± | 0 | | |
| D144 | 00001001 x | BT4tABJ | 1100100101 | ± | 0 | | |
| D145P | 10001001 x | FM4m5m7t | 1000100100 | + | –4 | D145A | 0111011011 |

Table 10. 8B10B-P Encoding with Local Parity

| Name Primary | ABCDEFGH K | Coding Class Primary | Primary abcdefghij | Pri DR | Pri DB | Name Alternate | Alternate abcdefghij |
|---|---|---|---|---|---|---|---|
| D146P | 01001001 x | FM4m5m7t | 0100100100 | + | –4 | D146A | 1011011011 |
| D147 | 11001001 x | BB3c'3t'I | 1100100110 | ± | 0 | | |
| D148P | 00101001 x | FM4m5m7t | 0010100100 | + | –4 | D148A | 1101011011 |
| D149 | 10101001 x | BB3c'3t'I | 1010100110 | ± | 0 | | |
| D150 | 01101001 x | BB3c'3t'I | 0110100110 | ± | 0 | | |
| D151P | 11101001 x | PU3c4u7u | 1110100100 | – | 0 | D151A | 0001011011 |
| D152P | 00011001 x | FM4m5m7t | 0001100100 | + | –4 | D152A | 1110011011 |
| D153 | 10011001 x | BB3c'3t'I | 1001100110 | ± | 0 | | |
| D154 | 01011001 x | BB3c'3t'I | 0101100110 | ± | 0 | | |
| D155P | 11011001 x | PU3u5c7u | 1101100100 | – | 0 | D155A | 0010011011 |
| D156 | 00111001 x | BB3c'3t'I | 0011100110 | ± | 0 | | |
| D157P | 10111001 x | PU3u5c7u | 1011100100 | – | 0 | D157A | 0100011011 |
| D158P | 01111001 x | PU3u5c7u | 0111100100 | – | 0 | D158A | 1000011011 |
| D159 | 11111001 x | BC4cABJ | 0011100101 | ± | 0 | | |
| D160 | 00000101 x | BT4tABJ | 1100010101 | ± | 0 | | |
| D161P | 10000101 x | FM4m5t6m | 1000010100 | + | –4 | D161A | 0111101011 |
| D162P | 01000101 x | FM4m5t6m | 0100010100 | + | –4 | D162A | 1011101011 |
| D163 | 11000101 x | BB3c'3t'I | 1100010110 | ± | 0 | | |
| D164P | 00100101 x | FM4m5t6m | 0010010100 | + | –4 | D164A | 1101101011 |
| D165 | 10100101 x | BB3c'3t'I | 1010010110 | ± | 0 | | |
| D166 | 01100101 x | BB3c'3t'I | 0110010110 | ± | 0 | | |
| D167P | 11100101 x | PU3c4u7u | 1110010100 | – | 0 | D167A | 0001101011 |
| D168P | 00010101 x | FM4m5t6m | 0001010100 | + | –4 | D168A | 1110101011 |
| D169 | 10010101 x | BB3c'3t'I | 1001010110 | ± | 0 | | |
| D170 | 01010101 x | BB3c'3t'I | 0101010110 | ± | 0 | | |
| D171 | 11010101 x | BU3c'5c'7c'3t' | 1101010100 | ± | 0 | | |
| D172 | 00110101 x | BB3c'3t'I | 0011010110 | ± | 0 | | |
| D173 | 10110101 x | BU3c'5c'7c'3t' | 1011010100 | ± | 0 | | |
| D174 | 01110101 x | BU3c'5c'7c'3t' | 0111010100 | ± | 0 | | |
| D175 | 11110101 x | BC4cABJ | 0011010101 | ± | 0 | | |
| D176 | 00001101 x | BM4tCJ | 0010110101 | ± | 0 | | |
| D177 | 10001101 x | BB3c'3t'I | 1000110110 | ± | 0 | | |
| D178 | 01001101 x | BB3c'3t'I | 0100110110 | ± | 0 | | |
| D179 | 11001101 x | BU3c'5c'7c'3t' | 1100110100 | ± | 0 | | |
| D180 | 00101101 x | BB3c'3t'I | 0010110110 | ± | 0 | | |
| D181 | 10101101 x | BU3c'5c'7c'3t' | 1010110100 | ± | 0 | | |
| D182 | 01101101 x | BU3c'5c'7c'3t' | 0110110100 | ± | 0 | | |
| D183P | 11101101 x | FC1u4u6c7cJ | 1110110101 | – | +4 | D183A | 0001001010 |
| D184P | 00011101 x | PB3t4mJ | 0001110101 | + | 0 | D184A | 1110001010 |
| D185 | 10011101 x | BU3c'5c'7c'3t' | 1001110100 | ± | 0 | | |

**Table 10. 8B10B-P Encoding with Local Parity**

| Name Primary | ABCDEFGH K | Coding Class Primary | Primary abcdefghij | Pri DR | Pri DB | Name Alternate | Alternate abcdefghij |
|---|---|---|---|---|---|---|---|
| D186 | 01011101 x | BU3c'5c'7c'3t' | 0101110100 | ± | 0 | | |
| D187P | 11011101 x | FC1u4u6c7cJ | 1101110101 | – | +4 | D187A | 0010001010 |
| D188 | 00111101 x | BU3c'5c'7c'3t' | 0011110100 | ± | 0 | | |
| D189P | 10111101 x | FC1u4u6c7cJ | 1011110101 | – | +4 | D189A | 0100001010 |
| D190 | 01111101 x | BC1m6c7cCEJ | 0101010101 | ± | 0 | | |
| D191 | 11111101 x | BV4cACDJ | 0100110101 | ± | 0 | | |
| D192 | 00000011 x | BT4tBDJ | 0101001101 | ± | 0 | | |
| D193 | 10000011 x | BM1u6tFJ | 1000011101 | ± | 0 | | |
| D194P | 01000011 x | FM1m4m6t | 0100001100 | + | –4 | D194A | 1011110011 |
| D195 | 11000011 x | BB3c'3t'I | 1100001110 | ± | 0 | | |
| D196P | 00100011 x | FM1m4m6t | 0010001100 | + | –4 | D196A | 1101110011 |
| D197 | 10100011 x | BB3c'3t'I | 1010001110 | ± | 0 | | |
| D198 | 01100011 x | BB3c'3t'I | 0110001110 | ± | 0 | | |
| D199P | 11100011 x | PU3c4u7u | 1110001100 | – | 0 | D199A | 0001110011 |
| D200P | 00010011 x | FM1m4m6t | 0001001100 | + | –4 | D200A | 1110110011 |
| D201 | 10010011 x | BB3c'3t'I | 1001001110 | ± | 0 | | |
| D202 | 01010011 x | BB3c'3t'I | 0101001110 | ± | 0 | | |
| D203 | 11010011 x | BU3c'5c'7c'3t' | 1101001100 | ± | 0 | | |
| D204 | 00110011 x | BB3c'3t'I | 0011001110 | ± | 0 | | |
| D205 | 10110011 x | BU3c'5c'7c'3t' | 1011001100 | ± | 0 | | |
| D206 | 01110011 x | BU3c'5c'7c'3t' | 0111001100 | ± | 0 | | |
| D207 | 11110011 x | BC4cADJ | 0110001101 | ± | 0 | | |
| D208 | 00001011 x | BM4tCJ | 0010101101 | ± | 0 | | |
| D209 | 10001011 x | BB3c'3t'I | 1000101110 | ± | 0 | | |
| D210 | 01001011 x | BB3c'3t'I | 0100101110 | ± | 0 | | |
| D211 | 11001011 x | BU3c'5c'7c'3t' | 1100101100 | ± | 0 | | |
| D212 | 00101011 x | BB3c'3t'I | 0010101110 | ± | 0 | | |
| D213 | 10101011 x | BU3c'5c'7c'3t' | 1010101100 | ± | 0 | | |
| D214 | 01101011 x | BU3c'5c'7c'3t' | 0110101100 | ± | 0 | | |
| D215P | 11101011 x | FC4u6uJ | 1110101101 | – | +4 | D215A | 0001010010 |
| D216P | 00011011 x | PB3t4mJ | 0001101101 | + | 0 | D216A | 1110010010 |
| D217 | 10011011 x | BU3c'5c'7c'3t' | 1001101100 | ± | 0 | | |
| D218 | 01011011 x | BU3c'5c'7c'3t' | 0101101100 | ± | 0 | | |
| D219P | 11011011 x | FC4u6uJ | 1101101101 | – | +4 | D219A | 0010010010 |
| D220 | 00111011 x | BU3c'5c'7c'3t' | 0011101100 | ± | 0 | | |
| D221P | 10111011 x | FC4u6uJ | 1011101101 | – | +4 | D221A | 0100010010 |
| D222P | 01111011 x | FC4u6uJ | 0111101101 | – | +4 | D222A | 1000010010 |
| D223 | 11111011 x | BV4cACDJ | 0100101101 | ± | 0 | | |
| D224 | 00000111 x | BM4tCJ | 0010011101 | ± | 0 | | |
| D225 | 10000111 x | BB3c'3t'I | 1000011110 | ± | 0 | | |

**Table 10. 8B10B-P Encoding with Local Parity**

| Name Primary | ABCDEFGH K | Coding Class Primary | Primary abcdefghij | Pri DR | Pri DB | Name Alternate | Alternate abcdefghij |
|---|---|---|---|---|---|---|---|
| D226 | 01000111 x | BB3c'3t'I | 0100011110 | ± | 0 | | |
| D227 | 11000111 x | BU3c'5c'7c'3t' | 1100011100 | ± | 0 | | |
| D228 | 00100111 x | BB3c'3t'I | 0010011110 | ± | 0 | | |
| D229 | 10100111 x | BU3c'5c'7c'3t' | 1010011100 | ± | 0 | | |
| D230 | 01100111 x | BU3c'5c'7c'3t' | 0110011100 | ± | 0 | | |
| D231P | 11100111 x | FC4u6uJ | 1110011101 | – | +4 | D231A | 0001100010 |
| D232P | 00010111 x | PB3t4mJ | 0001011101 | + | 0 | D232A | 1110100010 |
| D233 | 10010111 x | BU3c'5c'7c'3t' | 1001011100 | ± | 0 | | |
| D234 | 01010111 x | BU3c'5c'7c'3t' | 0101011100 | ± | 0 | | |
| D235P | 11010111 x | FC4u6uJ | 1101011101 | – | +4 | D235A | 0010100010 |
| D236 | 00110111 x | BU3c'5c'7c'3t' | 0011011100 | ± | 0 | | |
| D237P | 10110111 x | FC4u6uJ | 1011011101 | – | +4 | D237A | 0100100010 |
| D238P | 01110111 x | FC4u6uJ | 0111011101 | – | +4 | D238A | 1000100010 |
| D239 | 11110111 x | BV4cACDJ | 0100011101 | ± | 0 | | |
| D240 | 00001111 x | BB4tAGJ | 1000110101 | ± | 0 | | |
| D241 | 10001111 x | BU3c'5c'7c'3t' | 1000111100 | ± | 0 | | |
| D242 | 01001111 x | BU3c'5c'7c'3t' | 0100111100 | ± | 0 | | |
| D243P | 11001111 x | FCK'3u4bJ | 1100111101 | – | +4 | D243A | 0011000010 |
| D244 | 00101111 x | BU3c'5c'7c'3t' | 0010111100 | ± | 0 | | |
| D245P | 10101111 x | FCK'3u4bJ | 1010111101 | – | +4 | D245A | 0101000010 |
| D246P | 01101111 x | BC1m4bEGJ | 0110111101 | – | +4 | D246A | 1001000010 |
| D247P | 11101111 x | FV3c4u | 1110111100 | – | +4 | D247A | 0001000011 |
| **D**248 | 00011111 0 | BUK'3tBFHJ | 0101101001 | ± | 0 | | |
| D249P | 10011111 x | BC1u3mEFJ | 1001111101 | – | +4 | D249A | 0110000010 |
| D250P | 01011111 x | BC1m4bEHJ | 0101111101 | – | +4 | D250A | 1010000010 |
| D251 | 11011111 x | BV2u3uBDFJ | 1000101101 | ± | 0 | | |
| D252 | 00111111 x | BC1m4bEFJ | 0011001101 | ± | 0 | | |
| D253 | 10111111 x | BV2bCDHJ | 1000111001 | ± | 0 | | |
| D254 | 01111111 x | BV2bDFGJ | 0110100101 | ± | 0 | | |
| D255 | 11111111 x | BHBCEGJ | 1001010101 | ± | 0 | | |
| C9P | 10010000 1 | FTK1u3m4b | 1001000001 | + | –4 | C9A | 0110111110 |
| K81P | 10001010 1 | FMK3m4m5m6m7m | 1000101000 | + | –4 | K81A | 0111010111 |
| K82P | 01001010 1 | FMK3m4m5m6m7m | 0100101000 | + | –4 | K82A | 1011010111 |
| K84P | 00101010 1 | FMK3m4m5m6m7m | 0010101000 | + | –4 | K84A | 1101010111 |
| *K124P* | *00111110 1* | *PUK2m7c* | *0011111000* | – | 0 | *K124A* | *1100000111* |
| *K131P* | *11000001 1* | *FMK2u7t* | *1100000100* | + | –4 | *K131A* | *0011111011* |
| *K248P* | *00011111 1* | *PUK3t* | *0001111100* | + | 0 | *K248A* | *1110000011* |

## E) Generation of Encoded 10B Vectors

Generally, the encoded bits retain the value of the unencoded bit (a=A, b=B, etc), but a specific source bit is complemented (a=A', b=B', etc) if and only if (iff) the respective equation is true. In the Coding Labels and equations, some bit values are included redundantly to allow more circuit sharing for the coding of several bits. Redundant bit values are overlined and redundant vector names are preceded by an asterisk. In the Tables 11 through 24, the bit patterns common to several vectors are marked by bold type to logically classify the vectors by simple expressions listed in the column 'Coding Label'. The labels are used to write the encoding equations. In any of the Exclusive OR relationships between two Groups of 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 other expressions are included in the net.

### 1) Encoded Bit a

The 'a' column has bold entries in the Tables 9 and 10 for the vectors listed in Table 11.

**Table 11. a-bit Encoding**

| Name | ABCDEFGH K | a | Name | ABCDEFGH K | a | Coding Label |
|---|---|---|---|---|---|---|
| D80 | **0000**1010 x | 1 | D175 | **1111**0101 x | 0 | |
| D96 | **0000**0110 x | 1 | D159 | **1111**1001 x | 0 | A⊕B'•B⊕C'•C⊕D'•E⊕F•G⊕H |
| D144 | **0000**1001 x | 1 | D111 | **1111**0110 x | 0 | |
| D160 | **0000**0101 x | 1 | D95 | **1111**1010 x | 0 | |
| D128 | **000000**01 x | 1 | D127 | **111111**10 x | 0 | A⊕B'•B⊕C'•C⊕D'• |
| D64 | **000000**10 x | 1 | D191 | **111111**01 x | 0 | (D⊕E'•E⊕F'•G⊕H+ |
| D32 | **0000**01**00** x | 1 | D223 | **1111**10**11** x | 0 | A⊕G'•G⊕H'•E⊕F) |
| D16 | **0000**10**00** x | 1 | D239 | **1111**01**11** x | 0 | |
| D48 | **000011**00 x | 1 | D207 | **111100**11 x | 0 | A⊕B'•B⊕C'•C⊕D'• |
| D240 | **000011**11 x | 1 | D15 | **111100**00 x | 0 | D⊕E•E⊕F'•G⊕H' |
| | | | D55 | **11**10**1100** x | 0 | |
| | | | D59 | **11**01**1100** x | 0 | (C⊕D•A•B+A'•B'•C') |
| D56 | **000**1**1100** x | 1 | | | | •E•F•G'•H' |
| *D48 | **000**0**1100** x | 1 | | | | |

*a = A⊕{(D⊕E'•E⊕F'•G⊕H + A⊕G'•G⊕H'•E⊕F + D⊕E•E⊕F'•G⊕H' + E⊕F•G⊕H) •*
*A⊕B'•B⊕C'•C⊕D' +* (Pn1)
*(C⊕D•A•B + A'•B'•C') • E•F•G'•H'}* (Pn3)

### 2) Encoded Bit b

The ‘b’ column has bold entries in the Tables 9 and 10 for the vectors listed in Table 12.

**Table 12. b-bit Encoding**

| Name | ABCDEFGH K | b | Name | ABCDEFGH K | b | Coding Label |
|---|---|---|---|---|---|---|
| D80 | **0000**1010 x | 1 | D175 | **1111**0101 x | 0 | $\overline{K'}$•A⊕B’•B⊕C’•C⊕D’•E⊕F•G⊕H |
| D96 | **0000**0110 x | 1 | D159 | **1111**1001 x | 0 | |
| D144 | **0000**1001 x | 1 | D111 | **1111**0110 x | 0 | |
| D160 | **0000**0101 x | 1 | D95 | **1111**1010 x | 0 | |
| **D**248 | **000**11111 0 | 1 | D7 | **111**00000 x | 0 | K’•A⊕B’•B⊕C’• D⊕E’•E⊕F’•F⊕G’•G⊕H’ |
| D0 | **000**00000 x | 1 | D255 | **111**11111 x | 0 | |
| D4 | **00**1**000**00 x | 1 | D251 | **11**0**111**11 x | 0 | A⊕B’•B⊕E’•D⊕E’•E⊕F’•G⊕H’• (F⊕G’+B⊕C’) |
| *D0 | **00**0**000**00 x | 1 | *D255 | **11**1**111**11 x | 0 | |
| D192 | **00**0**000**11 x | 1 | D63 | **11**1**111**00 x | 0 | |
| D8 | **000**1**0000** x | 1 | | | | A’•B’•C’•E’•F’•G’•H’ |
| *D0 | **000**0**0000** x | 1 | | | | |

*b = B⊕{(D⊕E’•E⊕F’•F⊕G’•G⊕H’ + C⊕D’•E⊕F•G⊕H) • A⊕B’ • B⊕C’ • K’ +* (Pn7)
*A⊕B’•B⊕E’•D⊕E’•E⊕F’•G⊕H’•(B⊕C’+F⊕G’) + A’•B’•C’•E’•F’•G’•H’}*(Pn5+Pn8)

### 3) Encoded Bit c

The ‘c’ column has bold entries in the Tables 9 and 10 for the vectors listed in Table 13.

**Table 13. c-bit Encoding**

| Name | ABCDEFGH K | c | Name | ABCDEFGH K | c | Coding Label |
|---|---|---|---|---|---|---|
| D224 | **0000**01**11** x | 1 | D31 | **1111**10**00** x | 0 | A⊕B’•B⊕C’•C⊕D’•E⊕F•G⊕H’ |
| D208 | **0000**10**11** x | 1 | D47 | **1111**01**00** x | 0 | |
| D32 | **0000**01**00** x | 1 | D223 | **1111**10**11** x | 0 | |
| D16 | **0000**10**00** x | 1 | D239 | **1111**01**11** x | 0 | |
| D176 | **000011**01 x | 1 | D79 | **111100**10 x | 0 | A⊕B’•B⊕C’•C⊕D’•E⊕F’•G⊕H |
| D112 | **000011**10 x | 1 | D143 | **111100**01 x | 0 | |
| D128 | **000000**01 x | 1 | D127 | **111111**10 x | 0 | |
| D64 | **000000**10 x | 1 | D191 | **111111**01 x | 0 | |
| D0 | **00000000** x | 1 | D255 | **11111111** x | 0 | A⊕B’•B⊕C’•C⊕D’• C⊕H’•D⊕E’•E⊕F’ |
| *D64 | **00000010** x | 1 | D191 | **11111101** x | 0 | |
| D2 | 01**000000** x | 1 | D253 | 10**111111** x | 0 | A⊕B•A⊕G’•C⊕D’•D⊕E’• E⊕F’•C⊕H’ |
| D65 | 10**000010** x | 1 | D190 | 01**111101** x | 0 | |

*c = C⊕{A⊕B • A⊕G’ • C⊕D’ • C⊕H’ • D⊕E’ • E⊕F’ +* (Pn12)
*(C⊕H’•D⊕E’•E⊕F’ + E⊕F•G⊕H’ + E⊕F’•G⊕H) • A⊕B’•B⊕C’•C⊕D’}* (Pn11)

### 4) Encoded Bit d

The ‘d’ column has bold entries in the Tables 9 and 10 for the vectors listed in Table 14.

**Table 14. d-bit Encoding**

| Name | ABCDEFGH K | d | Name | ABCDEFGH K | d | Coding Label |
|---|---|---|---|---|---|---|
| D128 | 00000001 x | 1 | D127 | 11111110 x | 0 | A⊕B’•B⊕C’•C⊕D’•D⊕E’•E⊕F’• (F⊕G+G⊕H) |
| D64 | 00000010 x | 1 | D191 | 11111101 x | 0 | |
| D192 | 00000011 x | 1 | D63 | 11111100 x | 0 | |
| D32 | 00000100 x | 1 | D223 | 11111011 x | 0 | A⊕B’•B⊕C’•C⊕D’•C⊕H’•G⊕H’• (F⊕G+E⊕F) |
| D16 | 00001000 x | 1 | D239 | 11110111 x | 0 | |
| D48 | 00001100 x | 1 | D207 | 11110011 x | 0 | |
| D130 | 01000001 x | 1 | D125 | 10111110 x | 0 | A⊕B•C⊕D’•D⊕E’•E⊕F’•F⊕G’ |
| D129 | 10000001 x | 1 | D126 | 01111110 x | 0 | |
| D2 | 01000000 x | 1 | D253 | 10111111 x | 0 | |
| D1 | 10000000 x | 1 | D254 | 01111111 x | 0 | |
| D4 | 00100000 x | 1 | D251 | 11011111 x | 0 | A⊕G’•B⊕C•D⊕E’•E⊕F’•F⊕G’• G⊕H’ |
| *D2 | 01000000 x | 1 | *D253 | 10111111 x | 0 | |
| | | | D61 | 10111100 x | 0 | A•B’•C•D•E•F•$\overline{G}$’•H’ |
| | | | *D125 | 10111110 x | 0 | |

$$d = D\oplus\{(A\oplus G'\bullet B\oplus C\bullet G\oplus H' + A\oplus B\bullet C\oplus D') \bullet D\oplus E'\bullet E\oplus F'\bullet F\oplus G' + \quad \text{(Pn20)}$$
$$(F\oplus G + G\oplus H) \bullet D\oplus E'\bullet E\oplus F' \bullet A\oplus B'\bullet B\oplus C'\bullet C\oplus D' + \quad \text{(Pn19)}$$
$$(F\oplus G + E\oplus F) \bullet C\oplus H'\bullet G\oplus H' \bullet A\oplus B'\bullet B\oplus C'\bullet C\oplus D' + \quad \text{(Pn15)}$$
$$A\bullet B'\bullet C\bullet D \bullet E\bullet F\bullet \overline{G}'\bullet H'\}$$

### 5) Encoded Bit e

The ‘e’ column has bold entries in the Tables 9 and 10 for the vectors listed in Table 15.

**Table 15. e-bit Encoding**

| Name | ABCDEFGH K | e | Name | ABCDEFGH K | e | Coding Label |
|---|---|---|---|---|---|---|
| D0 | 00000000 x | 1 | D255 | 11111111 x | 0 | C⊕D’•D⊕E’•E⊕F’• (A⊕B’•F⊕G’•G⊕H’+ A⊕B•B⊕C’•G⊕H+ A⊕B•F⊕G’•G⊕H) |
| D3 | 11000000 x | 1 | D252 | 00111111 x | 0 | |
| D65 | 10000010 x | 1 | D190 | 01111101 x | 0 | |
| D130 | 01000001 x | 1 | D125 | 10111110 x | 0 | |
| D129 | 10000001 x | 1 | D126 | 01111110 x | 0 | |

$$e = E\oplus\{(A\oplus B'\bullet F\oplus G'\bullet G\oplus H' + A\oplus B \bullet B\oplus C'\bullet G\oplus H + A\oplus B \bullet F\oplus G'\bullet G\oplus H) \bullet \quad \text{(n21)}$$
$$C\oplus D' \bullet D\oplus E' \bullet E\oplus F'\}$$

### 6) Encoded Bit f

The ‘f’ column has bold entries in the Tables 9 and 10 for the vectors listed in Table 16.

**Table 16. f-bit Encoding**

| Name | ABCDEFGH K | f | Name | ABCDEFGH K | f | Coding Label |
|---|---|---|---|---|---|---|
| D193 | 10000011 x | 1 | D62 | 01111100 x | 0 | A⊕B•B⊕C’•C⊕D’•D⊕E’•E⊕F’• |
| D1 | 10000000 x | 1 | D254 | 01111111 x | 0 | G⊕H’ |
| D7 | 11100000 x | 1 | D248 | 00011111 0 | 0 | A⊕B’•D⊕E’•E⊕F’•F⊕G’•G⊕H’• |
| D3 | 11000000 x | 1 | D252 | 00111111 x | 0 | (A⊕G+B⊕C)•K’ |
| D4 | 00100000 x | 1 | D251 | 11011111 x | 0 | |
| *D4 | 00100000 x | 1 | | | | |
| D6 | 01100000 x | 1 | | | | A’•E’•F’•G’•H’•(B’•C’•D+C•D’) |
| D8 | 00010000 x | 1 | | | | |
| | | | D56 | 00011100 x | 0 | A⊕C’•B’•D•E•F•G’•H’ |
| | | | D61 | 10111100 x | 0 | |

$$f = F\oplus\{(A\oplus G+B\oplus C) \bullet A\oplus B' \bullet D\oplus E'\bullet E\oplus F'\bullet F\oplus G' \bullet G\oplus H' \bullet K' + \quad \text{(Pn22)}$$
$$A\oplus B\bullet B\oplus C'\bullet C\oplus D' \bullet D\oplus E'\bullet E\oplus F'\bullet G\oplus H' + \quad \text{(Pn24)}$$
$$(\overline{A}'\bullet B'\bullet C'\bullet D + C\bullet D') \bullet A' \bullet E'\bullet F'\bullet G'\bullet H' + A\oplus C'\bullet B' \bullet D\bullet E\bullet F\bullet G'\bullet H'\} \quad \text{(Pn26)}$$

## 7) Encoded Bit g

The ‘g’ column has bold entries in the Tables 9 and 10 for the vectors listed in Table 17.

**Table 17. g-bit Encoding**

| Name | ABCDEFGH K | g | Name | ABCDEFGH K | g | Coding Label |
|---|---|---|---|---|---|---|
| D131 | 11000001 0 | 1 | D124 | 00111110 0 | 0 | A⊕B’•B⊕E•E⊕F’•F⊕G’• |
| D15 | 11110000 x | 1 | D240 | 00001111 x | 0 | C⊕D’•C⊕H•K’ |
| D0 | 00000000 x | 1 | D255 | 11111111 x | 0 | B⊕C’•C⊕D’•D⊕E’•E⊕F’•F⊕G’• |
| D1 | 10000000 x | 1 | D254 | 01111111 x | 0 | G⊕H’ |
| *D1 | 10000000 x | 1 | | | | |
| D5 | 10100000 x | 1 | | | | (A•D’+C’•D)•B’•E’•F’•G’•H’•K’ |
| D8 | 00010000 x | 1 | | | | |
| D9 | 10010000 0 | 1 | | | | |
| D56 | 00011100 x | 1 | | | | A’•B’•C’•D•E•F•G’•H’ |

$$g = G\oplus\{A\oplus B' \bullet B\oplus E \bullet E\oplus F'\bullet F\oplus G' \bullet C\oplus D' \bullet C\oplus H \bullet K' + \quad \text{(Pn98)}$$
$$B\oplus C'\bullet C\oplus D' \bullet D\oplus E'\bullet E\oplus F'\bullet F\oplus G' \bullet G\oplus H' + \quad \text{(Pn30)}$$
$$(A\bullet D' + C'\bullet D) \bullet B' \bullet E'\bullet F'\bullet G'\bullet H' \bullet K' + A'\bullet B'\bullet C'\bullet D \bullet E\bullet F\bullet G'\bullet H'\} \quad \text{(Pn33)}$$

## 8) Encoded Bit h

The ‘h’ column has bold entries in the Tables 9 and 10 for the vectors listed in Table 18.

**Table 18. h-bit Encoding**

| Name | ABCDEFGH K | h | Name | ABCDEFGH K | h | Coding Label |
|---|---|---|---|---|---|---|
| D7 | 11100000 x | 1 | D248 | 00011111 0 | 0 | A⊕C'•B⊕E•D⊕E'•E⊕F'•F⊕G'• G⊕H' • K' |
| D2 | 01000000 x | 1 | D253 | 10111111 x | 0 | |
| D9 | 10010000 0 | 1 | | | | K'•A•B'•C'•D•E'•F'•G'•H' |
| D5 | 10100000 x | 1 | | | | (A+B)•C•D'•E'•F'•G'•H' |
| D6 | 01100000 x | 1 | | | | |
| *D7 | 11100000 x | 1 | | | | |
| D61 | 10111100 x | 1 | | | | A•B'•C•D•E•F•G'•H' |

$$h = H \oplus \{A \oplus C' \bullet B \oplus E \bullet D \oplus E' \bullet E \oplus F' \bullet F \oplus G' \bullet G \oplus H' \bullet K' + A \bullet B' \bullet C \bullet D \bullet E \bullet F \bullet G' \bullet H' + \quad \text{(Pn37)}$$
$$[K' \bullet A \bullet B' \bullet C' \bullet D + (A+B) \bullet C \bullet D'] \bullet E' \bullet F' \bullet G' \bullet H'\} \quad \text{(Pn35)}$$

## 9) Encoded Bit i

The 'i' column has bold entries in the Tables 2 and 3 for the 87 vectors listed in Table 19.

**Table 19. i-bit Encoding**

| Name | ABCDEFGH | Name | ABCDEFGH | Coding Label |
|---|---|---|---|---|
| D51 | 11001100 | D204 | 00110011 | (A⊕D•B⊕C+A⊕B•C⊕D)•E⊕F'•F⊕G•G⊕H' |
| D60 | 00111100 | D195 | 11000011 | |
| D53 | 10101100 | D202 | 01010011 | |
| D54 | 01101100 | D201 | 10010011 | |
| D57 | 10011100 | D198 | 01100011 | |
| D58 | 01011100 | D197 | 10100011 | |
| D147 | 11001001 | D108 | 00110110 | (A⊕D•B⊕C+A⊕B•C⊕D)•E⊕F•G⊕H |
| D156 | 00111001 | D99 | 11000110 | |
| D149 | 10101001 | D106 | 01010110 | |
| D150 | 01101001 | D105 | 10010110 | |
| D153 | 10011001 | D102 | 01100110 | |
| D154 | 01011001 | D101 | 10100110 | |
| D83 | 11001010 | D172 | 00110101 | |
| D92 | 00111010 | D163 | 11000101 | |
| D85 | 10101010 | D170 | 01010101 | |
| D86 | 01101010 | D169 | 10010101 | |
| D89 | 10011010 | D166 | 01100101 | |
| D90 | 01011010 | D165 | 10100101 | |
| D11 | 11010000 | | | (A•B•C'+A•B'•C+A'•B•C)•D•E'•F'•G'•H' |
| D13 | 10110000 | | | |
| D14 | 01110000 | | | |

Table 19. i-bit Encoding

| Name | ABCDEFGH | Name | ABCDEFGH | Coding Label |
|---|---|---|---|---|
| D27 | 110**11000** | D228 | 001**00111** | |
| D29 | 101**11000** | D226 | 010**00111** | |
| D30 | 011**11000** | D225 | 100**00111** | (A⊕B'•B⊕C•C⊕D+A⊕B•C⊕D')• |
| D43 | 110**10100** | D212 | 001**01011** | D⊕G•E⊕F•G⊕H' |
| D45 | 101**10100** | D210 | 010**01011** | |
| D46 | 011**10100** | D209 | 100**01011** | |
| D75 | 110**10010** | D180 | 001**01101** | |
| D77 | 101**10010** | D178 | 010**01101** | |
| D78 | 011**10010** | D177 | 100**01101** | (A⊕B'•B⊕C•C⊕D+A⊕B•C⊕D')• |
| D139 | 110**10001** | D116 | 001**01110** | D⊕E•E⊕F'•G⊕H |
| D141 | 101**10001** | D114 | 010**01110** | |
| D142 | 011**10001** | D113 | 100**01110** | |
| D49 | 100**01**10**0** | | | |
| D50 | 010**01**10**0** | | | |
| D52 | 001**01**10**0** | | | (A•B'•C'+A'•B•C'+A'•B'•C)• |
| **D**81 | 100**01**01**0** | | K=0 | F⊕G•K'•D'•E•H' |
| **D**82 | 010**01**01**0** | | K=0 | |
| **D**84 | 001**01**01**0** | | K=0 | |
| D19 | 1100**1000** | | | |
| D28 | 0011**1000** | | | |
| D21 | 1010**1000** | | | (A⊕D•B⊕C+A⊕B•C⊕D)•E•F'•G'•H' |
| D22 | 0110**1000** | | | |
| D25 | 1001**1000** | | | |
| D26 | 0101**1000** | | | |
| D35 | 1100**0100** | | | |
| D44 | 0011**0100** | | | |
| D37 | 1010**0100** | | | (A⊕D•B⊕C+A⊕B•C⊕D)•E'•F•G'•H' |
| D38 | 0110**0100** | | | |
| D41 | 1001**0100** | | | |
| D42 | 0101**0100** | | | |
| D67 | 1100**0010** | | | |
| D76 | 0011**0010** | | | |
| D69 | 1010**0010** | | | (A⊕D•B⊕C+A⊕B•C⊕D)•E'•F'•G•H' |
| D70 | 0110**0010** | | | |
| D73 | 1001**0010** | | | |
| D74 | 0101**0010** | | | |

$$i = (E \oplus F' \bullet F \oplus G \bullet G \oplus H' + E \oplus F \bullet G \oplus H) \bullet (A \oplus D \bullet B \oplus C + A \oplus B \bullet C \oplus D) + \quad \text{(n51)}$$
$$(A \oplus D \bullet B \oplus C + A \oplus B \bullet C \oplus D) \bullet (E \bullet F' \bullet G' + E' \bullet F \bullet G' + E' \bullet F' \bullet G) \bullet H' + \quad \text{(n52)}$$
$$(D \oplus G \bullet E \oplus F \bullet G \oplus H' + D \oplus E \bullet E \oplus F' \bullet G \oplus H) \bullet \quad \text{(Pn40)}$$
$$(A \oplus B' \bullet B \oplus C \bullet C \oplus D + A \oplus B \bullet C \oplus D') + \quad \text{(n50)}$$
$$F \oplus G \bullet K' \bullet D' \bullet E \bullet H' \bullet (A \bullet B' \bullet C' + A' \bullet B \bullet C' + A' \bullet B' \bullet C) + \quad \text{(Pn47)}$$
$$(A \bullet B \bullet C' + A \bullet B' \bullet C + A' \bullet B \bullet C) \bullet D \bullet E' \bullet F' \bullet G' \bullet H' + \quad \text{(Pn49)}$$

### 10) Encoded Bit j

The ‘j’ column has bold entries in the Tables 2, 4, 5, 6, 8, and 9 for the 129 vectors listed in Table 20.

**Table 20. j-bit Encoding**

| Name | ABCDEFGH | Name | ABCDEFGH | Coding Label |
|---|---|---|---|---|
| D0 | 0000**0000** | | | E’•F’•G’•H’ |
| D1 | 1000**0000** | | | |
| D2 | 0100**0000** | | | |
| D3 | 1100**0000** | | | |
| D4 | 0010**0000** | | | |
| D5 | 1010**0000** | | | |
| D6 | 0110**0000** | | | |
| D7 | 1110**0000** | | | |
| D8 | 0001**0000** | | | |
| **D**9 | 1001**0000** | | K=0 | |
| C9 | 1001**0000** | | K=1 | |
| D10 | 0101**0000** | | | |
| D11 | 1101**0000** | | | |
| D12 | 0011**0000** | | | |
| D13 | 1011**0000** | | | |
| D14 | 0111**0000** | | | |
| D15 | 1111**0000** | | | |
| D66 | **0**10**00010** | D189 | **1**01**11101** | (A⊕G•B⊕C•D⊕E’+A⊕D•B⊕C’•C⊕H’+ A⊕B’•B⊕C’•C⊕D)•E⊕F’•F⊕G•G⊕H |
| D68 | **0**01**00010** | D187 | **1**10**11101** | |
| D65 | 1**00**0**0010** | D190 | 0**11**1**1101** | |
| D72 | 0**00**1**0010** | D183 | 1**11**0**1101** | |
| *D72 | **000**1**0010** | *D183 | **111**0**1101** | |
| D71 | **111**0**0010** | D184 | **000**1**1101** | |
| | | D215 | **1110**10**11** | E⊕F•A•B•C•D’•G•H |
| | | D231 | **1110**01**11** | |
| | | D243 | 110**01111** | (A•B•C’+A•B’•C+A’•B•C)•D’•E•F•G•H |
| | | D245 | 101**01111** | |
| | | D246 | 011**01111** | |

**Table 20. j-bit Encoding**

| Name | ABCDEFGH | Name | ABCDEFGH | Coding Label |
|---|---|---|---|---|
| | | D55 | 11101100 | |
| | | D59 | 11011100 | (C⊕D•A•B+A⊕C'•B'•D)•E•F•G'•H' |
| | | D56 | 00011100 | |
| | | D61 | 10111100 | |
| *D0 | 00000000 | D255 | 11111111 | |
| D128 | 00000001 | D127 | 11111110 | |
| D64 | 00000010 | D191 | 11111101 | |
| D192 | 00000011 | D63 | 11111100 | |
| D32 | 00000100 | D223 | 11111011 | |
| D160 | 00000101 | D95 | 11111010 | |
| D96 | 00000110 | D159 | 11111001 | |
| D224 | 00000111 | D31 | 11111000 | A⊕B'•B⊕C'•C⊕D' |
| D16 | 00001000 | D239 | 11110111 | |
| D144 | 00001001 | D111 | 11110110 | |
| D80 | 00001010 | D175 | 11110101 | |
| D208 | 00001011 | D47 | 11110100 | |
| D48 | 00001100 | D207 | 11110011 | |
| D176 | 00001101 | D79 | 11110010 | |
| D112 | 00001110 | D143 | 11110001 | |
| D240 | 00001111 | *D15 | 11110000 | |
| *D192 | 00000011 | *D63 | 11111100 | B⊕C'•C⊕D'•D⊕E'•E⊕F'•F⊕G•G⊕H' |
| D193 | 10000011 | D62 | 01111100 | |
| *D16 | 00001000 | | | |
| D17 | 10001000 | | | |
| D18 | 01001000 | | | |
| D19 | 11001000 | | | |
| D20 | 00101000 | | | |
| D21 | 10101000 | | | |
| D22 | 01101000 | | | |
| D23 | 11101000 | | | E⊕F•D'•G'•H' |
| *D32 | 00000100 | | | |
| D33 | 10000100 | | | |
| D34 | 01000100 | | | |
| D35 | 11000100 | | | |
| D36 | 00100100 | | | |
| D37 | 10100100 | | | |
| D38 | 01100100 | | | |
| D39 | 11100100 | | | |

**Table 20. j-bit Encoding**

| Name | ABCDEFGH | Name | ABCDEFGH | Coding Label |
|---|---|---|---|---|
| *D64 | 00000010 | | | |
| *D65 | 10000010 | | | |
| *D66 | 01000010 | | | |
| D67 | 11000010 | | | D’•E’•F’•G•H’ |
| *D68 | 00100010 | | | |
| D69 | 10100010 | | | |
| D70 | 01100010 | | | |
| *D71 | 11100010 | | | |
| D119 | 11101110 | D136 | 00010001 | |
| D123 | 11011110 | D132 | 00100001 | (B⊕E’•C⊕D+A⊕G•B⊕C’)• |
| D120 | 00011110 | D135 | 11100001 | A⊕B’•E⊕F’•F⊕G’•G⊕H |
| *D112 | 00001110 | *D143 | 11110001 | |
| *D80 | 00001010 | | | |
| D81 | 10001010 | | K=0 | |
| D82 | 01001010 | | K=0 | |
| D84 | 00101010 | | K=0 | (A’•B’+A’•C’+B’•C’)•D⊕E•K’•F’•G•H‘ |
| *D72 | 00010010 | | | |
| D73 | 10010010 | | | |
| D74 | 01010010 | | | |
| D76 | 00110010 | | | |
| | K=0 | D248 | 00011111 | |
| | | D249 | 10011111 | |
| | | D250 | 01011111 | |
| | | D251 | 11011111 | K’•D•E•F•G•H |
| | | *D252 | 00111111 | |
| | | *D253 | 10111111 | |
| | | *D254 | 01111111 | |
| | | *D255 | 11111111 | |
| | | D216 | 00011011 | |
| | | D219 | 11011011 | |
| | | D221 | 10111011 | |
| | | D222 | 01111011 | |
| | | *D223 | 11111011 | (A’•B’•C’+A•B•$\overline{C}$’+A•$\overline{B}$’•C+B•C)• |
| | | D232 | 00010111 | E⊕F•D•G•H |
| | | D235 | 11010111 | |
| | | D237 | 10110111 | |
| | | D238 | 01110111 | |
| | | *D239 | 11110111 | |

**Table 20. j-bit Encoding**

| Name | ABCDEFGH | Name | ABCDEFGH | Coding Label |
|---|---|---|---|---|
| D24 | 00011000 | | | |
| D25 | 10011000 | | | |
| D26 | 01011000 | | | |
| D28 | 00111000 | | | |
| D40 | 00010100 | | | |
| D41 | 10010100 | | | (D•E•F'+D•E'•F+D'•E•F)• |
| D42 | 01010100 | | | (A'•B'+A'•C'+B'•C')•G'•H' |
| D44 | 00110100 | | | |
| *D48 | 00001100 | | | |
| D49 | 10001100 | | | |
| D50 | 01001100 | | | |
| D52 | 00101100 | | | |
| *D128 | 00000001 | *D127 | 11111110 | |
| D129 | 10000001 | D126 | 01111110 | |
| D130 | 01000001 | D125 | 10111110 | |
| **D**131 | 11000001 | **D**124 | 00111110 | |
| | K=0 | | K=0 | K'•C⊕D'•D⊕E'•E⊕F'•F⊕G' |
| *D0 | 00000000 | *D255 | 11111111 | |
| *D1 | 10000000 | D254 | 01111111 | |
| *D2 | 01000000 | D253 | 10111111 | |
| *D3 | 11000000 | D252 | 00111111 | |

$$
\begin{aligned}
j = {} & \{D\oplus E\bullet K'\bullet F'\bullet G+(D\bullet E\bullet F'+D\bullet E'\bullet F+D'\bullet E\bullet F)\bullet G'\}\bullet(A'\bullet B'+A'\bullet C'+B'\bullet C')\bullet H' + && \text{(n77)}\\
& (A\oplus G\bullet B\oplus C\bullet D\oplus E'+A\oplus D\bullet B\oplus C'\bullet C\oplus H'+A\oplus B'\bullet B\oplus C'\bullet C\oplus D)\bullet E\oplus F'\bullet F\oplus G\bullet G\oplus H && \text{(n76)}\\
& + (A\bullet B\bullet C' + A\bullet B'\bullet C + A'\bullet B\bullet C)\bullet D'\bullet E\bullet F\bullet G\bullet H + && \text{(n74)}\\
& (C\oplus D\bullet A\bullet B + A\oplus C'\bullet B'\bullet D)\bullet E\bullet F\bullet G'\bullet H' + (D'\bullet G + G')\bullet E'\bullet F'\bullet H' + && \text{(n71+n68)}\\
& (B\oplus C'\bullet F\oplus G\bullet G\oplus H' + F\oplus G'\bullet K')\bullet C\oplus D'\bullet D\oplus E'\bullet E\oplus F' + && \text{(n73)}\\
& (B\oplus E'\bullet C\oplus D + A\oplus G\bullet B\oplus C')\bullet A\oplus B'\bullet E\oplus F'\bullet F\oplus G'\bullet G\oplus H + && \text{(n75)}\\
& E\oplus F\bullet A\bullet B\bullet C\bullet D'\bullet G\bullet H + E\oplus F\bullet D'\bullet G'\bullet H' + K'\bullet D\bullet E\bullet F\bullet G\bullet H + A\oplus B'\bullet B\oplus C'\bullet C\oplus D' && \text{(n67*)}\\
& + (A'\bullet B'\bullet C' + A\bullet B\bullet\overline{C'} + A\bullet\overline{B'}\bullet C + B\bullet C)\bullet E\oplus F\bullet D\bullet G\bullet H && \text{(n69)}
\end{aligned}
$$

## F) Equations for the Required Disparity for Encoding DRE

It is assumed that with K=1 only the seven valid control vectors are presented at the input to the encoder so simple control vector labels can be derived from the last seven rows of Table 10.

### 1) Positive Required Disparity for Encoding: PDRE

A total of 46 vectors listed in the Tables 5, 7, and 8 require a positive entry disparity (PDRE). They are listed and sorted in Table 21. Redundant bits are overlined.

**Table 21. Positive Required Disparity PDRE**

| Name | ABCDEFGH | Coding Label |
|---|---|---|
| D66 | 01000010 | |
| D194 | 01000011 | |
| D196 | 00100011 | (D•G'•H'+D'•G)•B⊕C•A'•E'•F' |
| D68 | 00100010 | |
| D10 | 01010000 | |
| D12 | 00110000 | |
| D132 | 00100001 | |
| D133 | 10100001 | (A'+B')•C•D'•E'•F'•G'•H |
| D134 | 01100001 | |
| D24 | 00011000 | |
| D40 | 00010100 | |
| D72 | 00010010 | |
| D88 | 00011010 | |
| D104 | 00010110 | |
| D120 | 00011110 | |
| D136 | 00010001 | (E'•F'•G•H+E⊕F+G⊕H)•A'•B'•C'•D |
| D152 | 00011001 | |
| D168 | 00010101 | |
| D184 | 00011101 | |
| D200 | 00010011 | |
| D216 | 00011011 | |
| D232 | 00010111 | |
| D145 | 10001001 | |
| D146 | 01001001 | |
| D148 | 00101001 | |
| D137 | 10010001 | |
| D138 | 01010001 | |
| D140 | 00110001 | |
| D17 | 10001000 | |
| D18 | 01001000 | |
| D20 | 00101000 | (D⊕E•F'•G'•H+E⊕F•D'•G'•H'+G⊕H•D'•E'•F)• |
| D33 | 10000100 | (A•B'•C'+A'•B•C'+A'•B'•C) |
| D34 | 01000100 | |
| D36 | 00100100 | |
| D97 | 10000110 | |
| D98 | 01000110 | |
| D100 | 00100110 | |
| D161 | 10000101 | |
| D162 | 01000101 | |
| D164 | 00100101 | |

**Table 21. Positive Required Disparity PDRE**

| Name | ABCDEFGH | Coding Label |
|---|---|---|
| C9 | 10010000 | |
| K81 | 10001010 | |
| K82 | 01001010 | (F'+H)•K (from Table 10) |
| K84 | 00101010 | |
| *K131* | *11000001* | |
| *K248* | *00011111* | |

The equation for the positive required entry disparity PDRE can be written as follows:

$$PDRE=(D\oplus E \bullet F'\bullet G'\bullet H + E\oplus F \bullet D'\bullet G'\bullet H' + G\oplus H \bullet D'\bullet E'\bullet F) \bullet \qquad (n85)$$
$$(A\bullet B'\bullet C' + A'\bullet B\bullet C' + A'\bullet B'\bullet C) + (D\bullet G'\bullet H' + D'\bullet G) \bullet B\oplus C\bullet A'\bullet E'\bullet F' + \qquad (n46+Pn88^*)$$
$$(E'\bullet F'\bullet G\bullet H + E\oplus F + G\oplus H) \bullet A'\bullet B'\bullet C'\bullet D + \qquad (Pn87^*)$$
$$(A'+B') \bullet C\bullet D'\bullet E'\bullet F'\bullet G'\bullet H + (F' + H) \bullet K \qquad (n99)$$

**2)Negative Required Disparity for Encoding: NDRE**

A total of 43 vectors listed in the Tables 4 and 6 require a negative entry disparity (NDRE). They are listed and sorted in Table 22.

**Table 22. Negative Required Disparity NDRE**

| Name | ABCDEFGH | Coding Label |
|---|---|---|
| D199 | 11100011 | |
| D215 | 11101011 | (E'+F')•A•B•C•D'•G•H |
| D231 | 11100111 | |
| D119 | 11101110 | |
| D123 | 11011110 | (C⊕D•H'+D'•H)•A•B•E•F•G |
| *D243 | 11001111 | |
| D247 | 11101111 | |
| D121 | 10011110 | |
| D122 | 01011110 | A⊕B•C'•D•E•F•G |
| D249 | 10011111 | |
| D250 | 01011111 | |
| D87 | 11101010 | |
| D103 | 11100110 | |
| D151 | 11101001 | |
| D167 | 11100101 | (E⊕F•G'•H'+G⊕H•E'•F'+E⊕F•G⊕H)•A•B•C•D' |
| D23 | 11101000 | |
| D39 | 11100100 | |
| D71 | 11100010 | |
| D135 | 11100001 | |
| *K124* | *00111110* | K•C•D (from Table 10) |

**Table 22. Negative Required Disparity NDRE**

| Name | ABCDEFGH | Coding Label |
|---|---|---|
| D235 | 11010**111** | |
| D237 | 10110**111** | |
| D238 | 01110**111** | |
| D243 | 11001**111** | |
| D245 | 10101**111** | |
| D246 | 01101**111** | |
| D91 | 1101**1**01**0** | |
| D93 | 1011**1**01**0** | |
| D94 | 0111**1**01**0** | |
| D115 | 1100**1**11**0** | |
| D117 | 1010**1**11**0** | |
| D118 | 0110**1**11**0** | (D⊕E•F•G•H+D⊕F•E•G•H’+D•E•F’•H+D•E’•F•G)• |
| D155 | 110**110**0**1** | (A•B•C’+A•B’•C+A’•B•C) |
| D157 | 101**110**0**1** | |
| D158 | 011**110**0**1** | |
| D219 | 110**110**1**1** | |
| D221 | 101**110**1**1** | |
| D222 | 011**110**1**1** | |
| *D235 | 110**1011**1 | |
| *D237 | 101**1011**1 | |
| *D238 | 011**1011**1 | |
| D107 | 110**1011**0 | |
| D109 | 101**1011**0 | |
| D110 | 011**1011**0 | |
| D183 | **1**110**1101** | |
| D187 | **1**101**1101** | (B•C•D’+B•C’•D+B’•C•D)•A•E•F•G’•H |
| D189 | **1**011**1101** | |

The equation for the negative required entry disparity NDRE can be written as follows:

$NDRE=(D \oplus E \bullet F \bullet G \bullet H + D \oplus F \bullet E \bullet G \bullet H' + D \bullet E \bullet F' \bullet H + D \bullet E' \bullet F \bullet G) \bullet$ (n90)
$(A \bullet B \bullet C' + A \bullet B' \bullet C + A' \bullet B \bullet C) +$ (n48)
$(B \bullet C \bullet D' + B \bullet C' \bullet D + B' \bullet C \bullet D) \bullet A \bullet E \bullet F \bullet G' \bullet H +$ (n96*)
$(E' + F') \bullet A \bullet B \bullet C \bullet D' \bullet G \bullet H +$ (n92*)
$(E \oplus F \bullet G' \bullet H' + G \oplus H \bullet E' \bullet F' + E \oplus F \bullet G \oplus H) \bullet A \bullet B \bullet C \bullet D'$ (Pn94*)
$(C \oplus D \bullet H' + D' \bullet H) \bullet A \bullet B \bullet E \bullet F \bullet G +$ (Pn95*)
$A \oplus B \bullet C' \bullet D \bullet E \bullet F \bullet G + K \bullet C \bullet D +$ (Pn97*)

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

If the required entry disparity PDRE or NDRE does not match the running disparity RD, the alternate vector must be used. The alternate vector is generated by complementation of the primary vector. The running disparity at the vector boundaries is constrained to the two values plus or minus two. The positive or negative running disparity in front of a byte is referred to as PDFBY or NDFBY, respectively.

*CMPLP10 = PDRE•NDFBY + NDRE•PDFBY*

### 4) Equations for the Encoding of Block Disparity DB

*a) Positive Block Disparity of Four for Encoding: PDB*

A total of 19 vectors listed in the Table 6 have a positive disparity of four. They are listed and sorted in Table 23.

**Table 23. Positive Block Disparity PDB**

| Name | ABCDEFGH | Coding Label |
|---|---|---|
| D235 | 11010111 | (D⊕E•F+D•E•F')•G•H•(A•B•C'+A•B'•C+A'•B•C) |
| D237 | 10110111 | |
| D238 | 01110111 | |
| D243 | 11001111 | |
| D245 | 10101111 | |
| D246 | 01101111 | |
| D219 | 11011011 | |
| D221 | 10111011 | |
| D222 | 01111011 | |
| D183 | 11101101 | (B•C•D'+B•C'•D+B'•C•D)•A•E•F•G'•H |
| D187 | 11011101 | |
| D189 | 10111101 | |
| D215 | 11101011 | E⊕F•A•B•C•D'•G•H |
| D231 | 11100111 | |
| D119 | 11101110 | (C⊕D•H'+D'•H)•A•B•E•F•G |
| D123 | 11011110 | |
| *D243 | 11001111 | |
| D247 | 11101111 | |
| D249 | 10011111 | A⊕B•C'•D•E•F•G•H |
| D250 | 01011111 | |

The equation for the positive block disparity of four for encoding PDB can be written as follows:

$$PDB = (D \oplus E \bullet F + D \bullet E \bullet F') \bullet (A \bullet B \bullet C' + A \bullet B' \bullet C + A' \bullet B \bullet C) \bullet G \bullet H + \qquad \text{(PDB1)}$$
$$(B \bullet C \bullet D' + B \bullet C' \bullet D + B' \bullet C \bullet D) \bullet A \bullet E \bullet F \bullet G' \bullet H + \qquad \text{(PDB2)}$$
$$(C \oplus D \bullet H' + D' \bullet H) \bullet A \bullet B \bullet E \bullet F \bullet G + \qquad \text{(PDB3)}$$
$$E \oplus F \bullet A \bullet B \bullet C \bullet D' \bullet G \bullet H + A \oplus B \bullet C' \bullet D \bullet E \bullet F \bullet G \bullet H \qquad \text{(PDB4+PDB5)}$$

*b) Negative Block Disparity of Four for Encoding: NDB*

A total of 41 vectors listed in the Tables 7 and 8 have a negative disparity four. They are listed and sorted in Table 24.

Using the Coding Labels from the table, the equation for the negative block disparity of four for encoding NDB can be written as follows:

$$NDB = \{E \oplus F \bullet (G \bullet H)' + E' \bullet F' \bullet (G+H)\} \bullet A' \bullet B' \bullet C' \bullet D + \qquad \text{(NDB1)}$$
$$(D \oplus E \bullet F' \bullet G' \bullet H + E \oplus F \bullet D' \bullet G' \bullet H' + G \oplus H \bullet D' \bullet E' \bullet F) \bullet (A \bullet B' \bullet C' + A' \bullet B \bullet C' + A' \bullet B' \bullet C) \qquad \text{(NDB2)}$$
$$+ (D \bullet G' \bullet H' + D' \bullet G) \bullet B \oplus C \bullet A' \bullet E' \bullet F' + (A' + B') \bullet C \bullet D' \bullet E' \bullet F' \bullet G' \bullet H + K \bullet F' \qquad \text{(NDB4+NDB3)}$$

**5) Equation for Running Disparity at the End of the Byte**

For balanced vectors (BALBY), the starting and ending disparities are equal and complementary otherwise. Since for this code, the coded vectors are either balanced or have a disparity of plus or minus four, a vector is balanced, if neither PDB nor NDB is asserted. This approach results in less logic delay and significant logic circuit sharing compared to other possible solutions. The running disparity DEBY at the end of an encoded vector is determined as follows:

$$DEBY = (PDB + NDB)' \oplus DFBY$$

The running disparity DFBY at the start of the next byte is equal to the ending disparity DEBY of the preceding byte.

$$DFBY_{+1} = DEBY_0$$

The encoding circuitry includes a single latch (not shown) to keep track of the value of DFBY.

**Table 24. Negative Block Disparity NDB**

| Name | ABCDEFGH | Coding Label |
|---|---|---|
| D66 | 01000010 | |
| D194 | 01000011 | |
| D196 | 00100011 | (D•G’•H’+D’•G)•B⊕C•A’•E’•F’ |
| D68 | 00100010 | |
| D10 | 01010000 | |
| D12 | 00110000 | |
| D132 | 00100001 | |
| D133 | 10100001 | (A’+B’)•C•D’•E’•F’•G’•H |
| D134 | 01100001 | |
| D24 | 00011000 | |
| D40 | 00010100 | |
| D88 | 00011010 | |
| D104 | 00010110 | {E⊕F•(G•H)’+E’•F’•(G+H)}•A’•B’•C’•D |
| D152 | 00011001 | |
| D168 | 00010101 | |
| D72 | 00010010 | |
| D136 | 00010001 | |
| D200 | 00010011 | |
| D145 | 10001001 | |
| D146 | 01001001 | |
| D148 | 00101001 | |
| D137 | 10010001 | |
| D138 | 01010001 | |
| D140 | 00110001 | |
| D17 | 10001000 | |
| D18 | 01001000 | |
| D20 | 00101000 | (D⊕E•F‘•G’•H+E⊕F•D’•G’•H’+G⊕H•D’•E’•F)• |
| D33 | 10000100 | (A•B’•C’•+A’•B•C’+A’•B’•C) |
| D34 | 01000100 | |
| D36 | 00100100 | |
| D97 | 10000110 | |
| D98 | 01000110 | |
| D100 | 00100110 | |
| D161 | 10000101 | |
| D162 | 01000101 | |
| D164 | 00100101 | |
| C9 | 10010000 | |
| K81 | 10001010 | |
| K82 | 01001010 | K•F’ (from Table 10) |
| K84 | 00101010 | |
| K131 | 11000001 | |

## G) Receiver Circuits

### 1) Validity Checks at Receiver

As mentioned in the introduction, any odd number of errors within a byte produces an invalid byte. A full ten-bit vector set includes among others 252 balanced vectors, 120 vectors with a disparity of plus four and 120 vectors with a disparity of minus four. The 8B10B-P code uses 352 vectors, 232 balanced vectors and 60 complementary pairs of vectors with a disparity of four. All other 672 ten-bit vectors are invalid. The validity checks can be executed by circuits which identify either valid vectors or invalid vectors. The approach below identifies all valid vectors which are listed and sorted for easy identification in two tables below.

Note that every valid vector has a complement which is also valid and the respective vectors are listed side by side.

### 2) Valid Vectors ending with i≠j

All but four of the vectors with i≠j, can be paired with another valid vector which is identical in the first eight bits. The exception is illustrated in FIG. 12 and in the two rows of Table 25 with empty spaces in two i≠j columns. Table 25 lists all 204 valid vectors with i≠j:

- Primary vectors:
  60 from Table 3, 4 from Table 4, 4 from Table 5,
  18 from Table 6, 16 from Table 8, 60 from Table 9.
- The Alternate vectors are complements of vectors listed as primary vectors:
  4 from Table 4, 4 from Table 5,
  18 from Table 6, 16 from Table 8.

**Table 25. 204 Valid Vectors with i≠j**

| Name ij=10 | Name ij=01 | abcde f gh | Name ij=10 | Name ij=01 | abcde f gh | Valid Label |
|---|---|---|---|---|---|---|
| D12A | D243P | 110**01111** | D243A | D12P | 001**10000** | (a⊕b'•b⊕c•c⊕d'+a⊕b•c⊕d)• |
| D10A | D245P | 101**01111** | D245A | D10P | 010**10000** | d⊕e•e⊕f'•f⊕g'•g⊕h'•i⊕j |
| C9A | D246P | 011**01111** | D246A | C9P | 100**10000** | |
| D72A | D183P | **1**110**1101** | D183A | D72P | **0**001**0010** | (a⊕b•b⊕c•c⊕d'+a⊕b'•c⊕d)• |
| D68A | D187P | **1**101**1101** | D187A | D68P | **0**010**0010** | a⊕e'•e⊕f'•f⊕g•g⊕h•i⊕j |
| D66A | D189P | **1**011**1101** | D189A | D66P | **0**100**0010** | |
| D232A | D23P | **1110**1000 | D23A | D232P | **0001**0111 | |
| D216A | D39P | **1110**0100 | D39A | D216P | **0001**1011 | (d⊕h'•e⊕f•g⊕h'+d⊕e'•e⊕f'• |
| D184A | D71P | **1110**0010 | D71A | D184P | **0001**1101 | g⊕h)•a⊕b'•b⊕c'•c⊕d•i⊕j |
| D120A | D135P | **1110**0001 | D135A | D120P | **0001**1110 | |
| D136A | D119P | **11**101**110** | D119A | D136P | **00**010**001** | a⊕b'•b⊕g'•g⊕h• |
| D132A | D123P | **11**011**110** | D123A | D132P | **00**100**001** | c⊕d•e⊕f'•f⊕g'•i⊕j |
| | D249P | 10**011111** | D249A | | 01**100000** | a⊕b•d⊕e'•g⊕h'•d⊕i• |
| | D250P | 01**011111** | D250A | | 10**100000** | c⊕d•e⊕f'•f⊕g'•i⊕j |

**Table 25. 204 Valid Vectors with i≠j**

| Name ij=10 | Name ij=01 | abcdefgh | Name ij=10 | Name ij=01 | abcdefgh | Valid Label |
|---|---|---|---|---|---|---|
| D53 | D63 | 10101100 | D202 | D192 | 01010011 | (a⊕b'•b⊕c•c⊕d'+a⊕b•c⊕d)•(e⊕h•f⊕g+e⊕f•g⊕h)•i⊕j |
| D54 | D55 | 01101100 | D201 | D9 | 10010011 | |
| D57 | D48 | 10011100 | D198 | D207 | 01100011 | |
| D58 | D59 | 01011100 | D197 | D5 | 10100011 | |
| D51 | D3 | 11001100 | D204 | D252 | 00110011 | |
| D60 | D124 | 00111100 | D195 | D131 | 11000011 | |
| D85 | D65 | 10101010 | D170 | D190 | 01010101 | |
| D86 | D0 | 01101010 | D169 | D255 | 10010101 | |
| D89 | D56 | 10011010 | D166 | D6 | 01100101 | |
| D90 | D248 | 01011010 | D165 | D7 | 10100101 | |
| D83 | D80 | 11001010 | D172 | D175 | 00110101 | |
| D92 | D95 | 00111010 | D163 | D160 | 11000101 | |
| D101 | D125 | 10100110 | D154 | D130 | 01011001 | |
| D102 | D126 | 01100110 | D153 | D129 | 10011001 | |
| D105 | D1 | 10010110 | D150 | D254 | 01101001 | |
| D106 | D8 | 01010110 | D149 | D61 | 10101001 | |
| D99 | D96 | 11000110 | D156 | D159 | 00111001 | |
| D108 | D111 | 00110110 | D147 | D144 | 11001001 | |
| D113 | D253 | 10001110 | D142 | D2 | 01110001 | (a⊕b'•b⊕c•c⊕d+a⊕b•c⊕d')•(d⊕e•e⊕f'•f⊕g'•g⊕h)•i⊕j |
| D114 | D127 | 01001110 | D141 | D128 | 10110001 | |
| D116 | D112 | 00101110 | D139 | D143 | 11010001 | |
| D75 | D79 | 11010010 | D180 | D176 | 00101101 | (a⊕b'•b⊕c•c⊕d+a⊕b•c⊕d')•(d⊕f•e⊕g+e⊕f•g⊕h')•d⊕h•i⊕j |
| D77 | D64 | 10110010 | D178 | D191 | 01001101 | |
| D78 | D15 | 01110010 | D177 | D240 | 10001101 | |
| D27 | D31 | 11011000 | D228 | D224 | 00100111 | |
| D29 | D16 | 10111000 | D226 | D239 | 01000111 | |
| D30 | D62 | 01111000 | D225 | D193 | 10000111 | |
| D43 | D47 | 11010100 | D212 | D208 | 00101011 | |
| D45 | D32 | 10110100 | D210 | D223 | 01001011 | |
| D46 | D4 | 01110100 | D209 | D251 | 10001011 | |
| D40A | D215P | 11101011 | D215A | D40P | 00010100 | (a⊕b'•b⊕g'•c⊕d+a⊕b•c⊕d'•d⊕h')•e⊕f•g⊕h'•i⊕j |
| D36A | D219P | 11011011 | D219A | D36P | 00100100 | |
| D34A | D221P | 10111011 | D221A | D34P | 01000100 | |
| D33A | D222P | 01111011 | D222A | D33P | 10000100 | |
| D24A | D231P | 11100111 | D231A | D24P | 00011000 | |
| D20A | D235P | 11010111 | D235A | D20P | 00101000 | |
| D18A | D237P | 10110111 | D237A | D18P | 01001000 | |
| D17A | D238P | 01110111 | D238A | D17P | 10001000 | |

**3) Valid Vectors ending with i=j**

The 148 vectors with i=j are listed in Table 26 arranged as 74 complementary vector pairs:

- Primary vectors: 27 complementary primary vector pairs from Table 1 and 2
  20 primary vectors from Table 4
  1 primary vector each from Table 5 and 6
  25 primary vectors from Table 7
- Alternate vectors: 20 from Table 4
  1 each from Table 5 and 6
  25 from Table 7

**4) Validity Equation**

The equation for the validity of encoded vectors is composed from the Valid Labels of Tables 25 and 26. The expressions in parentheses at the right page edge refer to the net name of the logic circuit associated with the logic expression at the left.

$$
\begin{aligned}
&VALID= \{(e\oplus f'\bullet f\oplus g'\bullet g\oplus h\bullet h\oplus i + e\oplus f'\bullet f\oplus i\bullet g\oplus h + e\oplus f\bullet g\oplus h'\bullet h\oplus i) \bullet i\oplus j' + && (n0\bullet i\oplus j')\\
&\quad (d\oplus e\bullet e\oplus f'\bullet f\oplus g'\bullet g\oplus h' + e\oplus h\bullet f\oplus g + e\oplus f\bullet g\oplus h) \bullet i\oplus j\} \bullet && (n1\bullet i\oplus j)\\
&\qquad (a\oplus b'\bullet b\oplus c\bullet c\oplus d' + a\oplus b\bullet c\oplus d) + && (n4)\\
&\quad \{(d\oplus e\bullet e\oplus f'\bullet f\oplus g'\bullet g\oplus h + d\oplus h\bullet d\oplus f\bullet e\oplus g + d\oplus h\bullet e\oplus f\bullet g\oplus h') \bullet i\oplus j + && (n6\bullet i\oplus j)\\
&\qquad (e\oplus f'\bullet f\oplus g\bullet g\oplus h'\bullet h\oplus i + e\oplus f\bullet g\oplus h) \bullet d\oplus i\bullet i\oplus j'\} \bullet && (n8\bullet d\oplus i\bullet i\oplus j')\\
&\qquad (a\oplus b'\bullet b\oplus c\bullet c\oplus d + a\oplus b\bullet c\oplus d') + && (n11)\\
&\quad (e\oplus f'\bullet f\oplus g'\bullet g\oplus h'\bullet h\oplus i+e\oplus f\bullet g\oplus h)\bullet (a\oplus b\bullet c\oplus d'\bullet d\oplus i'+a\oplus b'\bullet b\oplus i'\bullet c\oplus d)\bullet i\oplus j' + && (Pn27)\\
&\quad (a\oplus b\bullet d\oplus e'\bullet g\oplus h'\bullet d\oplus i + a\oplus b'\bullet b\oplus g'\bullet g\oplus h) \bullet c\oplus d\bullet e\oplus f'\bullet f\oplus g'\bullet i\oplus j + && (Pn15, Pn29)\\
&\quad (b\oplus c'\bullet c\oplus d\bullet d\oplus e + b\oplus c\bullet d\oplus e') \bullet a\oplus e'\bullet e\oplus f'\bullet f\oplus g\bullet g\oplus h'\bullet h\oplus i\bullet i\oplus j' + && (Pn14)\\
&\quad (a\oplus b'\bullet b\oplus g'\bullet c\oplus d + a\oplus b\bullet c\oplus d'\bullet d\oplus h') \bullet e\oplus f\bullet g\oplus h'\bullet i\oplus j + && (Pn28)\\
&\quad (a\oplus b\bullet b\oplus c\bullet c\oplus d' + a\oplus b'\bullet c\oplus d) \bullet a\oplus e'\bullet e\oplus f'\bullet f\oplus g\bullet g\oplus h\bullet i\oplus j + && (Pn30)\\
&\quad (e\oplus f'\bullet g\oplus h'\bullet h\oplus i + e\oplus f\bullet g\oplus h) \bullet a\oplus b'\bullet b\oplus c'\bullet c\oplus d\bullet d\oplus i'\bullet i\oplus j' + && (Pn17)\\
&\quad (d\oplus e'\bullet e\oplus f'\bullet g\oplus h + d\oplus h'\bullet e\oplus f\bullet g\oplus h') \bullet a\oplus b'\bullet b\oplus c'\bullet c\oplus d\bullet i\oplus j + && (Pn31)
\end{aligned}
$$

The expression $(a\oplus b'\bullet b\oplus c\bullet c\oplus d' + a\oplus b\bullet c\oplus d)$ in n4 above can be simplified to $(a\oplus d\bullet b\oplus c + a\oplus b\bullet c\oplus d)$ but this requires an extra XNOR gate for the implementation shown in this report.

**Table 26. 148 Valid Vectors with i=j**

| Name | abcdefghi j | Name | abcdefghi j | Valid Label |
|---|---|---|---|---|
| D181 | 1010110100 | D74 | 0101001011 | |
| D182 | 0110110100 | D73 | 1001001011 | |
| D185 | 1001110100 | D70 | 0110001011 | |
| D186 | 0101110100 | D69 | 1010001011 | |
| D179 | 1100110100 | D76 | 0011001011 | |
| D188 | 0011110100 | D67 | 1100001011 | |
| D117P | 1010111000 | D117A | 0101000111 | |
| D118P | 0110111000 | D118A | 1001000111 | |
| D121P | 1001111000 | D121A | 0110000111 | |
| D122P | 0101111000 | D122A | 1010000111 | |
| D115P | 1100111000 | D115A | 0011000111 | |
| *K124P* | *0011111000* | *K124A* | *1100000111* | (a⊕b'•b⊕c•c⊕d'+a⊕b•c⊕d)• |
| D213 | 1010101100 | D42 | 0101010011 | (e⊕f'•f⊕i•g⊕h+e⊕f•g⊕h'•h⊕i)•i⊕j' |
| D214 | 0110101100 | D41 | 1001010011 | |
| D217 | 1001101100 | D38 | 0110010011 | |
| D218 | 0101101100 | D37 | 1010010011 | |
| D211 | 1100101100 | D44 | 0011010011 | |
| D220 | 0011101100 | D35 | 1100010011 | |
| D229 | 1010011100 | D26 | 0101100011 | |
| D230 | 0110011100 | D25 | 1001100011 | |
| D233 | 1001011100 | D22 | 0110100011 | |
| D234 | 0101011100 | D21 | 1010100011 | |
| D227 | 1100011100 | D28 | 0011100011 | |
| D236 | 0011011100 | D19 | 1100100011 | |
| D91P | 1101101000 | D91A | 0010010111 | |
| D93P | 1011101000 | D93A | 0100010111 | |
| D94P | 0111101000 | D94A | 1000010111 | |
| D155P | 1101100100 | D155A | 0010011011 | |
| D157P | 1011100100 | D157A | 0100011011 | |
| D158P | 0111100100 | D158A | 1000011011 | |
| D107P | 1101011000 | D107A | 0010100111 | (a⊕b'•b⊕c•c⊕d+a⊕b•c⊕d')•d⊕i•i⊕j'• |
| D109P | 1011011000 | D109A | 0100100111 | (e⊕f'•f⊕g•g⊕h'•h⊕i+e⊕f•g⊕h) |
| D110P | 0111011000 | D110A | 1000100111 | |
| D171 | 1101010100 | D84 | 0010101011 | |
| D173 | 1011010100 | D82 | 0100101011 | |
| D174 | 0111010100 | D81 | 1000101011 | |
| D203 | 1101001100 | D52 | 0010110011 | |
| D205 | 1011001100 | D50 | 0100110011 | |
| D206 | 0111001100 | D49 | 1000110011 | |

**Table 26. 148 Valid Vectors with i=j**

| Name | abcdefghi j | Name | abcdefghi j | Valid Label |
|---|---|---|---|---|
| K81P | 1000101000 | K81A | 0111010111 | |
| K82P | 0100101000 | K82A | 1011010111 | |
| K84P | 0010101000 | K84A | 1101010111 | |
| D88P | 0001101000 | D88A | 1110010111 | |
| D145P | 1000100100 | D145A | 0111011011 | |
| D146P | 0100100100 | D146A | 1011011011 | |
| D148P | 0010100100 | D148A | 1101011011 | |
| D152P | 0001100100 | D152A | 1110011011 | (a⊕b•c⊕d'•d⊕i'+a⊕b'•b⊕i'•c⊕d)• e⊕f•g⊕h•i⊕j' |
| D161P | 1000010100 | D161A | 0111101011 | |
| D162P | 0100010100 | D162A | 1011101011 | |
| D164P | 0010010100 | D164A | 1101101011 | |
| D168P | 0001010100 | D168A | 1110101011 | |
| D97P | 1000011000 | D97A | 0111100111 | |
| D98P | 0100011000 | D98A | 1011100111 | |
| D100P | 0010011000 | D100A | 1101100111 | |
| D104P | 0001011000 | D104A | 1110100111 | |
| D241 | 1000111100 | D14 | 0111000011 | |
| D242 | 0100111100 | D13 | 1011000011 | (a⊕b•c⊕d'•d⊕i'+a⊕b'•b⊕i'•c⊕d)• e⊕f'•f⊕g'•g⊕h'•h⊕i•i⊕j' |
| D244 | 0010111100 | D11 | 1101000011 | |
| K248P | 0001111100 | K248A | 1110000011 | |
| D194P | 0100001100 | D194A | 1011110011 | (b⊕c'•c⊕d•d⊕e+b⊕c•d⊕e')• a⊕e'•e⊕f'•f⊕g•g⊕h'•h⊕i•i⊕j' |
| D196P | 0010001100 | D196A | 1101110011 | |
| D200P | 0001001100 | D200A | 1110110011 | |
| D87P | 1110101000 | D87A | 0001010111 | |
| D151P | 1110100100 | D151A | 0001011011 | |
| D103P | 1110011000 | D103A | 0001100111 | (e⊕f'•g⊕h'•h⊕i+e⊕f•g⊕h)• a⊕b'•b⊕c'•c⊕d•d⊕i'•i⊕j' |
| D167P | 1110010100 | D167A | 0001101011 | |
| D199P | 1110001100 | D199A | 0001110011 | |
| D247P | 1110111100 | D247A | 0001000011 | |
| *K131P* | *1100000100* | *K131A* | *0011111011* | |
| D140P | 0011000100 | D140A | 1100111011 | |
| D133P | 1010000100 | D133A | 0101111011 | (a⊕b'•b⊕c•c⊕d'+a⊕b•c⊕d)• e⊕f'•f⊕g'•g⊕h•h⊕i•i⊕j' |
| D134P | 0110000100 | D134A | 1001111011 | |
| D137P | 1001000100 | D137A | 0110111011 | |
| D138P | 0101000100 | D138A | 1010111011 | |

Disparity violations are not monitored because they would in most cases not help significantly the error correction procedures associated with this type of code and of course, the results of such disparity checks are usually not available until a few bytes after the error.

## H) Decoding

Decoding restores the original eight bits and the K-bit. As for encoding, there are two types of bit changes to be made.

1. Complementation of an entire vector.
2. Complementation of individual bits.

The code was built such that these two operations can be totally separated and can be executed in parallel. The two extra bits i and j are included to select the vectors for the above operations and then simply dropped.

### 1) Circuit Simplification

The decoding equations can be significantly 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 overlined and eliminated in the final equations for the complementation of an entire vector or the complementation of individual bits. As a first example, the bit values ‘a’ and ‘b’ of a pair vectors might be 10 and 01, respectively. These bit values can be ignored for purposes of the logic expression, since the only possible other values are 00 or 11, both of which generate an invalid vector because the Huffman distance between vector classes is two. Of course only one such complementary pair if bits can be eliminated for each pair of vectors. In this context it is also useful to remember that the maximum run length is five and the runs are at most three at the leading and trailing ends of the coded 10-bit vectors and these second type violations can be included together with the first type of violations.

### 2) Complementation of entire Vectors

All disparity dependent code points have complementary representations, a primary vector and an alternate vector, identified in the tables by an appended letter P or A to the vector name, respectively. The primary or alternate versions are used to meet the disparity requirements. For decoding, all alternate vectors must be complemented. The 89 alternate vectors are the complements of the vectors listed in Tables 4 through 8 and are tabulated in Table 27.

The equation for the complementation of the first eight bits of a vector is composed from the Alternate Vector Labels of Table 27.

$$COMPL8 = (a \oplus b' \bullet b \oplus g' + c \oplus d' \bullet d \oplus h') \bullet e \oplus f \bullet g \oplus h' \bullet i \bullet j' + \quad \text{(n53)}$$
$$(d \oplus h' \bullet g \oplus h' + d \oplus e' \bullet e \oplus f') \bullet a \oplus b' \bullet b \oplus c' \bullet i \bullet j' + \quad \text{(n52)}$$
$$(a \oplus b \bullet b \oplus c \bullet f \oplus g + a \oplus b' \bullet f \oplus g + a \oplus b' \bullet f \oplus g') \bullet a \oplus e' \bullet e \oplus f' \bullet g \oplus h \bullet i \bullet j' + \quad \text{(n51)}$$
$$(d \oplus e \bullet g \oplus h' \bullet j' + g \oplus h \bullet j) \bullet (a \oplus b' \bullet c \oplus d' + c \oplus d) \bullet e \oplus f' \bullet f \oplus g' \bullet i + \quad \text{(n50)}$$
$$(b \oplus c \bullet a' + a \bullet b' \bullet c') \bullet d' \bullet e' \bullet f \bullet i \bullet j + (\overline{e} \bullet f \bullet g' \bullet h' + g \oplus h) \bullet a' \bullet b' \bullet c' \bullet i \bullet j + \quad \text{(n56+Pn59)}$$
$$(a' \bullet b' + c' \bullet d') \bullet e \bullet f' \bullet h \bullet i \bullet j + e' \bullet f' \bullet g' \bullet h' \bullet c \bullet d' \bullet j' + \quad \text{(Pn41+Pn42)}$$
$$\{a \oplus b' \bullet b \oplus c' \bullet e' \bullet f' + a \bullet e \bullet f \bullet (b+d)\} \bullet g' \bullet h' \bullet i \bullet j + \quad \text{(n58)}$$
$$(a \bullet b + c \bullet d) \bullet e \oplus f \bullet g \oplus h \bullet i \bullet j \quad \text{(Pn57)}$$

Table 27. 89 **Alternate Vectors**

| Name | abcdefghi j | Name | abcdefghi j | Alternate Vector Label |
|---|---|---|---|---|
| D40A | 111010**1110** | D215A | 000101**0010** | $(a\oplus b'\bullet b\oplus g'\bullet\overline{c\oplus d}+\overline{a\oplus b}\bullet c\oplus d'\bullet d\oplus h')\bullet e\oplus f\bullet g\oplus h'\bullet i\bullet j'$ |
| D36A | 110110**1110** | D219A | 001001**0010** | |
| D34A | 101110**1110** | D221A | 010001**0010** | |
| D33A | 011110**1110** | D222A | 100001**0010** | |
| D24A | 111001**1110** | D231A | 000110**0010** | |
| D20A | 110101**1110** | D235A | 001010**0010** | |
| D18A | 101101**1110** | D237A | 010010**0010** | |
| D17A | 011101**1110** | D238A | 100010**0010** | |
| D232A | **1110**1000**10** | D23A | **0001**0111**10** | $(d\oplus h'\bullet\overline{e\oplus f}\bullet g\oplus h'+d\oplus e'\bullet e\oplus f'\bullet\overline{g\oplus h})\bullet a\oplus b'\bullet b\oplus c'\bullet\overline{c\oplus d}\bullet i\bullet j'$ |
| D216A | **1110**0100**10** | D39A | **0001**1011**10** | |
| D184A | **1110**0010**10** | D71A | **0001**1101**10** | |
| D120A | **1110**000110 | D135A | **0001**1110**10** | |
| D72A | 1110**110110** | D183A | **0**001**001010** | $(a\oplus b\bullet b\oplus c\bullet\overline{c\oplus d}'+a\oplus b'\bullet\overline{c\oplus d})\bullet a\oplus e'\bullet e\oplus f'\bullet f\oplus g\bullet g\oplus h\bullet i\bullet j'$ |
| D68A | 1101**110110** | D187A | **0**010**001010** | |
| D66A | 1011**110110** | D189A | **0**100**001010** | |
| D136A | **1110111010** | D119A | **0001000110** | $a\oplus b'\bullet\overline{c\oplus d}\bullet a\oplus e'\bullet e\oplus f'\bullet f\oplus g'\bullet g\oplus h\bullet i\bullet j'$ |
| D132A | **1101111010** | D123A | **0010000110** | |
| D12A | 110**0111110** | D243A | 001**1000010** | $(a\oplus b'\bullet\overline{b\oplus c}\bullet c\oplus d'+\overline{a\oplus b}\bullet c\oplus d)\bullet d\oplus e\bullet e\oplus f'\bullet f\oplus g'\bullet g\oplus h'\bullet i\bullet j'$ |
| D10A | 101**0111110** | D245A | 010**1000010** | |
| C9A | 011**0111110** | D246A | 100**1000010** | |
| *K131A* | 0011**111011** | D115A | 0011**000111** | $(a\oplus b'\bullet\overline{b\oplus c}\bullet c\oplus d'+\overline{a\oplus b}\bullet c\oplus d)\bullet e\oplus f'\bullet f\oplus g'\bullet g\oplus h\bullet i\bullet j$ |
| D140A | 1100**111011** | *K124A* | 1100**000111** | |
| D133A | 0101**111011** | D117A | 0101**000111** | |
| D134A | 1001**111011** | D118A | 1001**000111** | |
| D137A | 0110**111011** | D121A | 0110**000111** | |
| D138A | 1010**111011** | D122A | 1010**000111** | |
| | | D91A | 001**001**01**11** | $(b\oplus c\bullet a'+a\bullet b'\bullet c')\bullet\overline{g\oplus h}\bullet d'\bullet e'\bullet f\bullet i\bullet j$ |
| | | D93A | 010**001**01**11** | |
| | | D94A | 100**001**01**11** | |
| | | D155A | 001**001**10**11** | |
| | | D157A | 010**001**10**11** | |
| | | D158A | 100**001**10**11** | |
| | | D151A | **0001**0110**11** | $(\overline{e\oplus f}\bullet g\oplus h+\overline{e}\bullet f\bullet g'\bullet h')\bullet a'\bullet b'\bullet c'\bullet\overline{d}\bullet i\bullet j$ |
| | | D167A | **0001**1010**11** | |
| | | D87A | **0001**0101**11** | |
| | | *D103A | **0001**1001**11** | |
| | | D199A | **0001**1100**11** | |
| | | D249A | 01**10000010** | $\overline{a\oplus b}\bullet c\bullet d'\bullet e'\bullet f'\bullet g'\bullet h'\bullet\overline{i}\bullet j'$ |
| | | D250A | 10**10000010** | |

**Table 27. 89 Alternate Vectors**

| Name | abcdefghi j | Name | abcdefghi j | Alternate Vector Label |
|---|---|---|---|---|
| | | D103A | 0001**100111** | |
| | | D107A | 0010**100111** | $(\overline{c\oplus d}\bullet a'\bullet b'+\overline{a\oplus b}\bullet c'\bullet d')\bullet e\bullet f'\bullet \overline{g'}\bullet h\bullet i\bullet j$ |
| | | D109A | 0100**100111** | |
| | | D110A | 1000**100111** | |
| | | D247A | 0001**000011** | |
| | | K248A | 1110**000011** | $a\oplus b'\bullet b\oplus c'\bullet \overline{c\oplus d}\bullet e'\bullet f'\bullet g'\bullet h'\bullet i\bullet j$ |
| | | D194A | **1011110011** | |
| | | D196A | **1101110011** | $(\overline{b\oplus c}\bullet d+b\bullet \overline{c\bullet d'})\bullet a\bullet e\bullet f\bullet g'\bullet h'\bullet i\bullet j$ |
| | | D200A | **1110110011** | |
| | | K81A | 01110101**11** | |
| | | K82A | 10110101**11** | |
| | | K84A | 11010101**11** | |
| | | D88A | 11100101**11** | |
| | | D145A | 01110110**11** | |
| | | D146A | 10110110**11** | |
| | | D148A | 11010110**11** | |
| | | D152A | 11100110**11** | $(\overline{a\oplus b}\bullet c\bullet d+\overline{c\oplus d}\bullet a\bullet b)\bullet e\oplus f\bullet g\oplus h\bullet i\bullet j$ |
| | | D161A | 01111010**11** | |
| | | D162A | 10111010**11** | |
| | | D164A | 11011010**11** | |
| | | D168A | 11101010**11** | |
| | | D97A | 01111001**11** | |
| | | D98A | 10111001**11** | |
| | | D100A | 11011001**11** | |
| | | D104A | 11101001**11** | |

The factors $\overline{c\oplus d}$ in the second label and $\overline{d}$ in the eighth label are redundant because $c\oplus d'$ and $d'$ generate both an invalid leading run of four.

### 3) Complementation of selected individual Bits

For decoding, the bold type bit values in the encoded columns of Table 9 have to be complemented back to their original values as indicate in the source vector columns ABCDEFGH. The decoding equations are similar to the encoding equations, except that the values of the i and j bits have to be included and the bold type values in Table 9 are the complements of those used in the encoding equations. In the Tables 28 through 36, the common bit patterns are marked by bold type to logically classify the vectors by simple expressions. Redundant terms are overlined.

*a) Decoded Bit A*

*The ‘a’ column* has bold entries in the Tables 9 and 10 for the vectors listed in Table 28.

**Table 28. A-bit Decoding**

| Name | abcdefghi j | A | Name | abcdefghi j | A | Decoding Label |
|---|---|---|---|---|---|---|
| D80 | **1100**1010**01** | 0 | D175 | **0011**0101**01** | 1 | |
| D96 | **1100**0110**01** | 0 | D159 | **0011**1001**01** | 1 | $a\oplus b'\bullet\overline{b\oplus c}\bullet c\oplus d'\bullet\overline{e\oplus f}\bullet g\oplus h\bullet i'\bullet j$ |
| D144 | **1100**1001**01** | 0 | D111 | **0011**0110**01** | 1 | |
| D160 | **1100**0101**01** | 0 | D95 | **0011**1010**01** | 1 | |
| D128 | **101100**01**01** | 0 | D127 | **010011**10**01** | 1 | |
| D64 | **101100**10**01** | 0 | D191 | **010011**01**01** | 1 | $(d\oplus e\bullet e\oplus f'\bullet\overline{g\oplus h}+b\oplus g'\bullet\overline{e\oplus f}\bullet g\oplus h')$ |
| D32 | **1011**01**0001** | 0 | D223 | **0100**10**1101** | 1 | $\bullet\ a\oplus b\bullet b\oplus c\bullet c\oplus d'\bullet i'\bullet j$ |
| D16 | **1011**10**0001** | 0 | D239 | **0100**01**1101** | 1 | |
| D48 | **100111**00**01** | 0 | D207 | **0110001101** | 1 | |
| D240 | **100011**01**01** | 0 | D15 | **0111001001** | 1 | $a\oplus b\bullet a\oplus e'\bullet b\oplus c'\bullet\overline{d\oplus h}\bullet e\oplus f'\bullet f\oplus g\bullet i'\bullet j$ |
| | | | D55 | **01**10**110001** | 1 | |
| | | | D59 | **01**01**110001** | 1 | $\overline{c\oplus d}\bullet a'\bullet b\bullet e\bullet f\bullet g'\bullet h'\bullet i'\bullet\overline{j}$ |
| D56 | **1001101001** | 0 | | | | |
| *D64 | **1011001001** | 0 | | | | $\overline{c\oplus e}\bullet a\bullet b'\bullet d\bullet f'\bullet g\bullet h'\bullet i'\bullet j$ |

Using the Decoding Labels, the decoding equation for bit 'A' can be written as follows:

$$
\begin{aligned}
A = a\oplus\{&(d\oplus e\bullet e\oplus f' + b\oplus g'\bullet g\oplus h')\bullet a\oplus b\bullet b\oplus c\bullet c\oplus d'\bullet i'\bullet j\ + && \text{(n70)}\\
&a\oplus b\bullet b\oplus c'\bullet a\oplus e'\bullet e\oplus f'\bullet f\oplus g\bullet i'\bullet j + a\bullet b'\bullet d\bullet f'\bullet g\bullet h'\bullet i'\bullet j\ + && \text{(Pn66+Pn68)}\\
&a\oplus b'\bullet c\oplus d'\bullet g\oplus h\bullet i'\bullet j + a'\bullet b\bullet e\bullet f\bullet g'\bullet h'\bullet i'\bullet\overline{j}\} && \text{(Pn67+Pn69)}
\end{aligned}
$$

*b) Decoded Bit B*

*The 'b' column* has bold entries in the Tables 9 and 10 for the vectors listed in Table 29.

**Table 29. B-bit Decoding**

| Name | abcdefghi j | B | Name | abcdefghi j | B | Decoding Label |
|---|---|---|---|---|---|---|
| D80 | **1100**1010**01** | 0 | D175 | **0011**0101**01** | 1 | |
| D96 | **1100**0110**01** | 0 | D159 | **0011**1001**01** | 1 | $a\oplus b'\bullet\overline{b\oplus c}\bullet c\oplus d'\bullet\overline{e\oplus f}\bullet g\oplus h\bullet i'\bullet j$ |
| D144 | **1100**1001**01** | 0 | D111 | **0011**0110**01** | 1 | |
| D160 | **1100**0101**01** | 0 | D95 | **0011**1010**01** | 1 | |
| **D**248 | **01**01**101001** | 0 | D7 | **101**0**010101** | 1 | |
| D0 | **01**10**101001** | 0 | D255 | **1**00**1010101** | 1 | $a\oplus b\bullet b\oplus g'\bullet\overline{c\oplus d}\bullet e\oplus f\bullet f\oplus g\bullet g\oplus h\bullet i'\bullet j$ |
| D4 | **01**110100**01** | 0 | D251 | **1**0001011**01** | 1 | |
| *D0 | **01**101010**01** | 0 | *D255 | **1**0010101**01** | 1 | $(b\oplus c\bullet\overline{c\oplus d}\bullet e\oplus f'\bullet g\oplus h'+b\oplus c'\bullet e\oplus f\bullet$ |
| D192 | **01**010011**01** | 0 | D63 | **1**0101100**01** | 1 | $\overline{c\oplus h})\bullet a\oplus b\bullet d\oplus e\bullet f\oplus g\bullet i'\bullet j$ |
| D8 | **01010**110**01** | 0 | | | | |
| *D192 | **0101001**1**01** | 0 | | | | $\overline{f\oplus h}\bullet a'\bullet b\bullet c'\bullet d\bullet e'\bullet g\bullet i'\bullet j$ |

Using the Decoding Labels of Table 29, the decoding equation for bit 'B' can be written as follows:

$$B = b\oplus\{(b\oplus c\bullet e\oplus f'\bullet g\oplus h' + b\oplus c'\bullet e\oplus f) \bullet a\oplus b\bullet d\oplus e\bullet f\oplus g \bullet i'\bullet j + \qquad \text{(n72)}$$
$$a\oplus b\bullet b\oplus g'\bullet e\oplus f \bullet f\oplus g\bullet g\oplus h \bullet i'\bullet j + \qquad \text{(Pn74)}$$
$$a\oplus b'\bullet c\oplus d'\bullet g\oplus h \bullet i'\bullet j + a'\bullet b\bullet c' \bullet d\bullet e'\bullet g \bullet i'\bullet j\} \qquad \text{(Pn67+Pn75)}$$

*c) Decoded Bit C*

*The 'c' column* has bold entries in the Tables 9 and 10 for the vectors listed in Table 30.

**Table 30. C-bit Decoding**

| Name | abcdefghi j | C | Name | abcdefghi j | C | Decoding Label |
|---|---|---|---|---|---|---|
| D176 | 0010110101 | 0 | D79 | 1101001001 | 1 | $(d\oplus e\bullet e\oplus f'\bullet \overline{g\oplus h}+d\oplus h\bullet \overline{e\oplus f}\bullet g\oplus h')\bullet a\oplus d'\bullet b\oplus c\bullet i'\bullet j$ |
| D112 | 0010111001 | 0 | D143 | 1101000101 | 1 | |
| D128 | 1011000101 | 0 | D127 | 0100111001 | 1 | |
| D64 | 1011001001 | 0 | D191 | 0100110101 | 1 | |
| D224 | 0010011101 | 0 | D31 | 1101100001 | 1 | |
| D208 | 0010101101 | 0 | D47 | 1101010001 | 1 | |
| D32 | 1011010001 | 0 | D223 | 0100101101 | 1 | |
| D16 | 1011100001 | 0 | D239 | 0100011101 | 1 | |
| D0 | 0110101001 | 0 | D255 | 1001010101 | 1 | $\overline{a\oplus b}\bullet c\oplus d\bullet d\oplus e\bullet e\oplus f\bullet f\oplus g\bullet g\oplus h\bullet i'\bullet j$ |
| D65 | 1010101001 | 0 | D190 | 0101010101 | 1 | |
| D2 | 0111000101 | 0 | D253 | 1000111001 | 1 | $\overline{a\oplus b}\bullet c\oplus d'\bullet d\oplus e\bullet e\oplus f'\bullet f\oplus g'\bullet g\oplus h\bullet i'\bullet j$ |
| *D128 | 1011000101 | 0 | *D127 | 0100111001 | 1 | |

Using these Decoding Labels, the decoding equation for bit 'C' can be written as follows:

$$C = c\oplus\{(d\oplus e \bullet e\oplus f' + d\oplus h \bullet g\oplus h') \bullet a\oplus d' \bullet b\oplus c \bullet i'\bullet j + \qquad \text{(n77)}$$
$$(c\oplus d \bullet e\oplus f \bullet f\oplus g + c\oplus d' \bullet e\oplus f' \bullet f\oplus g') \bullet d\oplus e \bullet g\oplus h \bullet i'\bullet j\} \qquad \text{(n78)}$$

*d) Decoded Bit D*

*The 'd' column* has bold entries in the Tables 9 and 10 for the vectors listed in Table 31.

Using these Decoding Labels, the decoding equation for bit 'D' can be written as follows:

$$D = d\oplus\{(d\oplus e\bullet e\oplus f' + b\oplus g'\bullet g\oplus h') \bullet a\oplus b\bullet b\oplus c\bullet c\oplus d' \bullet i'\bullet j + \qquad \text{(n70)}$$
$$(d\oplus e'\bullet e\oplus f\bullet f\oplus g' + a\oplus e'\bullet e\oplus f'\bullet f\oplus g) \bullet a\oplus b\bullet c\oplus d \bullet i'\bullet j + \qquad \text{(n81)}$$
$$(b\oplus g\bullet f\oplus g' + c\oplus d'\bullet b\oplus g) \bullet a\oplus b\bullet b\oplus c'\bullet d\oplus e \bullet i'\bullet j + \qquad \text{(n83)}$$
$$a\bullet b'\bullet c\bullet d' \bullet e\bullet g' \bullet i'\bullet j\} \qquad \text{(n84)}$$

**Table 31. D-bit Decoding**

| Name | abcdefghi j | D | Name | abcdefghi j | D | Decoding Label |
|---|---|---|---|---|---|---|
| D128 | **1011**0001**01** | 0 | D127 | **0100**1110**01** | 1 | |
| D64 | **1011**0010**01** | 0 | D191 | **0100**1101**01** | 1 | $(d\oplus e \bullet e\oplus f' \bullet \overline{g\oplus h} + b\oplus g' \bullet g\oplus h' \bullet \overline{e\oplus f})\bullet$ |
| D32 | **1011**0100**01** | 0 | D223 | **0100**1011**01** | 1 | $a\oplus b \bullet b\oplus c \bullet c\oplus d' \bullet i' \bullet j$ |
| D16 | **1011**1000**01** | 0 | D239 | **0100**0111**01** | 1 | |
| D48 | 10**01**1100**01** | 0 | D207 | 01**10**0011**01** | 1 | |
| D192 | 01**01**0011**01** | 0 | D63 | 10**10**1100**01** | 1 | $a\oplus b \bullet a\oplus e' \bullet c\oplus d \bullet e\oplus f' \bullet f\oplus g \bullet \overline{g\oplus h'} \bullet i' \bullet j$ |
| D130 | 01**01100101** | 0 | D125 | 10**10011001** | 1 | |
| D129 | 10**01100101** | 0 | D126 | 01**10011001** | 1 | $a\oplus b \bullet c\oplus d \bullet d\oplus e' \bullet e\oplus f \bullet f\oplus g' \bullet \overline{g\oplus h} \bullet i' \bullet j$ |
| D2 | 011**10**001**01** | 0 | D253 | 100**01**110**01** | 1 | |
| D1 | 100**10**110**01** | 0 | D254 | 011**01**001**01** | 1 | $a\oplus b \bullet b\oplus c' \bullet b\oplus g \bullet d\oplus e \bullet f\oplus g' \bullet \overline{g\oplus h} \bullet i' \bullet j$ |
| D4 | **01110**1**0**0**01** | 0 | D251 | **1000**10**1**1**01** | 1 | |
| *D2 | **01110**0**0**1**01** | 0 | *D253 | **10001**1**1**0**01** | 1 | $a\oplus b \bullet b\oplus c' \bullet c\oplus d' \bullet d\oplus e \bullet b\oplus g \bullet \overline{f\oplus h} \bullet i' \bullet j$ |
| | | | D61 | **10101**0**0**1**01** | 1 | |
| | | | *D63 | **10101**1**0**0**01** | 1 | $\overline{f\oplus h} \bullet a \bullet b' \bullet c \bullet d' \bullet e \bullet g' \bullet i' \bullet j$ |

*e) Decoded Bit E*

*The 'e' column* has bold entries in the Tables 9 and 10 for the vectors listed in Table 32.

**Table 32. E-bit Decoding**

| Name | abcdefghi j | E | Name | abcdefghi j | E | Coding Label |
|---|---|---|---|---|---|---|
| D65 | 10**10101001** | 0 | D190 | 01**01010101** | 1 | |
| D0 | 01**10101001** | 0 | D255 | 10**01010101** | 1 | $\overline{a\oplus b} \bullet c\oplus d \bullet d\oplus e \bullet e\oplus f \bullet f\oplus g \bullet g\oplus h \bullet i' \bullet j$ |
| D130 | 01**01100101** | 0 | D125 | 10**10011001** | 1 | |
| D129 | 10**01100101** | 0 | D126 | 01**10011001** | 1 | $\overline{a\oplus b} \bullet c\oplus d \bullet d\oplus e' \bullet e\oplus f \bullet f\oplus g' \bullet g\oplus h \bullet i' \bullet j$ |
| D3 | 11001100**01** | 0 | D252 | 00110011**01** | 1 | $a\oplus b' \bullet \overline{b\oplus c} \bullet c\oplus d' \bullet d\oplus e \bullet$ $\overline{a\oplus e'} \bullet e\oplus f' \bullet f\oplus g \bullet \overline{g\oplus h'} \bullet i' \bullet j$ |

Using these Decoding Labels, the decoding equation for bit 'E' can be written as follows:

$$E = e \oplus \{(d\oplus e \bullet f\oplus g + \overline{d\oplus e'} \bullet f\oplus g') \bullet c\oplus d \bullet e\oplus f \bullet g\oplus h \bullet i' \bullet j + \quad \text{(n86)}$$
$$a\oplus b' \bullet \overline{b\oplus c} \bullet c\oplus d' \bullet \overline{a\oplus e'} \bullet e\oplus f' \bullet f\oplus g \bullet i' \bullet j\} \quad \text{(n87)}$$

The redundant factor $\overline{b\oplus c}$ is included to enable circuit sharing.

*f) Decoded Bit F*

*The 'f' column* has bold entries in the Tables 9 and 10 for the vectors listed in Table 33.

**Table 33. F-bit Decoding**

| Name | abcdefghi j | F | Name | abcdefghi j | F | Decoding Label |
|---|---|---|---|---|---|---|
| D193 | 1000011101 | 0 | D62 | 0111100001 | 1 | |
| D1 | 1001011001 | 0 | D254 | 0110100101 | 1 | $a\oplus b\bullet b\oplus c'\bullet b\oplus g\bullet e\oplus f\bullet f\oplus g'\bullet\overline{d\oplus h}\bullet i'\bullet j$ |
| D3 | 1100110001 | 0 | D252 | 0011001101 | 1 | |
| D4 | 0111010001 | 0 | D251 | 1000101101 | 1 | $a\oplus d\bullet b\oplus g\bullet c\oplus d'\bullet\overline{d\oplus e}\bullet f\oplus g\bullet g\oplus h'\bullet i'\bullet j$ |
| *D1 | 1001011001 | 0 | *D254 | 0110100101 | 1 | |
| D6 | 0110010101 | 0 | D56 | 1001101001 | 1 | $a\oplus b\bullet c\oplus d\bullet c\oplus g\bullet e\oplus f\bullet\overline{g\oplus h}\bullet i'\bullet j$ |
| D7 | 1010010101 | 0 | D248 | 0101101001 | 1 | |
| D8 | 0101011001 | 0 | D61 | 1010100101 | 1 | |

Using these Decoding Labels, the decoding equation for bit ‘F’ can be written as follows:

$$F=f\oplus\{a\oplus b\bullet b\oplus c'\bullet b\oplus g \bullet e\oplus f\bullet f\oplus g' \bullet i'\bullet j + \qquad \text{(n90)}$$
$$a\oplus d\bullet b\oplus g\bullet c\oplus d' \bullet f\oplus g\bullet g\oplus h' \bullet i'\bullet j + \qquad \text{(Pn91)}$$
$$a\oplus b\bullet c\oplus d\bullet c\oplus g\bullet e\oplus f \bullet i'\bullet j\} \qquad \text{(Pn92)}$$

*g) Decoded Bit G*

*The ‘g’ column* has bold entries in the Tables 9 and 10 for the vectors listed in Table 34.

**Table 34. G-bit Decoding**

| Name | abcdefghi j | G | Name | abcdefghi j | G | Decoding Label |
|---|---|---|---|---|---|---|
| D131 | 1100001101 | 0 | D124 | 0011110001 | 1 | |
| D15 | 0111001001 | 0 | D240 | 1000110101 | 1 | $\overline{a\oplus c}\bullet b\oplus g'\bullet c\oplus d'\bullet d\oplus h\bullet e\oplus f'\bullet f\oplus g\bullet i'\bullet j$ |
| D0 | 0110101001 | 0 | D255 | 1001010101 | 1 | |
| D1 | 1001011001 | 0 | D254 | 0110100101 | 1 | $\overline{a\oplus b}\bullet b\oplus c'\bullet c\oplus d\bullet d\oplus e\bullet e\oplus f\bullet g\oplus h\bullet i'\bullet j$ |
| D5 | 1010001101 | 0 | | | | |
| D9 | 1001001101 | 0 | | | | $\overline{c\oplus d}\bullet a\bullet b'\bullet e'\bullet f'\bullet g\bullet h\bullet i'\bullet j$ |
| D8 | 0101011001 | 0 | | | | |
| D56 | 1001101001 | 0 | | | | $\overline{a\oplus b}\bullet a\oplus e'\bullet e\oplus f\bullet c'\bullet d\bullet g\bullet h'\bullet i'\bullet j$ |

Using these Decoding Labels, the decoding equation for bit ‘G’ can be written as follows:

$$G = g\oplus\{b\oplus g'\bullet c\oplus d'\bullet d\oplus h \bullet e\oplus f'\bullet f\oplus g \bullet i'\bullet j + \qquad \text{(Pn94)}$$
$$b\oplus c'\bullet c\oplus d\bullet d\oplus e \bullet e\oplus f\bullet g\oplus h \bullet i'\bullet j + \qquad \text{(Pn95)}$$
$$a\oplus e'\bullet e\oplus f\bullet c'\bullet d \bullet g\bullet h'\bullet i'\bullet j + a\bullet b'\bullet e'\bullet f' \bullet g\bullet h\bullet i'\bullet j\} \qquad \text{(n96+Pn97)}$$

*h) Decoded Bit H*

*The ‘h’ column* has bold entries in the Tables 9 and 10 for the vectors listed in Table 35.

**Table 35. H-bit Decoding**

| Name | abcdefghi j | H | Name | abcdefghi j | H | Decoding Label |
|---|---|---|---|---|---|---|
| D7 | 1010010101 | 0 | D248 | 0101101001 | 1 | $\overline{a\oplus b}\bullet a\oplus d\bullet c\oplus g\bullet d\oplus f\bullet e\oplus g'\bullet g\oplus h\bullet i'\bullet j$ |
| D2 | 0111000101 | 0 | D253 | 1000111001 | 1 | |
| D9 | 1001001101 | 0 | | | | $\overline{c\oplus d}\bullet a\bullet b'\bullet e'\bullet f'\bullet g\bullet h\bullet i'\bullet j$ |
| D5 | 1010001101 | 0 | | | | |
| D6 | 0110010101 | 0 | | | | $\overline{a\oplus b}\bullet a\oplus e'\bullet e\oplus f\bullet c\bullet d'\bullet g'\bullet h\bullet i'\bullet j$ |
| D61 | 1010100101 | 0 | | | | |

Using these Decoding Labels, the decoding equation for bit ‘H’ can be written as follows:

$$H = h\oplus\{a\oplus d\bullet c\oplus g\bullet d\oplus f \bullet e\oplus g'\bullet g\oplus h\bullet i'\bullet j + \quad \text{(Pn100)}$$
$$a\oplus e'\bullet e\oplus f\bullet c \bullet d'\bullet g'\bullet h \bullet i'\bullet j + a\bullet b'\bullet e'\bullet f' \bullet g\bullet h\bullet i'\bullet j\} \quad \text{(n101+Pn97)}$$

*i) Decoded Bit K*

The K-bit value for all vectors of Table 28 through 35 is zero. The seven encoded control characters which have a K-value of one are listed in Table 36. All seven control characters have alternate representations. The determination of the K-bit values is made directly from the primary or alternate representation rather than exclusively from the restored primary vectors in order to avoid the extra latency associated with serial operation of primary vector restoration and bit value determination.

**Table 36. K-bit Decoding**

| Name | abcdefghi j | Name | abcdefghi j | K | Decoding Label |
|---|---|---|---|---|---|
| C9P | 1001000001 | C9A | 0110111110 | 1 | $a\oplus c\bullet\overline{b\oplus d}\bullet c\oplus g'\bullet d\oplus h\bullet e\oplus f'\bullet$ |
| *K131P* | 1100000100 | *K131A* | 0011111011 | 1 | $f\oplus g'\bullet f\oplus i'\bullet h\oplus j$ |
| K81P | 1000101000 | K81A | 0111010111 | 1 | |
| K82P | 0100101000 | K82A | 1011010111 | 1 | $(\overline{a\oplus b'}\bullet b\oplus c\bullet c\oplus d+\overline{a\oplus b}\bullet c\oplus d')\bullet$ |
| K84P | 0010101000 | K84A | 1101010111 | 1 | $d\oplus e\bullet d\oplus i'\bullet e\oplus f\bullet f\oplus g\bullet g\oplus h\bullet i\oplus j'$ |
| *K124P* | 0011111000 | *K124A* | 1100000111 | 1 | $a\oplus b'\bullet b\oplus g\bullet\overline{c\oplus h}\bullet d\oplus e'\bullet e\oplus f'\bullet$ |
| *K248P* | 0001111100 | *K248A* | 1110000011 | 1 | $f\oplus g'\bullet f\oplus i\bullet i\oplus j'$ |

Using these Decoding Labels, the decoding equation for bit ‘K’ can be written as follows:

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

# III. 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, and XNOR gates (with one exception). 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.

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 and no assurance can be given that the circuit presented is minimum area. Another reason for equivalent different circuit implementation and opportunity to slightly improve the design is the selection of the specific redundant factors in the decoding circuits. In circuit areas which are suspected to be at the upper end 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 show no complex gates to allow maximum circuit sharing; the logic processing programs will introduce complex gates automatically where appropriate.

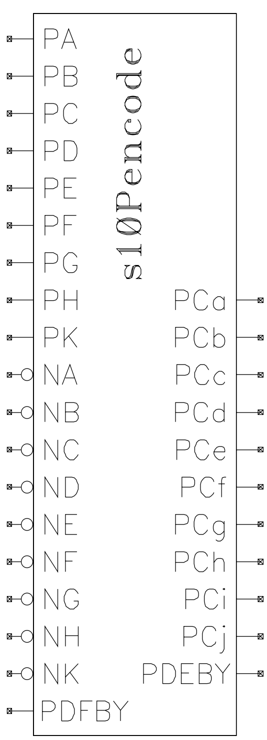


**FIG. 21**

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.

## A) Encoding Circuit

### 1) Bit Encoding

The block diagram for the encoding circuit with all inputs and outputs is shown in FIG. 21. A gate-level circuit diagram of the encoder is shown in FIGS. 22A, 22B and 22C which represent a single circuit with net sharing. Fig. 22A shows most of the encoding of the leading 8 bits (a through h), the encoding of the trailing i and j bits is shown in FIG. 22B, and FIG. 22C shows the implementation of the equations for the complementation of entire vectors on the upper left side and the determination whether an encoded vector is balanced on the lower left side. The upper right side shows the last two gate levels for bit encoding. The generation of the ending disparity DEBY which is equal to the starting disparity DFBY for the next byte is shown at the bottom of the right side. In between are 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.

**2) Complementation Circuit**

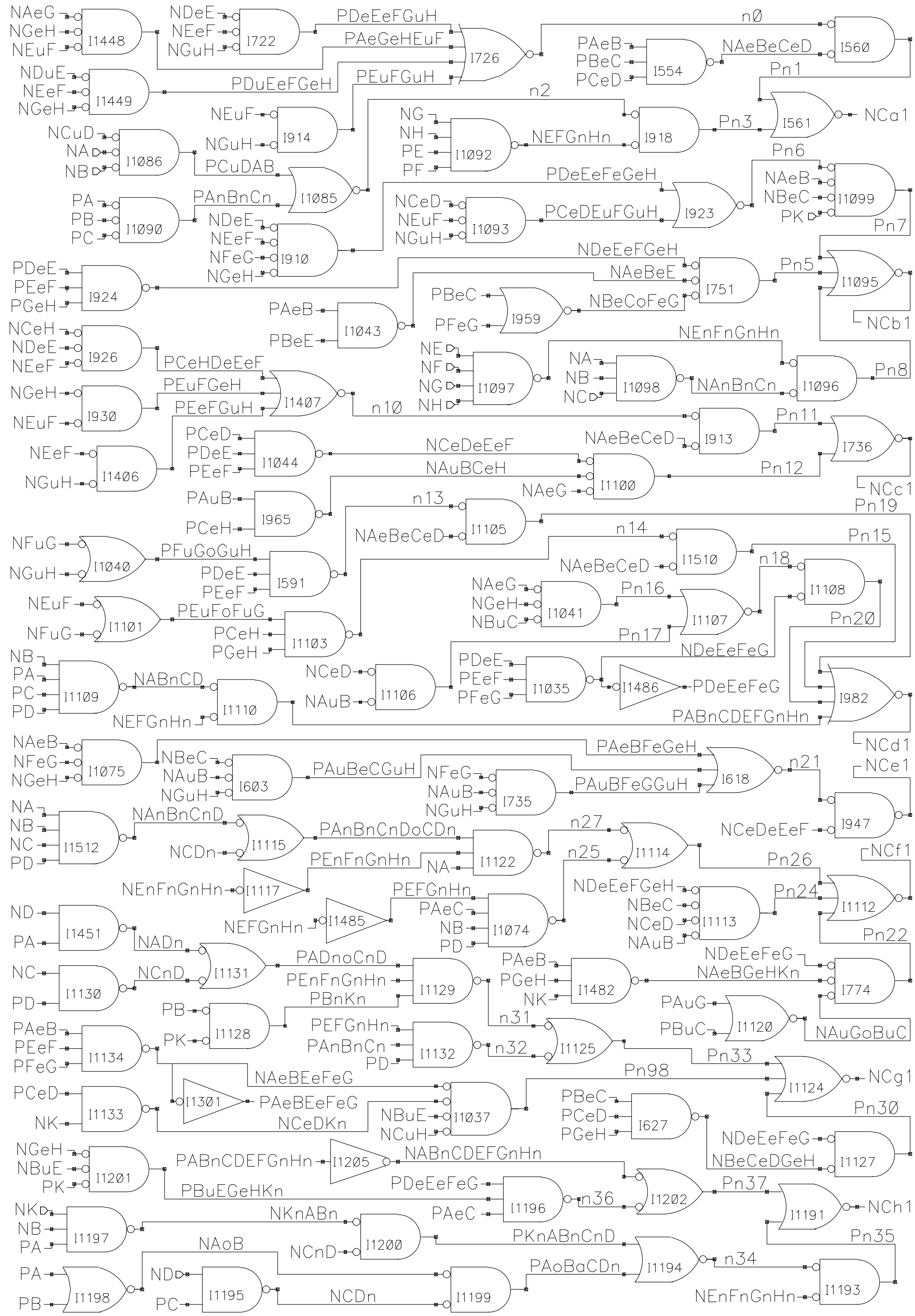
The signal CMPL10 which complements all 10 bits of a coded byte is orthogonal to the other signals (Ca1, Cb1, Cc1, Cd1, Ce1, Cf1, Cg1, Ch1) which cause complementation of individual bits. In other words, both for encoding and decoding, no individual bits are changed when a full vector is complemented 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. 22C, greatly simplifying the circuitry preceding the output EXCLUSIVE OR function. The CMPL10 signal is not explicitly present in the circuit version shown. It is dependent on the required entry disparity and the starting disparity DFBY which is equal to the ending disparity DEBY of the preceding byte. Note that the value of DFBY 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.

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

The encoder comprises 296 gates and a flip-flop (not shown) to keep track of the disparity. No logic path is 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 presented has been structured for easy forward pipe-lining for fast operation at the cost of a few extra gates. If a first encoding step is limited to six logic levels, all the trailing EXCLUSIVE OR functions for the coded bits and for the ending disparity can be moved into the next cycle which requires an extra 21 latches. The first encoding step can be reduced to five gating levels, if the OR functions immediately before the XOR are also moved to a second step which requires only five more latches, a total of 26 extra latches. A reduction to four gating levels in the first step requires moving the two trailing gates for bits e and i, and the three trailing gates for all other signal paths to a second step which requires 60 latches more than the non pipe-lined version (9 for bits A, B, C, D, E, F, G, H, and PDFBY; 19 for the inputs of the gates generating Ca1, Cb1, Cc1, Cd1, Cf1, Cg1, and Ch1; 1 for Ce1; 21 for the inputs of the gates generating PBi, Pn78/79/80, NDFBYaPDRE, and NPDFBYaNDRE; 10 for the inputs of the gates generating n102, NPDB1, and n103).

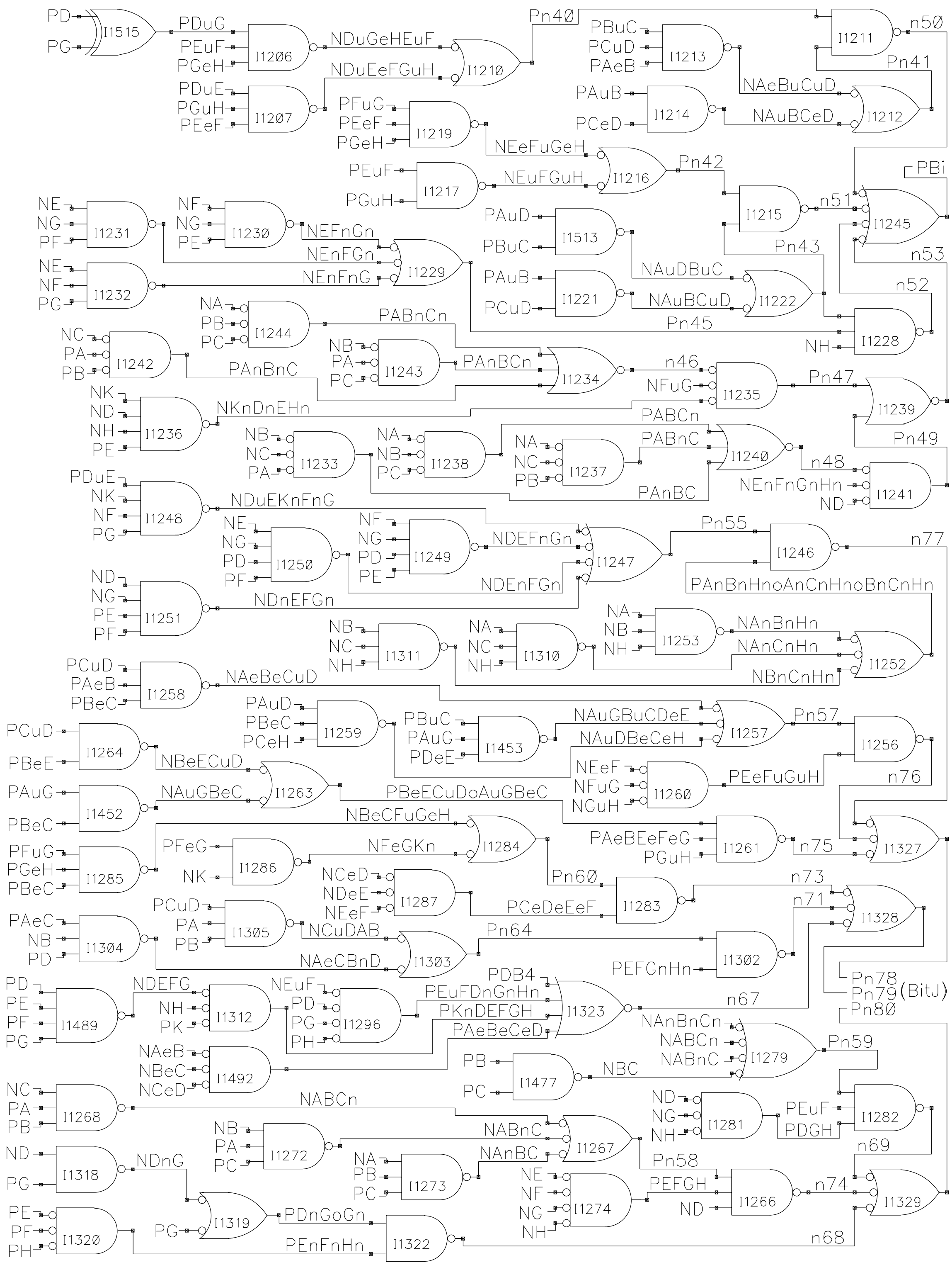
A further delay reduction can be accomplished by itself or in combination with any of the above versions by minor circuit modifications and moving the leading EXCLUSIVE OR functions into the preceding clock cycle in the data source path which requires at most 13 extra latches with complementary outputs.




8B/10B-P BIT ENCODING abcdefgh

11-21-06

**FIG. 22A**




8B/10B-P BIT ENCODING ij 11-27-06

**FIG. 22B**

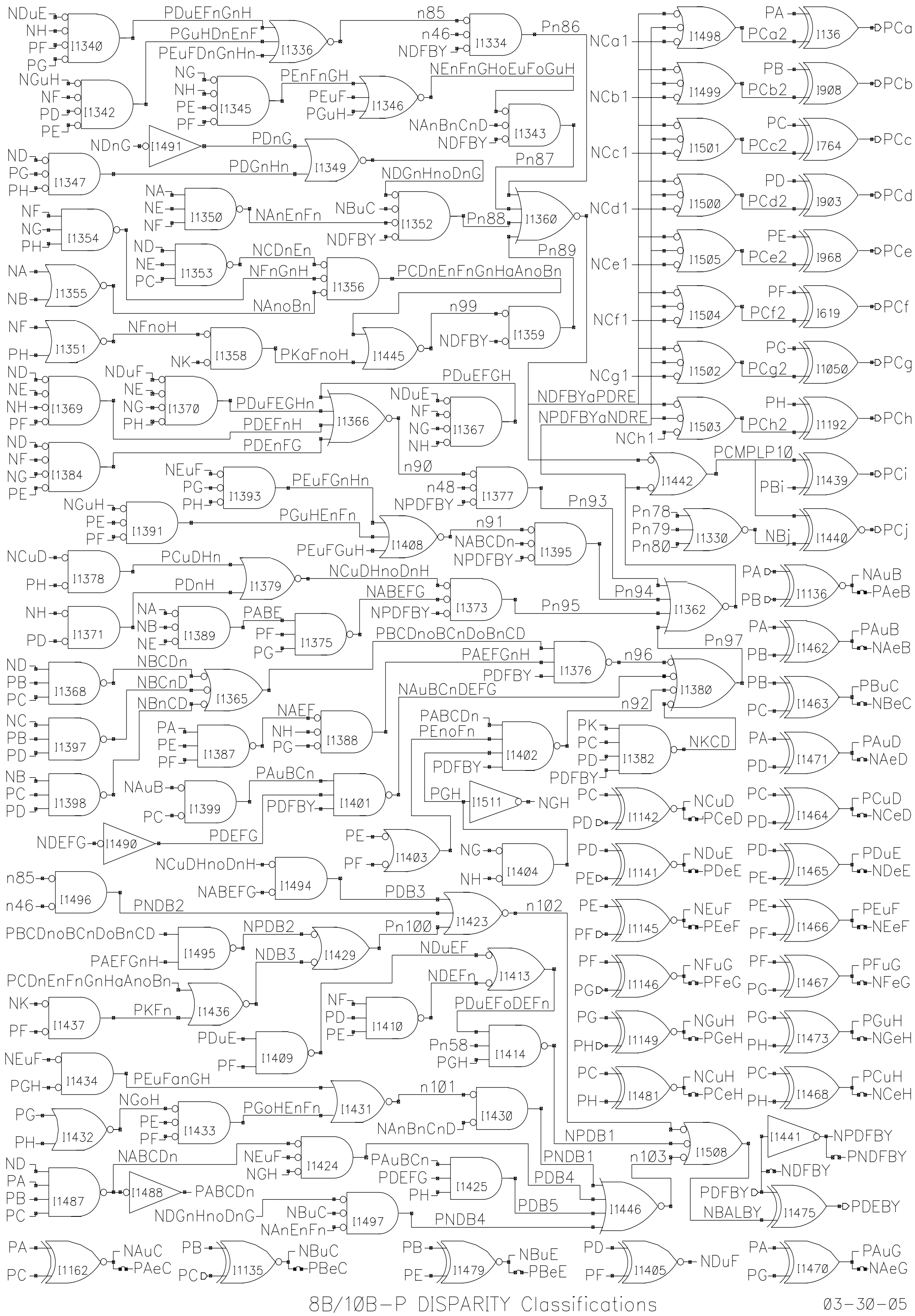

NDuE
NH
PF
PG
I1340
PDuEFnGnH
PGuHDnEnF
PEuFDnGnHn
I1336
n85
n46
NDFBY
I1334
Pn86
NCa1
I1498
PA
PCa2
I136
PCa
NGuH
NF
PD
PE
I1342
NG
NH
PE
PF
I1345
PEnFnGH
PEuF
PGuH
I1346
NEnFnGHoEuFoGuH
NCb1
I1499
PB
PCb2
I908
PCb
NAnBnCnD
NDFBY
I1343
NDnG
I1491
PDnG
ND
PG
PH
I1347
PDGnHn
I1349
NDGnHnoDnG
Pn87
NCc1
I1501
PC
PCc2
I764
PCc
NA
NE
NF
I1350
NAnEnFn
NBuC
NDFBY
I1352
Pn88
I1360
NCd1
I1500
PD
PCd2
I903
PCd
NF
NG
PH
I1354
ND
NE
PC
I1353
NCDnEn
NFnGnH
I1356
Pn89
PCDnEnFnGnHaAnoBn
NCe1
I1505
PE
PCe2
I968
PCe
NA
NB
I1355
NAnoBn
n99
NDFBY
I1359
NCf1
I1504
PF
PCf2
I619
PCf
NF
PH
I1351
NFnoH
NK
I1358
PKaFnoH
I1445
NCg1
I1502
PG
PCg2
I1050
PCg
ND
NE
NH
PF
I1369
NDuF
NE
NG
PH
I1370
PDuFEGHn
PDEFnH
PDEnFG
I1366
PDuEFGH
NDuE
NF
NG
NH
I1367
NDFBYaPDRE
NPDFBYaNDRE
NCh1
I1503
PH
PCh2
I1192
PCh
ND
NF
NG
PE
I1384
n90
n48
NPDFBY
I1377
I1442
PCMPLP10
PBi
I1439
PCi
NEuF
PG
PH
I1393
PEuFGnHn
Pn93
NGuH
PE
PF
I1391
PGuHEnFn
PEuFGuH
I1408
n91
NABCDn
NPDFBY
I1395
Pn78
Pn79
Pn80
I1330
NBj
I1440
PCj
NCuD
PH
I1378
PCuDHn
PDnH
I1379
NCuDHnoDnH
NABEFG
NPDFBY
I1373
Pn95
Pn94
I1362
PA
PB
I1136
NAuB
PAeB
NH
PD
I1371
NA
NB
NE
I1389
PABE
PF
PG
I1375
PBCDnoBCnDoBnCD
PAEFGnH
PDFBY
I1376
n96
Pn97
PA
PB
I1462
PAuB
NAeB
ND
PB
PC
I1368
NBCDn
NBCnD
NBnCD
I1365
NAuBCnDEFG
I1380
n92
PB
PC
I1463
PBuC
NBeC
NC
PB
PD
I1397
PA
PE
PF
I1387
NAEF
NH
PG
I1388
PABCDn
PEnoFn
PDFBY
I1402
PK
PC
PD
PDFBY
I1382
NKCD
PA
PD
I1471
PAuD
NAeD
NB
PC
PD
I1398
NAuB
PC
I1399
PAuBCn
PDFBY
I1401
PGH
I1511
NGH
PC
PD
I1142
NCuD
PCeD
PC
PD
I1464
PCuD
NCeD
NDEFG
I1490
PDEFG
PE
PF
I1403
NG
NH
I1404
PD
PE
I1141
NDuE
PDeE
PD
PE
I1465
PDuE
NDeE
n85
n46
I1496
NCuDHnoDnH
NABEFG
I1494
PDB3
PNDB2
I1423
n102
PE
PF
I1145
NEuF
PEeF
PE
PF
I1466
PEuF
NEeF
PBCDnoBCnDoBnCD
PAEFGnH
I1495
NPDB2
NDB3
I1429
Pn100
NDuEF
NDEFn
I1413
PF
PG
I1146
NFuG
PFeG
PF
PG
I1467
PFuG
NFeG
PCDnEnFnGnHaAnoBn
NK
PF
I1437
PKFn
I1436
NF
PD
PE
I1410
PDuEFoDEFn
PG
PH
I1149
NGuH
PGeH
PG
PH
I1473
PGuH
NGeH
PDuE
PF
I1409
Pn58
PGH
I1414
NEuF
PGH
I1434
PEuFanGH
I1431
n101
NAnBnCnD
I1430
PC
PH
I1481
NCuH
PCeH
PC
PH
I1468
PCuH
NCeH
NGoH
PE
PF
PG
PH
I1432
I1433
PGoHEnFn
NPDB1
PNDB1
n103
I1508
I1441
NPDFBY
PNDFBY
NDFBY
NEuF
NGH
I1424
PAuBCn
PDEFG
PH
I1425
PDB4
PDB5
I1446
ND
PA
PB
PC
I1487
NABCDn
I1488
PABCDn
NDGnHnoDnG
NBuC
NAnEnFn
I1497
PNDB4
PDFBY
NBALBY
I1475
PDEBY
PA
PC
I1162
NAuC
PAeC
PB
PC
I1135
NBuC
PBeC
PB
PE
I1479
NBuE
PBeE
PD
PF
I1405
NDuF
PA
PG
I1470
PAuG
NAeG
8B/10B-P DISPARITY Classifications
03-30-05

**FIG. 22C**

## B) Decoding Circuit

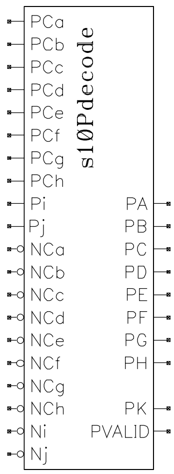


**FIG. 23**

### 1) Bit Decoding

The block diagram for the decoding circuit with all inputs and outputs is shown in FIG. 23. A gate-level circuit diagram of the decoder is shown in FIGS. 24A, 24B, and 24C which represent a single circuit with net sharing. FIG. 24A shows the vector validity checks. The circuit which controls the vector complementation signal (COMPL10) is shown in FIG. 24B. The shared EXCLUSIVE OR functions of all 3 diagrams are shown at the right sides. Again, inverters can be substituted for some of these gates depending on speed requirements. FIG. 24C shows the implementation of the equations for the complementation of individual bits (a, b, c, d, e, f, g, h) to restore the original values (A, B, C, D, E, F, G, H). At the bottom right side, the control bit K is generated. At the top, the trailing two gating levels are shown for the eight data bits.

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

The decoder comprises 275 gates. No logic path exceeds seven gates, all of the inverting type except some XOR gates. The VALID path can be reduced to six logic levels by replacing the inverter I1837 at the center left of FIG. 24A with a NOR3(neg) gate. The PK path is five logic levels.

For fast operation, the circuit presented has been structured for easy forward pipe-lining at the cost of a few extra gates similar to the encoding circuit. For a reduction to six logic levels in the first step, the eight trailing EXCLUSIVE OR functions generating the bits A through H at the top of FIG. 24C are moved into a second step, which requires an extra 16 latches plus two latches to align the PK and PVALID signals. For a reduction to five levels, the OR functions immediately before the XNOR and the trailing gate of the VALID path are also moved to the second step and the K-value is carried forward; this version of pipe-lining requires 23 extra latches. For a reduction of the first step to four gating levels, a total of 48 pipe-lining latches are required (12 for Valid, 7 for CMPL10, 3 for K, 18 for the inputs of the gates generating the signals PCMPLa1 through PCMPLh1, and 8 for the bits PCa thorough PCh).

Again, a further delay reduction to three levels can be accomplished by minor circuit modifications and moving the leading EXCLUSIVE OR functions into the preceding clock cycle which requires at most 23 extra latches with complementary outputs.

## C) Circuit Verification

A soft, technology independent macro has been written in VHDL. The encoder generated all the expected outputs with correct disparity. The decoder restored all the original vector values. A random sequence of all possible 10-bit patterns applied to the decoder input identified all invalid inputs and correctly decoded the valid inputs.

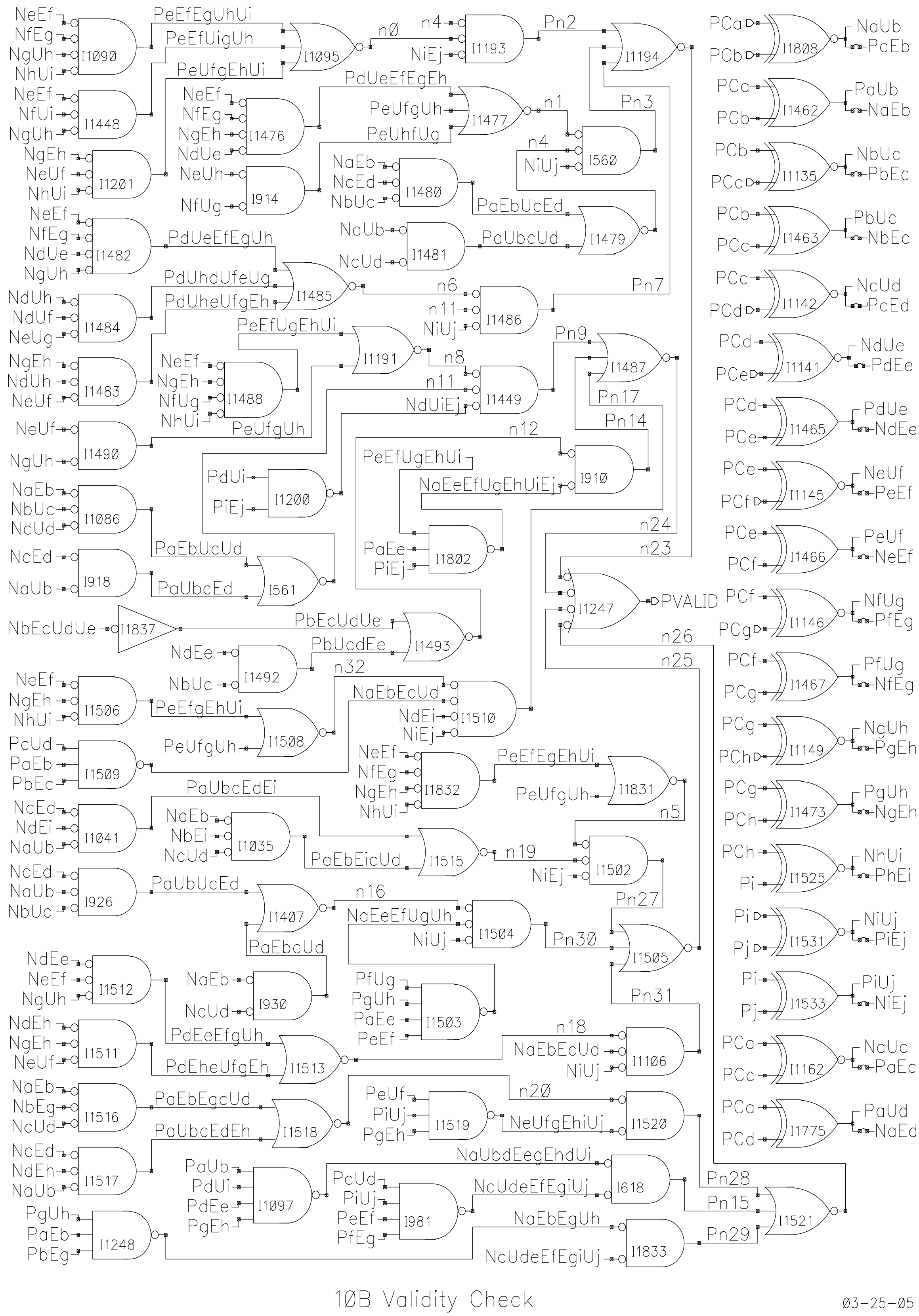



FIG. 24A

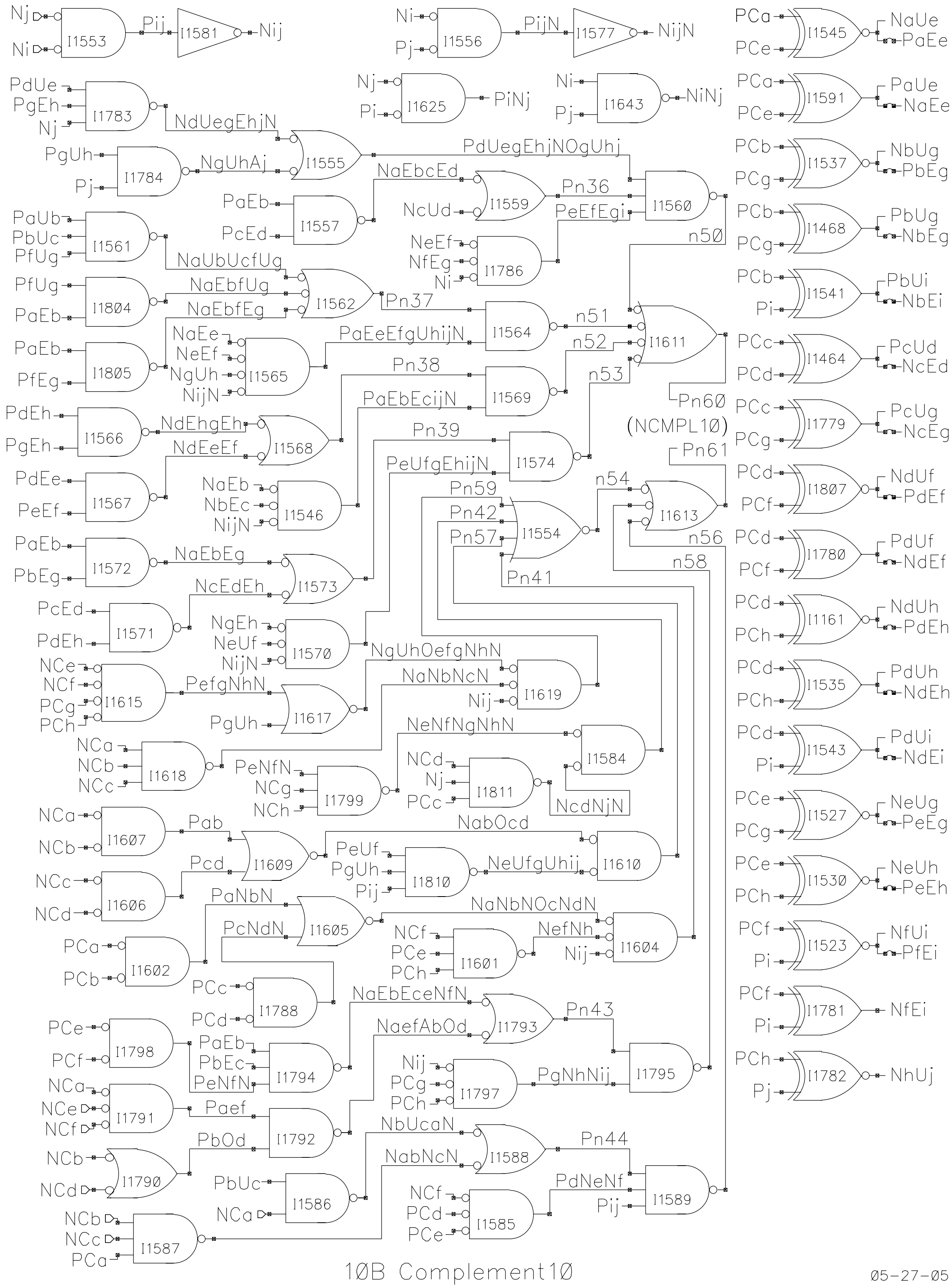

Nj
Ni
I1553
Pij
I1581
Nij
Ni
Pj
I1556
PijN
I1577
NijN
PCa
PCe
I1545
NaUe
PaEe
PdUe
PgEh
Nj
I1783
Nj
Pi
I1625
PiNj
Ni
Pj
I1643
NiNj
PCa
PCe
I1591
PaUe
NaEe
NdUegEhjN
PgUh
Pj
I1784
NgUhAj
I1555
PdUegEhjNOgUhj
PCb
PCg
I1537
NbUg
PbEg
NaEbcEd
PaEb
PcEd
I1557
NcUd
I1559
Pn36
PeEfEgi
I1560
PCb
PCg
I1468
PbUg
NbEg
PaUb
PbUc
PfUg
I1561
NeEf
NfEg
Ni
I1786
n50
NaUbUcfUg
PfUg
PaEb
I1804
NaEbfUg
NaEbfEg
I1562
Pn37
PCb
Pi
I1541
PbUi
NbEi
I1564
n51
PaEb
PfEg
I1805
NaEe
NeEf
NgUh
NijN
I1565
PaEeEfgUhijN
n52
I1611
PCc
PCd
I1464
PcUd
NcEd
Pn38
n53
PdEh
PgEh
I1566
PaEbEcijN
I1569
Pn60
(NCMPL10)
PCc
PCg
I1779
PcUg
NcEg
NdEhgEh
NdEeEf
I1568
Pn39
I1574
Pn61
PdEe
PeEf
I1567
NaEb
NbEc
NijN
I1546
PeUfgEhijN
Pn59
n54
I1613
PCd
PCf
I1807
NdUf
PdEf
Pn42
I1554
Pn57
n56
PCd
PCf
I1780
PdUf
NdEf
PaEb
PbEg
I1572
NaEbEg
I1573
Pn41
n58
NcEdEh
PcEd
PdEh
I1571
NgEh
NeUf
NijN
I1570
PCd
PCh
I1161
NdUh
PdEh
NCe
NCf
PCg
PCh
I1615
NgUhOefgNhN
NaNbNcN
I1619
PefgNhN
PgUh
I1617
Nij
PCd
PCh
I1535
PdUh
NdEh
NeNfNgNhN
NCa
NCb
NCc
I1618
PeNfN
NCg
NCh
I1799
NCd
Nj
PCc
I1811
I1584
NcdNjN
PCd
Pi
I1543
PdUi
NdEi
NCa
NCb
I1607
Pab
I1609
NabOcd
PeUf
PgUh
Pij
I1810
NeUfgUhij
I1610
PCe
PCg
I1527
NeUg
PeEg
NCc
NCd
I1606
Pcd
PaNbN
I1605
NaNbNOcNdN
PCe
PCh
I1530
NeUh
PeEh
PcNdN
NCf
PCe
PCh
I1601
NefNh
Nij
I1604
PCa
PCb
I1602
PCf
Pi
I1523
NfUi
PfEi
PCc
PCd
I1788
NaEbEceNfN
I1793
Pn43
PCf
Pi
I1781
NfEi
PCe
PCf
I1798
PaEb
PbEc
PeNfN
I1794
NaefAbOd
Nij
PCg
PCh
I1797
PgNhNij
I1795
PCh
Pj
I1782
NhUj
NCa
NCe
NCf
I1791
Paef
I1792
PbOd
NCb
NCd
I1790
NbUcaN
NabNcN
I1588
Pn44
PbUc
NCa
I1586
NCf
PCd
PCe
I1585
PdNeNf
Pij
I1589
NCb
NCc
PCa
I1587
10B Complement10
05-27-05

**FIG. 24B**

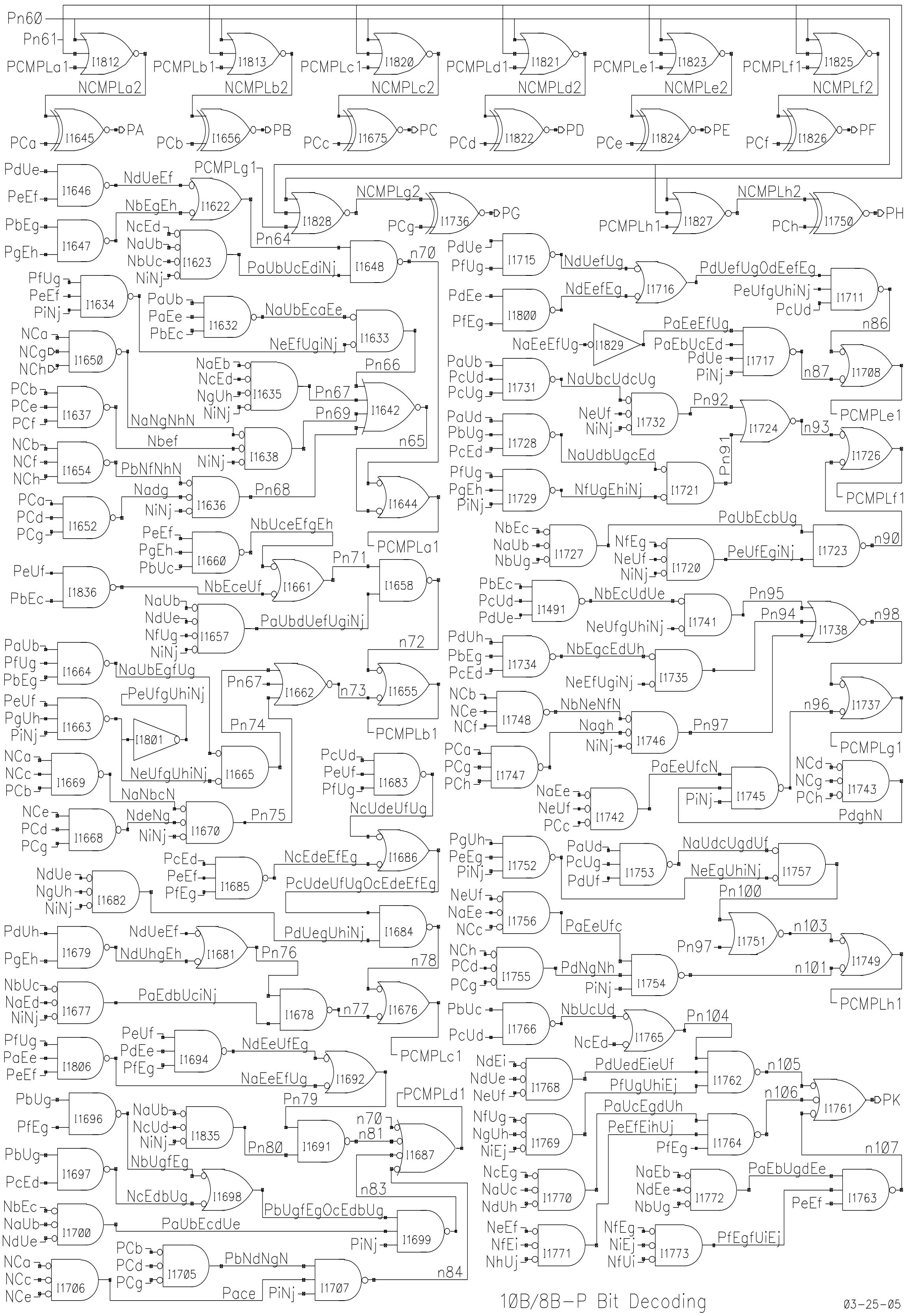



**FIG. 24C**

## References


1. A. X.Widmer, “Error Correction based on Transmission Code Violations and Parity”, IBM Research Report RC 19106 (83280) 8/13/93.
2. IBM US Patent 5,699,062, Albert X. Widmer, “TRANSMISSION CODE HAVING LOCAL PARITY”, Dec. 16, 1997.
3. IBM US Patent 6,876,315 B1, Albert X. Widmer, “DC-BALANCED 6B/8B-P TRANSMISSION CODE WITH LOCAL PARITY”, Apr. 5, 2005.
   Also IBM Research Report RC 23410, Nov. 4, 2004.[1]
4. A. X. Widmer, “The ANSI Fibre Channel Transmission Code”, IBM Research Report RC 18855, 4/23/93.
5. IBM US Patent 6,911,921 B2, Albert X. Widmer, “5B/6B-T, 3B/4B-T, AND PARTITIONED 8B/10B-T AND 10B/12B TRANSMISSION CODE, AND THEIR IMPLEMENTATION FOR HIGH OPERATING RATES”, Jun. 28, 2005.
   Also IBM Research Report RC 23409, Nov. 03, 2004.[1]
6. Optical Internetworking Forum (OIF), Physical Link Layer Working Group (PLL), “Proposal for a very Short Reach (VSR) OC-192/STM64 Interface based on Parallel Optics”, OIF-99,120, Oct. 19-20, 1999 OIF meeting, Los Angeles CA.
7. US Patent 6,978,416 B2, Albert X. Widmer, “ERROR CORRECTION WITH LOW LATENCY FOR BUS STRUCTURES”, Dec. 20, 2005.
8. G. N. N. Martin, “A Rate 8/10 DC Balanced Code with Local Parity”, IBM Technical Disclosure Bulletin, 27(9): 5272-5279, February 1985.


[1] URL for IBM Research Reports:
http://domino.watson.ibm.com/library/cyberdig.nsf/index.html


**Acknowledgment**

Charles Haymes built the VHDL macro and did the computer based verification work.


Reference File locations:

Full Report (Frame Maker): /u/widmer/doc/coding/code8B10B-P-RC
Circuit Diagrams (Cadence cteCds): define ether/homes/axw/widmer/artist/serdesg
→ ether → s10Pencode, s10Pdecode