

NB/MB CODING APPARATUS AND METHOD USING BOTH DISPARITY INDEPENDENT AND DISPARITY DEPENDENT ENCODED VECTORS

Field of the Invention

The present invention generally relates to communications systems and, more particularly, to an NB/MB coding apparatus and method.

Background of the Invention

Coding is employed in communications systems for a variety of purposes. Among these are the improvement of transmission reliability, DC balance, the detection of errors, and the correction of errors. United States Patent No. 5,699,062 to Widmer discloses a transmission code having local parity. The '062 patent describes a method and apparatus for converting 8-bit bytes to a set of coded 10-bit bytes such that if an error accrues in a bit location of a particular coded byte, it generates an invalid coded byte. Further, the set of coded bytes includes a comma which is confined to a single byte. The bit sequence of the comma is singular, i.e., the sequence of bits corresponding to the comma cannot be found in any byte sequence with another alignment relative to the byte boundaries. Vertical parity is used to identify the bit location of errors within a byte known to be erroneous.

The '062 patent does not provide a specific assignment of source vectors to encoded vectors. Accordingly, it would be desirable to provide a coding implementation that can be efficiently implemented in hardware.

Summary of the Invention

Principles of the present invention provide techniques for implementing a coding scheme. An exemplary method, according to the first aspect of the present invention, encodes N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, where M is greater than N which in turn is greater than 0. The exemplary method can include the steps of obtaining a plurality of NB source data vectors and encoding the NB source data vectors into a plurality of MB encoded vectors according to an encoding scheme. The encoding scheme can map at least a first portion of the NB

source data vectors into MB encoded vectors that are disparity independent. Further, the encoding scheme can map a second portion of the NB source data vectors into MB encoded vectors that are disparity dependent. The disparity dependent encoded vectors can have primary representations and alternate representations that are complementary to the primary representations. The MB encoded vectors can have M-N binary symbols appended to them by the encoding scheme. A fraction of the MB encoded vectors can have binary symbol changes other than whole-vector complementation, as compared to corresponding ones of the NB source data vectors. The encoding scheme can be preselected to reduce and/or substantially eliminate the amount of disparity dependent encoded vectors that have binary symbol changes, other than whole-vector complementation. The reduction of the amount of disparity dependent encoded vectors that have individual binary symbol changes can be in comparison to at least some other possible NB to MB encoding schemes.

In an exemplary embodiment of the invention, $N=8$, $M=10$, and an exemplary encoding scheme substantially eliminates the amount of disparity dependent encoded vectors having binary symbol changes other than whole vector complementation. The exemplary encoding scheme can produce a DC-balanced transmission code, and the NB source data vectors that map to disparity independent encoded vectors can include at least 60 balanced source data vectors that have a leading run-length no greater than 2 and are appended with a complementary pair of appended binary symbols corresponding to the aforementioned M-N binary symbols. The MB encoded vectors that have binary symbol changes compared to the source data vectors can be those of the disparity independent encoded vectors that have 8 binary symbols identical to corresponding ones of the disparity independent encoded vectors and two binary symbols that are complements of the pair of appended binary symbols. The exemplary encoding scheme can assign a majority of the NB source data vectors corresponding to the encoded vectors that have binary symbol changes, in complementary pairs, to corresponding pairs of the disparity independent MB encoded vectors that are complementary in all binary symbol positions except the appended binary symbols. Further, the exemplary encoding scheme can assign at least 8 vectors of the

source data vectors that correspond to the disparity independent encoded vectors, having a leading run of 4 followed by two pairs of complementary binary symbols, to certain ones of the MB encoded vectors that have two leading binary symbols complemented with respect to corresponding ones of the 8 vectors. The exemplary encoding scheme can also assign a majority of the source data vectors that correspond to the encoded vectors with binary symbol changes, in complementary pairs, to corresponding pairs of disparity independent encoded vectors that are complementary in all binary symbol positions except the appended binary symbols.

The exemplary encoding scheme can also assign at least 8 vectors of the source data vectors that correspond to the disparity independent encoded vectors, having a leading run of 4, followed by a single binary symbol in 4 subsequent binary symbols, where said single binary symbol matches binary symbols in the leading run of 4, to selected ones of the encoded vectors that have a third binary symbol complemented with respect to corresponding ones of the 8 vectors. The exemplary encoding scheme can still further assign a majority of the NB source data vectors corresponding to the fraction of encoded vectors having binary symbol changes, in complementary pairs, to corresponding pairs of disparity independent encoded vectors that are complementary in all binary symbol positions except the appended binary symbols. The exemplary encoding scheme can yet further assign at least 8 vectors of the source data vectors that correspond to the disparity independent encoded vectors, having a leading run of 4 followed by a single binary symbol in 4 subsequent binary symbols, where said single binary symbol does not match binary symbols in the leading run of 4, to selected ones of the encoded vectors that have first, third and fourth binary symbols complemented with respect to corresponding ones of the 8 vectors.

In further exemplary embodiments thereof, the source data vectors that correspond to the disparity independent encoded vectors can include at least 54 source data vectors that have an overall disparity of either +2 or -2, and have a running disparity of no greater than 2 after any given binary symbol position. These vectors can be appended with binary symbols 00 for those that have an overall disparity of +2, and with binary symbols 11, for those which have an

overall disparity of -2. The source data vectors that correspond to disparity dependent encoded vectors can include at least 19 source data vectors that have a disparity of +2 and that are mapped to 19 disparity dependent balanced encoded vectors that have a required negative starting disparity and are identical to the source data vectors with two zeros appended to them. The 19 source data vectors in this case can satisfy at least one of the following conditions: (a) end in binary symbols 11001, (b) start with binary symbols 1110 and end with binary symbol 1, and (c) end with binary symbols 10 and have at most 3 leading ones or at most one leading 0.

The source data vectors that correspond to the disparity dependent encoded vectors can also include at least four balanced source data vectors that start with the binary symbols 1110 and are mapped to four disparity dependent balanced encoded vectors having a required negative starting disparity. The encoded vectors can have binary symbols identical to the balanced source data vectors with the binary symbols 01 appended to them. The source data vectors that correspond to the disparity dependent encoded vectors can further include at least four balanced source data vectors that start with the binary symbols 0001 and are mapped to four disparity dependent balanced encoded vectors that have a required positive starting disparity. These encoded vectors can have binary symbols identical to the source data vectors with the binary symbol 01 appended to them. Yet further, the source data vectors that correspond to the disparity dependent encoded vectors can include at least 18 source data vectors that have a disparity of +4 and that satisfy one of the following conditions: (a) the trailing four binary symbols include a complementary pair of binary symbols followed by 11, (b) the trailing four binary symbols are 1111 and the leading four binary symbols are either two pairs of complementary binary symbols or 1100, (c) the leading binary symbol is 1 and the trailing four binary symbols are 1101, and (d) the leading two binary symbols are 1 and the trailing four binary symbols are 1110. These 18 source data vectors can be mapped to 18 encoded vectors that have a disparity of +4 and a required negative starting disparity. The encoded vectors can have binary symbols identical to the 18 source data vectors with 01 appended thereto.

Yet further, the source data vectors that correspond to the disparity-dependent encoded vectors can include at least one source data vector in the form 11101111 that is mapped to one encoded vector having a disparity of +4 and a required negative starting disparity and is in the form 1110111100. Still further, the source data vectors that are mapped to disparity-dependent encoded vectors can include at least 21 vectors that have a disparity of -2 and that satisfy one of the following conditions: (a) the trailing four binary symbols include 01 followed by a pair of complementary binary symbols, (b) the trailing binary symbols are either 1001 or 11010, (c) the leading binary symbol is 0 and the trailing four binary symbols are 0011, (d) the trailing five binary symbols are 10001, and (e) the trailing six binary symbols are 100001. The 21 source data vectors can be mapped to 21 encoded vectors that have a disparity of -4 and a required positive starting disparity, and the encoded vectors can have binary symbols that are identical to the source data vectors with the binary symbols 00 appended to them.

Still further, the source data vectors that correspond to the disparity dependent encoded vectors can include at least 15 vectors that have a disparity of -4 and satisfy one of the following conditions: (a) the trailing four binary symbols include a pair of complementary binary symbols followed by the binary symbols 00, (b) the first binary symbol is 0 and the last five binary symbols are 10000, (c) the leading binary symbol is 0 and the trailing four binary symbols are 0010, and (d) the leading two binary symbols are 00 and the trailing four binary symbols are 0001. The 15 data vectors can be mapped to 15 encoded vectors that have a disparity of -4 and a required positive starting disparity, and the binary symbols of the encoded vectors can be the same as those of the source data vectors with the binary symbols 01 appended to them.

In another embodiment of the present invention, the encoding scheme can assign at least 7 NB vectors, as control vectors, to at least 7 MB disparity dependent corresponding encoded control vectors that have primary representations and alternate representations that are complementary to the primary representations. The encoded control vectors can include the 7 NB vectors plus two appended binary symbols. The primary representations of the 7 NB corresponding encoded control vectors can satisfy at least one of the following conditions:

(i) disparity equals minus four, (ii) required starting disparity is positive, (iii) third, fifth, sixth, seventh and ninth binary symbols have a value of zero and first binary symbol has a value of one, (iv) fourth and eighth binary symbols are complementary, and (v) eighth and tenth binary symbols are complementary;

(i) disparity equals minus four, (ii) required starting disparity is positive, (iii) fourth, sixth, eighth, ninth and tenth binary symbols have a value of zero and fifth and seventh binary symbols have a value of one;

(i) disparity equals zero, (ii) required starting disparity is negative, (iii) first two binary symbols and last two binary symbols have a value of zero and third, fourth, fifth, sixth and seventh binary symbols have a value of one; and

(i) disparity equals zero, (ii) required starting disparity is positive, (iii) first two binary symbols and last two binary symbols have a value of zero and fourth, fifth, sixth and seventh and eighth binary symbols have a value of one.

The seven NB control vectors can also have complementary alternate vectors. In certain exemplary embodiments of the present invention, the primary representations of the disparity dependent encoded vectors can end with either of two 2-binary symbol patterns. This can help to simplify the decoding process.

In still another exemplary embodiment of the present invention, the encoding step can include the steps of appending M-N binary symbols to the NB source data vectors to obtain augmented vectors, complementing M binary symbols of a given one of the augmented vectors to obtain one of the alternate representations of one of the MB encoded vectors and complementing less than N binary symbols of another given one of the NB source data vectors. The M binary symbols can be complemented responsive to a determination that the given encoded vector is one of the disparity dependent encoded vectors and the current running disparity does not match the required starting disparity for the particular one of the MB encoded vectors. The complementing of less than N binary symbols can be responsive, at least in part, to a determination that the given encoded vector in such case is one of the disparity independent encoded vectors. The two complementing steps can be performed substantially in parallel.

According to a second aspect of the present invention, there is provided a method of decoding M-binary symbol (MB) encoded vectors into decoded N-binary symbol (NB) source data vectors, $M > N > 0$, comprising the steps of: obtaining a plurality of MB encoded vectors that were encoded from a plurality of NB source data vectors according to an encoding scheme, said encoding scheme mapping at least a first portion of said NB source data vectors into MB encoded vectors comprising disparity independent encoded vectors, said encoding scheme mapping at least a second portion of said NB source data vectors into MB encoded vectors comprising disparity dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said MB encoded vectors having M-N binary symbols appended thereto by said encoding scheme, a fraction of said MB encoded vectors having binary symbol changes, other than whole-vector complementation, compared to corresponding ones of said NB source data vectors, said encoding scheme being preselected to perform at least one of: (i) reducing, compared to at least some other possible NB to MB encoding schemes, and (ii) substantially eliminating an amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors; and decoding said MB encoded vectors into a plurality of NB source data vectors according to decoding rules of said encoding scheme.

According to a third aspect of the present invention, there is provided an apparatus for encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, $M > N > 0$, at least some of said MB encoded vectors comprising disparity-dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said apparatus comprising: a binary symbol appending module configured to append M-N binary symbols to said NB source data vectors to obtain augmented vectors; a full vector complementing module configured to complement M binary symbols of a given one of said augmented vectors to obtain one of said alternate representations of one of said MB encoded vectors corresponding to a given one of said NB source data vectors from which said given one of said augmented vectors was obtained, said complementation of said M binary symbols being at least in part

responsive to a determination that said one of said MB encoded vectors comprises one of said disparity dependent encoded vectors and a current running disparity does not match a required starting disparity for said one of said MB encoded vectors; and a binary symbol complementing module configured to complement less than N binary symbols of another given one of said NB source data vectors to obtain a corresponding portion of another given one of said MB encoded vectors corresponding to said other given one of said NB source data vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said MB encoded vectors comprises a disparity-independent encoded vector, said binary symbol complementing module and said full vector complementing module being configured to operate substantially in parallel, said binary symbol complementing module and said full vector complementing module being coupled to each other and being configured to implement an encoding scheme wherein, in assignment of MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity-independent.

According to a fourth aspect of the present invention, there is provided an apparatus for decoding M-binary symbol (MB) encoded vectors into N-binary symbol (NB) source data vectors, $M > N > 0$, at least some of said MB encoded vectors comprising disparity-dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said apparatus comprising: a full vector complementing module configured to complement at least N binary symbols of a given one of said MB encoded vectors to recover a given one of said NB source data vectors corresponding to said given one of said MB encoded vectors, said complementation of said at least N binary symbols being at least in part responsive to a determination that said given one of said MB encoded vectors comprises one of said alternate versions of one of said disparity-dependent encoded vectors; and a binary symbol complementing module coupled to said full vector complementing module and configured to complement less than N binary symbols of another given one of said MB encoded vectors to recover a corresponding portion of another given one of said NB source data vectors corresponding to said other given one of said MB encoded vectors, said complementation of said less than N binary symbols being at least in part responsive to a

determination that said other given one of said MB encoded vectors comprises a disparity-independent encoded vector, said binary symbol complementing module and said full vector complementing module being configured to operate substantially in parallel, said binary symbol complementing module and said full vector complementing module being configured to implement an encoding scheme wherein, in assignment of MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity-independent.

According to a fifth aspect of the present invention, there is provided a method for encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, $M > N > 0$, at least some of said MB encoded vectors comprising disparity-dependent encoded vectors having primary representations and alternate representations complementary to said primary, said method comprising the steps of: appending M-N binary symbols to said NB source data vectors to obtain augmented vectors; complementing M binary symbols of a given one of said augmented vectors to obtain one of said alternate representations of one of said NB encoded vectors corresponding to a given one of said NB source data vectors from which said given one of said augmented vectors was obtained, said complementation of said M binary symbols being at least in part responsive to a determination that said one of said MB encoded vectors comprises one of said disparity dependent encoded vectors and a current running disparity does not match a required starting disparity for said one of said MB encoded vectors; and complementing less than N binary symbols of another given one of said NB source data vectors to obtain a corresponding portion of another given one of said MB encoded vectors corresponding to said other given one of said NB source data vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said MB encoded vectors comprises a disparity-independent encoded vector, said less than N binary symbol complementing and said N binary symbol complementing steps being performed substantially in parallel and implementing an encoding scheme wherein, in assignment of MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity-independent.

According to a sixth aspect of the present invention, there is provided a method for decoding M-binary symbol (MB) encoded vectors into N-binary symbol (NB) source data vectors, $M > N > 0$, at least some of said MB encoded vectors comprising disparity-dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said method comprising: complementing at least N binary symbols of a given one of said MB encoded vectors to recover a given one of said NB source data vectors corresponding to said given one of said MB encoded vectors, said complementation of said at least N binary symbols being at least in part responsive to a determination that said given one of said MB encoded vectors comprises one of said alternate versions of one of said disparity-dependent encoded vectors; and complementing less than N binary symbols of another given one of said MB encoded vectors to recover a corresponding portion of another given one of said NB source data vectors corresponding to said other given one of said MB encoded vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said NB encoded vectors comprises a disparity-independent encoded vector, said less than N binary symbol complementing and said N binary symbol complementing being performed substantially in parallel and implementing an encoding scheme wherein, in assignment of MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity-independent.

According to seventh aspect of the present invention, there is provided a computer program product comprising a computer useable medium including computer usable program code for encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, $M > N > 0$, said computer program product including: computer usable program code for obtaining a plurality of NB source data vectors; and computer usable program code for encoding said NB source data vectors into a plurality of MB encoded vectors according to an encoding scheme, said encoding scheme mapping at least a first portion of said NB source data vectors into MB encoded vectors comprising disparity independent encoded vectors, said encoding scheme mapping at least a second portion of said NB source data vectors into MB encoded vectors comprising disparity dependent encoded vectors having primary representations and alternate representations

complementary to said primary representations, said MB encoded vectors having M-N binary symbols appended thereto by said encoding scheme, a fraction of said MB encoded vectors having binary symbol changes, other than whole-vector complementation, compared to corresponding ones of said NB source data vectors; wherein said encoding scheme is preselected to perform at least one of: (i) reducing, compared to at least some other possible NB to MB encoding schemes, and (ii) substantially eliminating an amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors.

According to an eighth aspect of the present invention, there is provided a computer program product comprising a computer useable medium including computer usable program code for decoding M-binary symbol (MB) encoded vectors into decoded N-binary symbol (NB) source data vectors, $M > N > 0$, said computer program product including: computer usable program code for obtaining a plurality of MB encoded vectors that were encoded from a plurality of NB source data vectors according to an encoding scheme, said encoding scheme mapping at least a first portion of said NB source data vectors into MB encoded vectors comprising disparity independent encoded vectors, said encoding scheme mapping at least a second portion of said NB source data vectors into MB encoded vectors comprising disparity dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said MB encoded vectors having M-N binary symbols appended thereto by said encoding scheme, a fraction of said MB encoded vectors having binary symbol changes, other than whole-vector complementation, compared to corresponding ones of said NB source data vectors, said encoding scheme being preselected to perform at least one of: (i) reducing, compared to at least some other possible NB to MB encoding schemes, and (ii) substantially eliminating an amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors; and computer usable program code for decoding said MB encoded vectors into a plurality of NB source data vectors according to decoding rules of said encoding scheme.

An exemplary method of decoding encoded vectors into decoded source data vectors, according to another embodiment of the present invention includes the step of obtaining a plurality of MB encoded vectors encoded according to a scheme of the kind described, and then decoding the encoded vectors into a plurality of source data vectors according to decoding rules of the encoding scheme.

An exemplary embodiment of an apparatus for encoding source data vectors into encoded vectors, according, for example, to a coding scheme of the kind described, and in accordance with yet another embodiment of the invention, can include a binary symbol appending module, a full vector complementing module, and a binary symbol complementing module.

An exemplary embodiment of an apparatus for decoding MB encoded vectors into NB source data vectors, in accordance with a coding scheme of the kind described, and according to still a further embodiment of the present invention, can include a full vector complementing module and a binary symbol complementing module.

These and other objects, features and advantages of the present invention will become apparent from the following detailed description of illustrative embodiments thereof, which is to be read in connection with the accompanying drawings.

Brief Description of the Drawings

FIG. 1 is a flow chart depicting method steps of an exemplary method of encoding in accordance with one embodiment of the invention;

FIG. 2 is a flow chart showing one possible specific manner in which encoding can be performed;

FIG. 3 is a flow chart showing method steps of an exemplary method of decoding in accordance with the present invention;

FIG. 4 is a flow chart showing one possible manner of decoding in accordance with the present invention;

FIG. 5 shows an exemplary embodiment of an apparatus for encoding in accordance with an embodiment of the present invention;

FIG. 6 shows an exemplary embodiment of an apparatus for decoding in accordance with an aspect of the present invention;

FIG. 7 depicts trellis diagrams applicable to certain exemplary embodiments of the present invention;

FIGS. 8-19 depict individual trellis diagrams illustrative of certain aspects of an embodiment of the present invention;

FIGS. 20-23 present tables of various encoded vectors according to aspects of an embodiment of the present invention;

FIG. 24 shows an individual trellis diagram pertinent to certain aspects of an embodiment of the present invention;

FIGS. 25-27 are tables showing encoded vectors according to aspects of an exemplary embodiment of the present invention;

FIGS. 28-29 are trellis diagrams of certain vectors according to an illustrative embodiment of the invention;

FIG. 30 is a table showing encoded vectors of an illustrative embodiment of the invention;

FIGS. 31-32 are trellis diagrams for vectors according to an embodiment of the present invention;

FIG. 33 is a table depicting encoded vectors according to an embodiment of the present invention;

FIGS. 34A-34G are a summary table showing source data vectors and encoded vectors, as well as control vectors, in accordance with one specific embodiment of the invention;

FIGS. 35-48 are tables depicting generation of encoded 10B vectors according to an embodiment of the present invention;

FIGS. 51A-60 present tables depicting aspects of a decoding process in accordance with an embodiment of the present invention;

FIG. 61 is a block diagram of one exemplary form of encoding circuit according to an aspect of the present invention;

FIGS. 62A, B and C are logic gate diagrams of the encoding circuit of FIG. 61 according to one exemplary embodiment of the present invention;

FIG. 63 is a block diagram of a decoding circuit according to an exemplary embodiment of the present invention;

FIGS. 64A, B and C are logic gate diagrams of the decoding circuit of FIG. 63 according to an embodiment of the present invention; and

FIG. 65 is a system diagram of an exemplary computer system on which one or more embodiments of the present invention can be implemented.

Detailed Description of Preferred Embodiments

Attention should now be given to FIG. 1, which shows a flow chart 100 depicting method steps of an exemplary method, in accordance with one aspect of the present invention, of encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, with $M > N > 0$, including the steps of obtaining a plurality of NB source data vectors, as per block 102, and encoding the NB source data vectors into a plurality of MB encoded vectors according to an encoding scheme, as per block 104. The encoding scheme will typically include both encoding and decoding rules. In the encoding scheme, at least a first portion of the NB source data vectors are mapped into MB encoded vectors that are disparity independent. Further, at least a second portion of the NB source data vectors are mapped into MB encoded vectors that are disparity dependent and have primary representations and alternate representations that are complementary to the primary representations. The MB encoded vectors typically have M-N binary symbols appended to them by the encoding scheme. A fraction of the MB encoded vectors have binary symbol changes, other than whole-vector complementation, as compared to corresponding ones of the NB source data vectors (by way of a general example, whole vector complementation can be performed to obtain alternate representations of disparity dependent encoded vectors while binary symbol changes other than whole vector complementation can be made in connection with disparity-independent encoded vectors).

The encoding scheme can be preselected to reduce or substantially eliminate the amount of disparity dependent encoded vectors that fall within the fraction of the MB encoded vectors that have binary symbol changes other than whole vector complementation compared to corresponding ones of the NB source data vectors. As used herein, "reducing" the amount of disparity dependent encoded vectors that fall within the fraction described contemplates a reduction compared to at least some other possible NB to MB encoding schemes. Furthermore, as used herein "substantially eliminating" the amount of disparity dependent encoded vectors that fall within the described fraction includes both a complete elimination and an elimination of a sufficient number of such disparity dependent encoded vectors falling within the described fraction

such that encoding and decoding according to the encoding scheme can be implemented in hardware in a manner permitting advantageous gains due to parallel processing in at least some aspects of the encoding and decoding process.

In one exemplary encoding scheme according to the present invention, to be described in detail below, $N=8$, $M=10$, and the encoding scheme substantially eliminates the amount of disparity dependent encoded vectors that fall within the fraction of the MB encoded vectors that have binary symbol changes, other than whole-vector complementation, as compared to corresponding ones of the NB source data vectors.

Attention should now be given to FIG. 2, which presents a flow chart 200 showing exemplary method steps for encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, with $M > N > 0$, according to aspects of another exemplary embodiment of the present invention. At least some of the MB encoded vectors can include disparity dependent encoded vectors that have primary representations and alternate representations complementary to the primary representations. The method can include the steps of appending M-N binary symbols to the NB source data vectors to obtain augmented vectors, as per block 202, complementing M binary symbols of a given one of the augmented vectors as per block 204, and complementing less than N binary symbols of another given source data vector to obtain a corresponding portion of another given MB encoded vector, as per block 206. Blocks 204 and 206 are preferably performed substantially in parallel. As used herein, "substantially in parallel" means either entirely in parallel or with sufficient parallelism that desirable enhancements in processing associated with encoding and/or decoding can be achieved.

In the step of complementing M binary symbols of the given one of the augmented vectors, such complementing is performed to obtain one of the alternate representations of one of the MB encoded vectors that corresponds to a given one of the NB source data vectors from which the given one of the augmented vectors was obtained. The complementation of the M binary symbols is, at least in part, responsive to a determination that the one of the MB encoded vectors is one of the disparity dependent

encoded vectors, and that a current running disparity does not match a required starting disparity for the one of the MB encoded vectors.

In the step of complementing less than N binary symbols of the other given one of the NB source data vectors, such complementing is carried out to obtain a corresponding portion of another given one of the MB encoded vectors that corresponds to the other given one of the NB source data vectors. The complementation of less than N binary symbols is, at least in part, responsive to a determination that the other given one of the MB encoded vectors is a disparity independent encoded vector. The method of FIG. 2 can implement an encoding scheme where, in the assignment of MB encoded vectors that have binary symbol changes compared to corresponding ones of the NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity independent. As described, the complementing of less than N-binary symbols can be performed directly on the source data vectors; however, it is believed advantageous that such complementing be carried out on individual bits of the augmented vectors formed in the step illustrated in block 202. In one specific implementation of the method shown in FIG. 2, $N=8$, $M=10$, and the encoding scheme assigns substantially all of the MB encoded vectors that have binary symbol changes compared to corresponding ones of NB source data vectors to MB encoded vectors that are balanced and disparity independent. As used herein, "substantially all" includes both all and a sufficient number such that advantageous benefits associated with parallel processing, as discussed above, can be achieved.

The method steps depicted in FIG. 2 can also be adapted as sub-steps for performing the encoding step 104 depicted in FIG. 1 (using, for example, an encoding scheme as described in connection therewith). In such case, the appending step is substantially as previously described. In the step of complementing M binary symbols, the NB source data vectors from which the augmented vectors were obtained fall in the second portion of NB source data vectors as described above.

Attention should now be given to FIG. 3, which shows a flow chart 300 depicting exemplary method steps of decoding M-binary symbol (MB) encoded vectors into decoded N-binary symbol (NB) source data vectors, with $M > N > 0$, according to

another exemplary method of the present invention. The method includes the steps of obtaining a plurality of MB encoded vectors encoded according to an encoding scheme, as per block 302, and decoding the MB encoded vectors into a plurality of NB source data vectors according to decoding rules of the encoding scheme, as per block 304. The encoding scheme can be of any type in accordance with the present invention.

Now viewing FIG. 4, a flow chart 400 is depicted illustrating method steps, of another exemplary method, according to another aspect of the present invention, for decoding M-binary symbol (MB) encoded vectors into N-binary symbol (NB) source data vectors, $M > N > 0$, where at least some of the MB encoded vectors are disparity dependent encoded vectors that have primary representations and alternate representations complementary to the primary representations. The method can include the step, as per block 402, of complementing at least N binary symbols of a given one of the MB encoded vectors to recover a given one of the NB source data vectors that corresponds to the given one of the MB encoded vectors. The complementation of the at least N binary symbols can be responsive, at least in part, to a determination that the given one of the MB encoded vectors is one of the alternate versions of one of the disparity dependent encoded vectors. The method can also include the step of complementing less than N binary symbols of another given one of the MB encoded vectors, as per block 404. Such complementing of less than N binary symbols can be performed to recover a corresponding portion of another given one of the NB source data vectors that corresponds to the other given one of the MB encoded vectors. The complementation of less than N binary symbols can be, at least in part, responsive to a determination that the other given one of the MB encoded vectors is a disparity independent encoded vector. The two complementing steps 402, 404 can be performed substantially in parallel, and can implement an encoding scheme in accordance with the present invention. In one more particular aspect of the method of FIG. 4, $N=8$, $M=10$, and the encoding scheme assigns substantially all of the MB encoded vectors that have binary symbol changes compared to corresponding ones of the NB source data vectors to MB encoded vectors that are balanced and disparity independent.

The method steps illustrated in FIG. 4 can also represent one manner in which step 304 of FIG. 3 can be performed. In such case, in the step 402, the NB source data vectors are contained within the aforementioned second portion of NB source data vectors. Further, in complementing step 404, the NB source data vectors are contained within the aforementioned first portion of NB source data vectors.

In illustrative embodiments of the invention, the encoding scheme produces a DC-balanced transmission code. Further, in illustrative embodiments of the invention, the primary representations of the disparity dependent encoded vectors end with one of two 2-binary symbol patterns. Further, in illustrative embodiments of the invention, NB control vectors and corresponding MB encoded control vectors can each have a complementary alternate vector. In an exemplary encoding scheme discussed below, there are 7 NB control vectors and 7 MB corresponding encoded control vectors, each with complementary alternate vectors.

Reference should now be had to FIG. 5, which depicts an exemplary apparatus 500 for encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, according to an aspect of the present invention. $M > N > 0$, and at least some of the MB encoded vectors are disparity dependent encoded vectors having primary representations and alternate representations complementary to the primary representations. The apparatus 500 can include a binary symbol appending module 502; optionally, a disparity monitoring module 504; a full vector complementing module 506; and a binary symbol complementing module 508. Binary symbol appending module 502 can be configured to append M-N binary symbols to the NB source data vectors so as to obtain augmented vectors. Where employed, disparity monitoring module 504 can be coupled to the full vector complementing module 506, and can be configured to determine current running disparity for use in assigning proper ones of the disparity dependent encoded vectors to given ones of the NB source data vectors.

Full vector complementing module 506 can be configured to complement M binary symbols of a given one of the augmented vectors to obtain one of the alternate representations of one of the MB encoded vectors that corresponds to a given one of the NB source data vectors from which the given one of the augmented vectors was obtained.

The complementation of M binary symbols can be, at least in part, responsive to a determination that the one of the MB encoded vectors comprises one of the disparity dependent encoded vectors and that a current running disparity does not match the required starting disparity for the one of the MB encoded vectors. The binary symbol complementing module 508 can be configured to complement less than N binary symbols of another given one of the NB source data vectors to obtain a corresponding portion of another given one of the MB encoded vectors that corresponds to the other given one of the NB source data vectors. The complementation of less than N binary symbols can be performed, at least in part, responsive to a determination that the other given one of the MB encoded vectors is a disparity independent encoded vector. The binary symbol complementing module 508 and the full vector complementing module 506 can be configured to operate substantially in parallel. As used herein, "substantially in parallel" should have the same meaning as set forth above. The modules 506, 508 can be coupled to each other and can be configured to implement any of the encoding schemes described herein. It is believed preferable that the module 508 complements appropriate individual bits of the augmented vector, but any appropriate scheme for complementing one or more individual binary symbols is encompassed within the scope of the present invention. Further, note that as used herein, "coupled" should be understood broadly to include direct coupling, indirect coupling through one or more other components, sharing of one or more logic gates as discussed below, and the like. Note in addition that, in exemplary embodiments, modules 602 and 604 can "see" all vectors at their inputs but act only on those identified by their labels on the input lines, if required. In the exemplary encoding scheme discussed below (which is meant to be exemplary and not limiting), apart from the appended binary symbols, for random data, about half the source data vectors and all the control vectors remain unchanged upon encoding.

Attention should now be given to FIG. 6, which illustrates an exemplary embodiment of apparatus 600 for decoding M-binary symbol (MB) encoded vectors into N-binary symbol (NB) source data vectors, in accordance with an aspect of the present invention ($M > N > 0$). At least some of the MB encoded vectors are disparity dependent encoded vectors that have primary representation and alternate representations

complementary to the primary representations. Apparatus 600 includes a full vector complementing module 602, a binary symbol complementing module 604, and, optionally, a validity check module 606. Full vector complementing module 602 can be configured to complement at least N binary symbols of a given one of the MB encoded vectors to recover a given one of the NB source data vectors that corresponds to the given one of the MB encoded vectors. The complementation of the at least N binary symbols can be, at least in part, responsive to a determination that the given one of the MB encoded vectors is one of the alternate versions of one of the disparity dependent encoded vectors.

The binary symbol complementing binary module 604 can be coupled to the full vector complementing module 602 and can be configured to complement less than N binary symbols of another given one of the MB encoded vectors to recover a corresponding portion of another given one of the NB source data vectors corresponding to the other given one of the MB encoded vectors. The complementation of less than N binary symbols can be performed, at least in part, responsive to a determination that the other given one of the MB encoded vectors is a disparity independent encoded vector. Modules 602, 604, and (optionally) 606 can be configured to operate substantially in parallel, where "substantially in parallel" has the meaning set forth above. Modules 602, 604, and 606 can be configured to implement any encoding scheme in accordance with the present invention. For invalid vectors, modules 602 and 604 can be allowed to generate arbitrary outputs. In the exemplary embodiment depicted in FIG. 6, modules 602, 604 are also configured to strip off the appended binary symbols. Modules 602, 604, 606 can "see" the appended binary symbols at the inputs but such symbols can be dropped before complementation. Note that the full vector complementing module does not have to complement vectors that are already in their primary (as opposed to alternate) form.

Where employed, validity check module 606 can be coupled to modules 602, 604 and can be configured to obtain putative encoded vectors and to determine if given ones of the putative encoded vectors are valid MB encoded vectors. Note that this can be performed by comparing received vectors to valid vectors to determine whether they are

valid, or, conversely, by determining whether they are invalid, for example, by comparing them to invalid vectors.

The methods described herein can be implemented in a number of different ways: for example, the apparatuses depicted in FIGS. 5 and 6 can be employed. Such apparatuses can in turn be implemented using a number of techniques. At present, it is believed that a logic gate implementation of the kind described below is preferable. Various changes and modifications can be made to the exemplary logic gate implementation, as noted herein.

The following is an exemplary 8B/10B encoding scheme that illustrates principles and techniques of the present invention. Note that the capital "B" in 8B/10B refers generally to "Binary Symbol," and is not limited to the more specific term "bit" (typically represented by "b"), 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 (NB/MB schemes according to the present invention may, if desired, also have one or more extra inputs, such as control characters). The bits of the uncoded 8B data vectors are labeled with the upper case letters "ABCDEFGH" and the control input for special non-data characters is labeled with "K." The bits of the coded 10B vectors are labeled with the lower case letters "abcdefghij."

In the trellis diagrams such as shown in FIG. 7, 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. 7 are labeled 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 (balance) indicates a disparity of 0
- u (up, uni) 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 (minus) 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 (cube) 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 (three) 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 (quint) 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 (hepta) 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 (seven) 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 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 (nine, negative) 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. 7 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. 7 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.

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 **balanced** coded vectors.
 - P indicates a complementary pair of disparity dependent balanced coded vectors which are selected based on the **Polarity** of the running disparity.
 - F indicates a complementary pair of coded vectors with a disparity of **four**.

- 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. 7 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. 7.
- 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.

At all 10B boundaries, the running disparity can assume one of two values $D=\pm 2$. 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."

The 8B/10B-P code includes a total of 263 source vectors each translated into one of 352 coded 10B vectors as illustrated by the trellis diagrams of FIGS. 8-19 and 24. All the other 672 10B vectors are invalid. The use and interpretation of trellis diagrams for this kind of application is explained in the '062 patent and is known to the skilled artisan. 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, and the remaining 60 source vectors are each encoded into one of a complementary pair of vectors with a disparity of four.

Three extra control characters have been added, as compared to the code described in the '062 patent, which can generate contiguous runs of five. It is the user's 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.

There are 174 disparity independent balanced vectors as illustrated in FIGS. 8 and 9. 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.

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. 10 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 FIGS. 8 and 9 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. 8 redundantly. They can be excluded by the addition of the term $5u'5m'$.

FIGS. 10 to 12 show balanced trellises with a required negative starting disparity. For a positive running disparity, their complements must be used. The three vectors of FIG. 10 can be described by the expression $P3u5c7u8u$ and their complements by $P3m5t7m8m$. The eleven vectors of FIG. 11 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. The fourteen balanced vectors of FIG. 12 can be described by the expression $P4c'6u7c$. The vector through the node $2m$ is assigned to an optional primary control character K124P.

The thirty-two vectors of FIG. 13 with a disparity of plus four can be described by the expression F4u6u. Its complement is described by the expression F4m6m. The nine vectors of FIG. 14 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 bold lines represent the singular comma sequence. The nine vectors of FIG. 15 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. The four vectors of FIG. 16 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. The single vector of FIG. 17 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. The four vectors of FIG. 18 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. The single vector of FIG. 19 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.

The trellis diagrams of FIGS. 8 to 19 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 combination of any trailing and leading runs of true and complement forms with their associated running disparity at the byte boundaries shows the conformance with the run length rules. Similarly, the singularity of the comma character can be assured by the examination of all possible combinations of trailing and leading bit patterns.

The specific assignment of source vectors to coded vectors 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 the tables of FIGS. 20-27, where the tables of FIGS. 23-27 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. Other endings could have been chosen; it is the limitation to just two ending patterns that helps to simplify the decoding, and not necessarily the two specific patterns chosen. All source vectors which require individual bit changes for encoding are listed in the table of FIG. 28. The 60 coded vectors of the table of FIG. 28 can be identified as disparity independent, balanced vectors ending with $ij=01$.

The value of the K-bit is not listed with the source vectors of the table of FIG. 28. 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. The 114 disparity independent balanced vectors listed in the tables of FIGS. 20, 21, and 22 are the subset of the vectors of FIGS. 8 and 9 which end with $ij = 00, 11$, or 10 , but not those which end with 01 . The 27 vectors of the table of FIG. 20 can be identified by BU3c'5c'7c'3t'. The 27 vectors of the table of FIG. 21 can be identified by BM3c'3t'5t'3t'7t'IJ. It will be appreciated that the vectors in the tables of FIGS. 20 and 21 result from 54 source data vectors having an overall disparity of one of plus 2 and minus 2, and having a running disparity of at most 2 after any given binary symbol position, that are appended with binary symbols 00, for those of the 54 source data vectors having an overall disparity of plus 2, and binary symbols 11, for those of the 54 source data vectors having an overall disparity of minus 2. The 60 vectors of the table of FIG. 22 can be identified by BB3c'3t'I. It will be appreciated that the 60 vectors are the result of taking 60 balanced source data vectors having a leading run-length no greater than 2 and appending a complementary pair of binary symbols.

The 24 balanced vectors listed in the table of FIG. 23 preserve the values of the source bits and are the subset of the vectors of FIGS. 10-12 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. 12. It will be appreciated that the table of FIG. 23 shows 19 vectors

that result from starting with source data vectors having a disparity of plus 2 and satisfying at least one of:

ending in binary symbols 11001;

starting with binary symbols 1110 and ending with binary symbol 1; and

ending with binary symbols 10 and having one of: (i) at most three leading ones, and (ii) at most one leading zero; and

mapping them to 19 disparity dependent balanced encoded vectors having a required negative starting disparity (the encoded vectors have binary symbols identical to the source data vectors with two zeroes appended thereto). Further, it will be appreciated that the table of FIG. 23 shows 4 vectors resulting from balanced source data vectors starting with binary symbols 1110, the 4 balanced source data vectors being mapped to 4 disparity dependent balanced encoded vectors having a required negative starting disparity, the encoded vectors having binary symbols identical to the 4 balanced source data vectors with binary symbols 01 appended thereto.

The five balanced vectors of FIG. 24 and the table of FIG. 25 require a positive entry disparity. They are the complements of those five vectors of FIG. 11 which end with $ij=10$ or 11 . 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 $ij=01$. It will be appreciated that the table of FIG. 25 shows 4 vectors resulting from 4 balanced source data vectors starting with binary symbols 0001, the 4 balanced source data vectors being mapped to 4 disparity dependent balanced encoded vectors having a required positive starting disparity, the encoded vectors having binary symbols identical to the source data vectors with binary symbols 01 appended thereto. The nineteen vectors of the table of FIG. 26 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. 17 and the others are a subset of the vectors of FIGS. 13-16 and 18, and end with $ij = 01$. It will be appreciated that the table of FIG. 26 includes 18 vectors that result from source data vectors having a disparity of plus 4 and satisfying one of:

trailing 4 binary symbols comprise a complementary pair of binary symbols followed by binary symbols 11;

trailing 4 binary symbols are 1111 and leading 4 binary symbols are one of:

(i) two pairs of complementary binary symbols, and (ii) 1100;

leading binary symbol is 1 and trailing 4 binary symbols are 1101; and

leading two binary symbols are 1 and trailing 4 binary symbols are 1110;

the 18 source data vectors being mapped to 18 encoded vectors having a disparity of plus 4 and a required negative starting disparity, the encoded vectors having binary symbols identical to said 18 source data vectors with 01 appended thereto. Further, the table of FIG. 26 includes at least one vector resulting from a source data vector in the form 11101111, the one source data vector being mapped to one encoded vector having a disparity of plus 4 and a required negative starting disparity, the encoded vector having the form 1110111100.

The twenty-five vectors of the table of FIG. 27 have a disparity of minus four and the value of the source bits is preserved. They are the complements of a subset of vectors of FIGS. 13-15 and 18 and end with $ij = 00$. FIG. 28 is the trellis of the top 16 vectors in the table of FIG. 27 which are a complemented subset of the vectors of FIG. 13. FIG. 29 is the trellis of the bottom data vector sets which are complemented subsets of the vectors of FIGS. 14, 15, and 18, respectively. They all have a disparity of minus four and the value of the source bits is preserved. K131 is not shown. It will be appreciated that the table of FIG. 27 includes 21 vectors resulting from source vectors having a disparity of minus 2 and satisfying one of:

trailing 4 binary symbols comprise 01 followed by a pair of complementary binary symbols;

trailing binary symbols are one of 1001 and 11010;

leading binary symbol is 0 and trailing 4 binary symbols are 0011;

trailing 5 binary symbols are 10001; and

trailing 6 binary symbols are 100001;

the 21 source data vectors being mapped to 21 encoded vectors having a disparity of minus 4 and a required positive starting disparity, the 21 encoded vectors having binary

symbols identical to said 21 source data vectors with binary symbols 00 appended thereto.

The sixteen vectors of the table of FIG. 30 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. 13-16 and end with $ij=01$ and are illustrated with their trellises in FIG. 31. It will be appreciated that the table of FIG. 30 includes 15 vectors that result from source vectors having a disparity of minus 4 and satisfying one of:

trailing 4 binary symbols comprise a pair of complementary binary symbols followed by binary symbols 00;

first binary symbol is 0 and last 5 binary symbols are 10000;

leading binary symbol is 0 and trailing 4 binary symbols are 0010; and

leading two binary symbols are 00 and trailing 4 binary symbols are 0001;

the 15 source data vectors being mapped to 15 encoded vectors having a disparity of minus 4 and a required positive starting disparity, the 15 encoded vectors having binary symbols identical to the 15 source data vectors with binary symbols 01 appended thereto.

Up to this point, 203 source vectors (196 data, 7 control) have been assigned to encoded vectors as listed in the tables of FIGS. 20-23, 25-27 & 30. 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. 8 and six from FIG. 9, 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. 32. The bold lines 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. 32. The complete set of vectors with individual source bit changes is listed in the table of FIG. 33. 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. It will be appreciated that in the table of FIG. 33, the fraction of encoded vectors having binary symbol changes compared to corresponding source data vectors correspond to those of the disparity independent encoded vectors having 8 binary symbols identical to the corresponding ones of the disparity independent encoded vectors and 2 binary symbols that are complements of the pair of appended binary symbols. Furthermore, it will be appreciated that the exemplary encoding scheme assigns a majority of the source data vectors corresponding to the fraction of encoded vectors having binary symbol changes, in complementary pairs of the majority, to corresponding pairs of the disparity independent encoded vectors that are complementary in all binary symbol positions except the appended binary symbols. Yet further, note that in the table of FIG. 33, at least eight source data vectors, having a leading run of four followed by two pairs of complementary binary symbols, are assigned to selected encoded vectors having two leading binary symbols complemented with respect to corresponding ones of the eight vectors. Still further, note that in the table of FIG. 33, at least eight source data vectors, having a leading run of four followed by a single binary symbol in four subsequent binary symbols which matches binary symbols in the leading run of four, are assigned to selected encoded vectors having a third binary symbol complemented with respect to corresponding ones of the eight vectors. Even further, note that in the table of FIG. 33, at least eight source data vectors, having a leading run of four followed by a single binary symbol in four subsequent binary symbols which does not match binary symbols in the leading run of four, are assigned to selected encoded vectors having a first, third and fourth binary symbol complemented with respect to corresponding ones of the eight vectors.

The table of FIGS. 34A-34G lists the selected vector assignments of the tables of FIGS. 20-23, 25-27, 30, and 33 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.

It may be preferable in many cases that the assignment of encoded vectors to the uncoded source vectors is done in a way such that the complexity of the encoder and/or decoder (typically, both) is minimized. It may be assumed that this will be accomplished if the number of classifications is minimized both for bit mapping and disparity control and if the number of bits changed for bit mapping is minimized as well. In the interest of logic circuit sharing, preference in this example is given to solutions with bit changes concentrated to fewer vectors over solutions with the same number of bit changes spread over more vectors. An important feature of this exemplary design is the containment of vectors with individual bit changes to a single class 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. 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.

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 of FIGS. 35-48, 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 diagrams discussed below. An asterisk (*) following the net name means that other expressions are included in the net.

The "a" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 35.

$$a = A \oplus \left\{ \left(D \oplus E' \cdot E \oplus F' \cdot G \oplus H + A \oplus G' \cdot G \oplus H' \cdot E \oplus F + D \oplus E \cdot E \oplus F' \cdot G \oplus H' + \right) \cdot \right. \\ \left. E \oplus F \cdot G \oplus H \right. \\ \left. A \oplus B' \cdot B \oplus C' \cdot C \oplus D' + \right. \quad (\text{Pn1})$$

$$\left. (C \oplus D \cdot A \cdot B + A' \cdot B' \cdot C' \cdot D) \cdot E' \cdot F \cdot G' \cdot H' \right\} \quad (\text{Pn3})$$

The "b" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 36.

$$b = B \oplus \left\{ \left(D \oplus E' \cdot E \oplus F' \cdot F \oplus G' \cdot G \oplus H' + C \oplus D' \cdot E \oplus F \cdot G \oplus H \right) \cdot \right. \\ \left. A \oplus B' \cdot B \oplus C' \cdot K' + \right. \quad (\text{Pn7})$$

$$\left. A \oplus B' \cdot B \oplus E' \cdot D \oplus E' \cdot E \oplus F' \cdot G \oplus H' \cdot (B \oplus C' + F \oplus G') + A' \cdot B' \cdot C' \cdot E' \cdot F \cdot G' \cdot H' \right\} (\text{Pn5} + \text{Pn8})$$

The "c" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 37.

$$c = C \oplus \left\{ A \oplus B \cdot A \oplus G' \cdot C \oplus D' \cdot C \oplus H' \cdot D \oplus E' \cdot E \oplus F' + \right. \quad (\text{Pn12})$$

$$\left. (C \oplus H' \cdot D \oplus E' \cdot E \oplus F' + E \oplus F \cdot G \oplus H' + E \oplus F' \cdot G \oplus H) \cdot A \oplus B' \cdot B \oplus C' \cdot C \oplus D' \right\} (\text{Pn11})$$

The "d" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the tables of FIG. 38.

$$d = D \oplus \left\{ (A \oplus G' \cdot B \oplus C \cdot G \oplus H' + A \oplus B \cdot C \oplus D') \cdot D \oplus E' \cdot E \oplus F' \cdot F \oplus G' + \right. \quad (\text{Pn20})$$

$$\left. (F \oplus G + G \oplus H) \cdot D \oplus E' \cdot E \oplus F' \cdot A \oplus B' \cdot B \oplus C' \cdot C \oplus D' + \right. \quad (\text{Pn19})$$

$$\left. (F \oplus G + E \oplus F) \cdot C \oplus H' \cdot G \oplus H' \cdot A \oplus B' \cdot B \oplus C' \cdot C \oplus D' + \right. \quad (\text{Pn15})$$

$$\left. A \cdot B' \cdot C \cdot D \cdot E \cdot F \cdot \overline{G'} \cdot H' \right\}$$

The "e" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 39.

$$e = E \oplus \left\{ (A \oplus B' \cdot F \oplus G' \cdot G \oplus H' + A \oplus B \cdot B \oplus C' \cdot G \oplus H + A \oplus B \cdot F \oplus G' \cdot G \oplus H) \cdot \right. \quad (\text{n21})$$

$$\left. C \oplus D' \cdot D \oplus E' \cdot E \oplus F' \right\}$$

The "f" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 40.

$$f = F \oplus \left\{ (A \oplus G + B \oplus C) \cdot A \oplus B' \cdot D \oplus E' \cdot E \oplus F' \cdot F \oplus G' \cdot G \oplus H' + \right. \quad (\text{Pn22})$$

$$A \oplus B \bullet B \oplus C' \bullet C \oplus D' \bullet D \oplus E' \bullet E \oplus F' \bullet G \oplus H' + \quad (\text{Pn24})$$

$$(\bar{A}' \bullet B' \bullet C' \bullet D + C \bullet D') \bullet A' \bullet E' \bullet F' \bullet G' \bullet H' + A \oplus C' \bullet B' \bullet D \bullet E \bullet F \bullet G' \bullet H' \} \quad (\text{Pn26})$$

The "g" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 41.

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

$$B \oplus C' \bullet C \oplus D' \bullet D \oplus E' \bullet E \oplus F' \bullet F \oplus G' \bullet G \oplus H' + \quad (\text{Pn30})$$

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

The "h" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 42.

$$h = H \oplus \{A \oplus C' \bullet B \oplus E \bullet D \oplus E' \bullet E \oplus F' \bullet F \oplus G' \bullet G \oplus H' + \quad (\text{n36})$$

$$A \bullet B' \bullet C \bullet D \bullet E \bullet F \bullet G' \bullet H' + \quad (\text{Pn37*})$$

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

The "i" column has bold entries in the tables of FIGS. 21 and 22 for the 87 vectors listed in the table of FIGS. 43A and 43B.

$$i = (E \oplus F' \bullet F \oplus G \bullet G \oplus H' + E \oplus F \bullet F \oplus G' \bullet G \oplus H + E \oplus F \bullet F \oplus G \bullet G \oplus H) \bullet \quad (\text{Pn42})$$

$$(A \oplus B' \bullet B \oplus C \bullet C \oplus D' + A \oplus B \bullet C \oplus D) + \quad (\text{Pn43})$$

$$(D \oplus E' \bullet E \oplus F \bullet F \oplus G' \bullet G \oplus H' + D \oplus E \bullet E \oplus F \bullet F \oplus G \bullet G \oplus H' +$$

$$D \oplus E \bullet E \oplus F' \bullet F \oplus G \bullet G \oplus H +$$

$$D \oplus E \bullet E \oplus F' \bullet F \oplus G' \bullet G \oplus H) \bullet (A \oplus B' \bullet B \oplus C \bullet C \oplus D + A \oplus B \bullet C \oplus D) + \quad (\text{n50})$$

$$F \oplus G \bullet K' \bullet D' \bullet E \bullet H' \bullet (A \bullet B' \bullet C' + A' \bullet B \bullet C' + A' \bullet B' \bullet C) + \quad (\text{Pn47})$$

$$(A \bullet B \bullet C' + A \bullet B' \bullet C + A' \bullet B \bullet C) \bullet D \bullet E' \bullet F' \bullet G' \bullet H' + \quad (\text{Pn49})$$

$$(A \oplus B' \bullet B \oplus C \bullet C \oplus D' + A \oplus B \bullet C \oplus D) \bullet \quad (\text{Pn43})$$

$$(E \bullet F' \bullet G' \bullet H' + E' \bullet F \bullet G' \bullet H' + E' \bullet F' \bullet G \bullet H') \quad (\text{Pn45})$$

The "j" column has bold entries in the tables of FIGS. 21, 23, 25, 26, 30, and 33 for the 129 vectors listed in the tables of FIGS. 44A-44D.

$$j = \{D \oplus E \bullet K' \bullet F' \bullet G + (D \bullet E \bullet F' + D \bullet E' \bullet F + D' \bullet E \bullet F) \bullet G' \} \bullet \quad (\text{n77})$$

$$(A' \bullet B' + A' \bullet C' + B' \bullet C') \bullet H' +$$

$$(A \oplus G \cdot B \oplus C \cdot D \oplus E' + A \oplus D \cdot B \oplus C' \cdot C \oplus H' + A \oplus B' \cdot B \oplus C' \cdot C \oplus D) \cdot E \oplus F' \cdot F \oplus G \cdot G \oplus H \quad (n76)$$

$$+ (A \cdot B \cdot C' + A \cdot B' \cdot C + A' \cdot B \cdot C) \cdot D' \cdot E \cdot F \cdot G \cdot H + \quad (n74)$$

$$(C \oplus D \cdot A \cdot B + A \oplus C' \cdot B' \cdot D) \cdot E \cdot F \cdot G' \cdot H' + (D' \cdot G + G') \cdot E' \cdot F' \cdot H' + \quad (n71+n68)$$

$$(B \oplus C' \cdot F \oplus G \cdot G \oplus H' + F \oplus G' \cdot K') \cdot C \oplus D' \cdot D \oplus E' \cdot E \oplus F' + \quad (n73)$$

$$(B \oplus E' \cdot C \oplus D + A \oplus G \cdot B \oplus C') \cdot A \oplus B' \cdot E \oplus F' \cdot F \oplus G' \cdot G \oplus H + \quad (n75)$$

$$E \oplus F \cdot A \cdot B \cdot C \cdot D' \cdot G \cdot H + E \oplus F \cdot D' \cdot G' \cdot H' + K' \cdot D \cdot E \cdot F \cdot G \cdot H + A \oplus B' \cdot B \oplus C' \cdot C \oplus D' \quad (n67^*)$$

$$+ (A' \cdot B' \cdot C' + A \cdot B \cdot \overline{C'} + A \cdot \overline{B'} \cdot C + B \cdot C) \cdot E \oplus F \cdot D \cdot G \cdot H \quad (n69)$$

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 the table of FIGS. 34A-34G. A total of 46 vectors listed in the tables of FIGS. 25, 27, and 30 require a positive entry disparity (PDRE). They are listed and sorted in the table of FIGS. 45A and 45B. Redundant bits are overlined. The equation for the positive required entry disparity PDRE can be written as follows:

$$PDRE = (D \oplus E \cdot F' \cdot G' \cdot H + E \oplus F \cdot D' \cdot G' \cdot H' + G \oplus H \cdot D' \cdot E' \cdot F) \cdot \quad (n85)$$

$$(A \cdot B' \cdot C' + A' \cdot B \cdot C' + A' \cdot B' \cdot C) + (D \cdot G' \cdot H' + D' \cdot G) \cdot B \oplus C \cdot A' \cdot E' \cdot F' + (n46+Pn88^*)$$

$$(E' \cdot F' \cdot G \cdot H + E \oplus F + G \oplus H) \cdot A' \cdot B' \cdot C' \cdot D + \quad (Pn87^*)$$

$$(A' + B') \cdot C \cdot D' \cdot E' \cdot F' \cdot G' \cdot H + (F' + H) \cdot K \quad (n99)$$

A total of 43 vectors listed in the tables of FIGS. 23 and 26 require a negative entry disparity (NDRE). They are listed and sorted in the table of FIGS. 46A and 46B. The equation for the negative required entry disparity NDRE can be written as follows:

$$NDRE = (D \oplus E \cdot F \cdot G \cdot H + D \oplus F \cdot E \cdot G \cdot H' + D \cdot E \cdot F' \cdot H + D \cdot E' \cdot F \cdot G) \cdot (n90)$$

$$(A \cdot B \cdot C' + A \cdot B' \cdot C \cdot A' \cdot B \cdot C) + \quad (n48)$$

$$(B \cdot C \cdot D' + B \cdot C' \cdot D + B' \cdot C \cdot D) \cdot A \cdot E \cdot F \cdot G' \cdot H + \quad (n96^*)$$

$$(E' + F') \cdot A \cdot B \cdot C \cdot D' \cdot G \cdot H + \quad (n92^*)$$

$$(E \oplus F \cdot G' \cdot H' + G \oplus H \cdot E' \cdot F' + E \oplus F \cdot G \oplus H) \cdot A \cdot B \cdot C \cdot D' \quad (Pn94^*)$$

$$(C \oplus D \cdot H' + D' \cdot H) \cdot A \cdot B \cdot E \cdot F \cdot G + \quad (Pn95^*)$$

$$A \oplus B \cdot C' \cdot D \cdot E \cdot F \cdot G + K \cdot C \cdot D \quad (\text{Pn97*})$$

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.

$$CMPLP10 = PDRE \cdot NDFBY + NDRE \cdot PDFBY$$

A total of 19 vectors listed in the table of FIG. 26 have a positive disparity of four. They are listed and sorted in the table of FIG. 47. The equation for the positive block disparity of four for encoding PDB can be written as follows:

$$PDB = (D \oplus E \cdot F + D \cdot E \cdot F') \cdot (A \cdot B \cdot C' + A \cdot B' \cdot C + A' \cdot B \cdot C) \cdot G \cdot H + \quad (\text{PDB1})$$

$$(B \cdot C \cdot D' + B \cdot C' \cdot D + B' \cdot C \cdot D) \cdot A \cdot E \cdot F \cdot G' \cdot H + \quad (\text{PDB2})$$

$$(C \oplus D \cdot H' + D' \cdot H) \cdot A \cdot B \cdot E \cdot F \cdot G + \quad (\text{PDB3})$$

$$E \oplus F \cdot A \cdot B \cdot C \cdot D' \cdot G \cdot H + A \oplus B \cdot C' \cdot D \cdot E \cdot F \cdot G \cdot H \quad (\text{PDB4+PDB5})$$

A total of 41 vectors listed in the tables of FIGS. 27 and 30 have a negative disparity of four. They are listed and sorted in the table of FIG. 48. Using the coding labels from the table, the equation for the negative block disparity of four for encoding NDB can be written as follows:

$$NDB = \{E \oplus F \cdot (G \cdot H) + E' \cdot F' \cdot (G + H)\} \cdot A' \cdot B' \cdot C' \cdot D + \quad (\text{NDB1})$$

$$(D \oplus E \cdot F' \cdot G' \cdot H + E \oplus F \cdot D' \cdot G' \cdot H' + G \oplus H \cdot D' \cdot E' \cdot F) \cdot (A \cdot B' \cdot C' + A' \cdot B \cdot C' + A' \cdot B' \cdot C) + \quad (\text{NDB2})$$

$$(D \cdot G' \cdot H' + D' \cdot G) \cdot B \oplus C \cdot A' \cdot E' \cdot F' + (A' + B') \cdot C \cdot D' \cdot E' \cdot F' \cdot G' \cdot H + K \cdot F' \quad (\text{NDB4+NDB3})$$

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 exemplary encoding circuitry can include a single latch (not shown) to keep track of the value of DFBY.

As noted above, 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 8B/10B-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 the tables of FIGS. 49 and 50. Note that every valid vector has a complement which is also valid and the respective vectors are listed side by side.

All but four of the vectors with $i \neq j$ can be paired with another valid vector which is identical in the first eight bits. The exception is illustrated in FIG. 18 and in the two rows of the table of FIGS. 50A and B with empty spaces in the $ij=10$ or $ij=01$ columns. The table of FIGS. 49A and B lists all 204 valid vectors with $i \neq j$:

- Primary vectors:
 - 60 from table of FIG. 22, 4 from table of FIG. 23, 4 from table of FIG. 25,
 - 18 from table of FIG. 26, 16 from table of FIG. 30, 60 from table of FIG. 33.
- The alternate vectors are complements of vectors listed as primary vectors:
 - 4 from table of FIG. 23, 4 from table of FIG. 25,

18 from table of FIG. 26, 16 from table of FIG. 30.

The 148 vectors with $i=j$ are listed in the table of FIGS. 50A and B arranged as 74 complementary vector pairs:

- Primary vectors:
 - 27 complementary primary vector pairs from the tables of FIGS. 20 and 21,
 - 20 primary vectors from the table of FIG. 23,
 - 1 primary vector each from the tables of FIGS. 25 and 26,
 - 25 primary vectors from the table of FIG. 27.
- Alternate vectors:
 - 20 from the table of FIG. 23,
 - 1 each from the tables of FIGS. 25 and 26,
 - 25 from the table of FIG. 27.

The equation for the validity of encoded vectors is composed from the valid labels of the tables of FIGS. 49 and 50. The expressions in parentheses at the right page edge refer to the net number of the logic associated with the logic expression at the left; in the case of those annotated with an asterisk, the logic expression on the left includes an additional term, namely, $\bullet i \oplus j'$ for $n0^*$, $\bullet i \oplus j$ for $n1^*$ and $n6^*$, and $\bullet d \oplus i \bullet i \oplus j'$ for $n8^*$.

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

$$(a \oplus b \bullet b \oplus c \bullet c \oplus d' + a \oplus b' \bullet c \oplus d) \bullet a \oplus e' \bullet e \oplus f' \bullet f \oplus g \bullet g \oplus h \bullet i \oplus j + \quad (\text{Pn30})$$

$$(e \oplus f' \bullet g \oplus h' \bullet h \oplus i + e \oplus f \bullet g \oplus h) \bullet a \oplus b' \bullet b \oplus c' \bullet c \oplus d \bullet d \oplus i' \bullet i \oplus j' + \quad (\text{Pn17})$$

$$(d \oplus e' \bullet e \oplus f' \bullet g \oplus h + d \oplus h' \bullet e \oplus f \bullet g \oplus h') \bullet a \oplus b' \bullet b \oplus c' \bullet c \oplus d \bullet i \oplus j \quad (\text{Pn31})$$

Disparity violations are not monitored for this example because they would in most cases not significantly help 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.

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.

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 of 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 of 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.

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 the tables of FIGS. 23, 25-27, and 30, and are tabulated in the table of FIGS. 51A and B. The equation for the complementation of the first eight bits of a vector is composed from the alternate vector labels of the table of FIGS. 51A and 51B.

$$COMPL8 = (a \oplus b' \cdot b \oplus g' + c \oplus d' \cdot d \oplus h') \cdot e \oplus f \cdot g \oplus h' \cdot i \cdot j' + \quad (n53)$$

$$(d \oplus h' \cdot g \oplus h' + d \oplus e' \cdot e \oplus f') \cdot a \oplus b' \cdot b \oplus c' \cdot i \cdot j' + \quad (n52)$$

$$(a \oplus b' \cdot b \oplus c' \cdot f \oplus g + a \oplus b' \cdot f \oplus g + a \oplus b' \cdot f \oplus g') \cdot a \oplus e' \cdot e \oplus f' \cdot g \oplus h' \cdot i \cdot j' + \quad (n51)$$

$$(d \oplus e' \cdot g \oplus h' \cdot j' + g \oplus h \cdot j) \cdot (a \oplus b' \cdot c \oplus d' + c \oplus d) \cdot e \oplus f' \cdot f \oplus g' \cdot i + \quad (n50)$$

$$(b \oplus c \cdot a' + a \cdot b' \cdot c') \cdot d' \cdot e' \cdot f \cdot i \cdot j + (\bar{e} \cdot f \cdot g' \cdot h' + g \oplus h) \cdot a' \cdot b' \cdot c' \cdot i \cdot j + (n56 + Pn59)$$

$$(a' \cdot b' + c' \cdot d') \cdot e \cdot f' \cdot h' \cdot i \cdot j + e' \cdot f' \cdot g' \cdot h' \cdot c \cdot d' \cdot j' + \quad (Pn41 + Pn42)$$

$$\{a \oplus b' \cdot b \oplus c' \cdot e' \cdot f' + a \cdot e \cdot f \cdot (b + d)\} \cdot g' \cdot h' \cdot i \cdot j + \quad (n58)$$

$$(a \cdot b + c \cdot d) \cdot e \oplus f \cdot g \oplus h' \cdot i \cdot j \quad (Pn57)$$

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

For decoding, the bold type bit values in the encoded columns of the table of FIG. 33 have to be complemented back to their original values as indicated 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 the table of FIG. 33 are the complements of those used in the encoding equations. In the tables of FIGS. 52 through 60, the common bit patterns are marked by bold type to logically classify the vectors by simple expressions. Redundant terms are overlined.

The "a" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 52. Using the decoding labels, the decoding equation for bit "A" can be written as follows:

$$A = a \oplus \{(d \oplus e \cdot e \oplus f' + b \oplus g' \cdot g \oplus h') \cdot a \oplus b \cdot b \oplus c \cdot c \oplus d' \cdot i' \cdot j' + \quad (n70)$$

$$a \oplus b \cdot b \oplus c' \cdot a \oplus e' \cdot e \oplus f' \cdot f \oplus g \cdot i' \cdot j + a \cdot b' \cdot d \cdot f' \cdot g \cdot h' \cdot i' \cdot j + \quad (\text{Pn66+Pn68})$$

$$a \oplus b' \cdot c \oplus d' \cdot g \oplus h \cdot i' \cdot j + a' \cdot b \cdot e \cdot f \cdot g' \cdot h' \cdot i' \cdot \bar{j} \} \quad (\text{Pn67+Pn69})$$

The "b" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 53. Using the decoding labels of the table of FIG. 53, the decoding equation for bit "B" can be written as follows:

$$B = b \oplus \{ (b \oplus c \cdot e \oplus f' \cdot g \oplus h' + b \oplus c' \cdot e \oplus f) \cdot a \oplus b \cdot d \oplus e \cdot f \oplus g \cdot i' \cdot j + \quad (\text{n72})$$

$$a \oplus b \cdot b \oplus g' \cdot e \oplus f \cdot f \oplus g \cdot g \oplus h \cdot i' \cdot j + \quad (\text{Pn74})$$

$$a \oplus b' \cdot c \oplus d' \cdot g \oplus h \cdot i' \cdot j + a' \cdot b \cdot c' \cdot d \cdot e' \cdot g \cdot i' \cdot j \} \quad (\text{Pn67+Pn75})$$

The "c" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 54. Using these decoding labels, the decoding equation for bit "C" can be written as follows:

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

$$(c \oplus d \cdot e \oplus f \cdot f \oplus g + c \oplus d' \cdot e \oplus f' \cdot f \oplus g') \cdot d \oplus e \cdot g \oplus h \cdot i' \cdot j \} \quad (\text{n78})$$

The "d" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 55. Using these decoding labels, the decoding equation for bit "D" can be written as follows:

$$D = d \oplus \{ (d \oplus e \cdot e \oplus f' + b \oplus g' \cdot g \oplus h') \cdot a \oplus b \cdot b \oplus c \cdot c \oplus d' \cdot i' \cdot j + \quad (\text{n70})$$

$$(d \oplus e' \cdot e \oplus f \cdot f \oplus g' + a \oplus e' \cdot e \oplus f' \cdot f \oplus g) \cdot a \oplus b \cdot c \oplus d \cdot i' \cdot j + \quad (\text{n81})$$

$$(b \oplus g \cdot f \oplus g' + c \oplus d' \cdot b \oplus g) \cdot a \oplus b \cdot b \oplus c' \cdot d \oplus e \cdot i' \cdot j + \quad (\text{n83})$$

$$a \cdot b' \cdot c \cdot d' \cdot e \cdot g' \cdot i' \cdot j \} \quad (\text{n84})$$

The "e" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 56. Using these decoding labels, the decoding equation for bit "E" can be written as follows:

$$E = e \oplus \{ (d \oplus e \cdot f \oplus g + d \oplus e' \cdot f \oplus g') \cdot c \oplus d \cdot e \oplus f \cdot g \oplus h \cdot i' \cdot j + \quad (\text{n86})$$

$$a \oplus b' \cdot \overline{b \oplus c} \cdot c \oplus d' \cdot d \oplus e \cdot \overline{a \oplus e'} \cdot e \oplus f' \cdot f \oplus g \cdot i' \cdot j \} \quad (\text{n87})$$

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

The "F" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 57. Using these decoding labels, the decoding equation for bit "F" can be written as follows:

$$F = f \oplus \{a \oplus b \oplus c \oplus g \oplus e \oplus f \oplus g' \oplus i' \oplus j + \quad (n90)$$

$$a \oplus d \oplus b \oplus g \oplus c \oplus d' \oplus f \oplus g \oplus g \oplus h' \oplus i' \oplus j + \quad (Pn91)$$

$$a \oplus b \oplus c \oplus d \oplus c \oplus g \oplus e \oplus f \oplus i' \oplus j\} \quad (Pn92)$$

The "g" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 58. Using these decoding labels, the decoding equation for bit "G" can be written as follows:

$$G = g \oplus \{b \oplus g' \oplus c \oplus d' \oplus d \oplus h \oplus e \oplus f' \oplus f \oplus g \oplus i' \oplus j + \quad (Pn94)$$

$$b \oplus c' \oplus c \oplus d \oplus d \oplus e \oplus e \oplus f \oplus g \oplus h \oplus i' \oplus j + \quad (Pn95)$$

$$a \oplus e' \oplus e \oplus f \oplus c' \oplus d \oplus g \oplus h' \oplus i' \oplus j + a \oplus b' \oplus e' \oplus f' \oplus g \oplus h \oplus i' \oplus j\} \quad (n96+Pn97)$$

The "h" column has bold entries in the tables of FIGS. 33 and 34 for the vectors listed in the table of FIG. 59. Using these decoding labels, the decoding equation for bit "H" can be written as follows:

$$H = h \oplus \{a \oplus d \oplus c \oplus g \oplus d \oplus f \oplus e \oplus g' \oplus g \oplus h \oplus i' \oplus j + \quad (Pn100)$$

$$a \oplus e' \oplus e \oplus f \oplus c \oplus d' \oplus g' \oplus h \oplus i' \oplus j + a \oplus b' \oplus e' \oplus f' \oplus g \oplus h \oplus i' \oplus j\} \quad (n101+Pn97)$$

The K-bit value for all vectors of the tables of FIGS. 52-59 is zero. The seven encoded control characters which have a K-value of one are listed in the table of FIG. 60. All seven control characters have alternate representations. It will be appreciated that the table of FIG. 7 shows the results of the mapping of 7 control vectors to 7 disparity dependent encoded control vectors having primary representations and alternate representations complementary to the primary representations, the encoded control vectors including the 7 control vectors plus two appended binary symbols, the primary representations of the encoded control vectors satisfying at least one of:

- (i) disparity equals minus four, (ii) required starting disparity is positive, (iii) third, fifth, sixth, seventh and ninth binary symbols have a value of zero and first binary symbol has a value of one, (iv) fourth and eighth binary symbols are complementary, and (v) eighth and tenth binary symbols are complementary;

(i) disparity equals minus four, (ii) required starting disparity is positive, (iii) fourth, sixth, eighth, ninth and tenth binary symbols have a value of zero and fifth and seventh binary symbols have a value of one;

(i) disparity equals zero, (ii) required starting disparity is negative, (iii) first two binary symbols and last two binary symbols have a value of zero and third, fourth, fifth, sixth and seventh binary symbols have a value of one; and

(i) disparity equals zero, (ii) required starting disparity is positive, (iii) first two binary symbols and last two binary symbols have a value of zero and fourth, fifth, sixth and seventh and eighth binary symbols have a value of one.

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. Using these decoding labels, the decoding equation for bit "K" can be written as follows:

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

(PK)

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 lead to different ranges in circuit sharing and circuit counts and thus the exemplary circuit may not necessarily be minimum area. Another reason for

variation in circuit implementation and opportunity to somewhat enhance the exemplary 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 in the event that simulation results are available. Similarly, the circuit diagrams show no complex gates to allow maximum circuit sharing; logic processing programs can be employed that will introduce complex gates automatically where appropriate. Note that some of the logic variables of the equations are not present explicitly in the circuit diagrams. If so, they have been merged with other functions in a single gate to reduce overall circuit delay.

The block diagram for the encoding circuit with all inputs and outputs is shown in FIG. 61. A gate-level circuit diagram of the encoder is shown in FIGS. 62A-C which represent a single circuit with net sharing. Fig. 62A 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. 62B, and FIG. 62C 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.

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. 62C, 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 or fourth level which facilitates pipelining of this logic path into the next cycle.

The encoder includes 298 gates and can include a flip-flop (not shown) to keep track of the disparity. No logic path exceeds 7 gates; all gates are of the inverting type with shorter delay except some XOR gates which for most power and loading levels have comparable or only slightly more delay than XNOR gates.

The circuit presented has been structured for easy forward pipelining 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 nonpipelined 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.

The block diagram for an exemplary decoding circuit with all inputs and outputs is shown in FIG. 63. A gate-level circuit diagram of the decoder is shown in FIGS. 64A-C, representing a single circuit with net sharing. FIG. 64A shows the vector validity

checks. The circuit which controls the vector complementation signal (COMPL10) is shown in FIG. 64B. 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. 64C 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.

The decoder includes 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 with a NAND3 gate, and the PK path is five logic levels.

For fast operation, the circuit presented has been structured for easy forward pipelining 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. 64C 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 pipelining requires 23 extra latches. For a reduction of the first step to four gating levels, a total of 48 pipelining 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.

For verification purposes, a soft, technology independent macro was 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.

As noted above, a variety of techniques utilizing dedicated hardware, general purpose processors, firmware, software, or a combination of the foregoing may be employed to implement the present invention, in addition to the preferred implementation in hardware using logic gates. With reference to FIG. 65, such alternate implementations might employ, for example, a processor 6502, a memory 6504, and an input/output interface formed, for example, by a display 6506 and a keyboard 6508. The term "processor" as used herein is intended to include any processing device, such as, for example, one that includes a CPU (central processing unit) and/or other forms of processing circuitry. Further, the term "processor" may refer to more than one individual processor. The term "memory" is intended to include memory associated with a processor or CPU, such as, for example, RAM (random access memory), ROM (read only memory), a fixed memory device (e.g., hard drive), a removable memory device (e.g., diskette), a flash memory and the like. In addition, the phrase "input/output interface" as used herein, is intended to include, for example, one or more mechanisms for inputting data to the processing unit (e.g., mouse), and one or more mechanisms for providing results associated with the processing unit (e.g., printer). The processor 6502, memory 6504, and input/output interface such as display 6506 and keyboard 6508 can be interconnected, for example, via bus 6510 as part of a data processing unit 6512. Suitable interconnections, for example via bus 6510, can also be provided to a network interface 6514, such as a network card, which can be provided to interface with a computer network, and to a media interface 6516, such as a diskette or CD-ROM drive, which can be provided to interface with medium 6518.

Accordingly, computer software including instructions or code for performing the methodologies of the invention, as described herein, may be stored in one or more of the associated memory devices (e.g., ROM, fixed or removable memory) and, when ready to be utilized, loaded in part or in whole (e.g., into RAM) and executed by a CPU. Such software could include, but is not limited to, firmware, resident software, microcode, and the like. Note that implementations of one or more embodiments of the present invention

involving software may take advantage of the potential for parallelism described above to employ, for example, a vectorized or parallelized solution.

Furthermore, the invention can take the form of a computer program product accessible from a computer-usable or computer-readable medium (e.g., medium 6518) providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer usable or computer readable medium can be any apparatus for use by or in connection with the instruction execution system, apparatus, or device.

The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid-state memory (e.g. memory 6504), magnetic tape, a removable computer diskette (e.g. media 6518), a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk-read only memory (CD-ROM), compact disk-read/write (CD-R/W) and DVD.

A data processing system suitable for storing and/or executing program code will include at least one processor 6502 coupled directly or indirectly to memory elements 6504 through a system bus 6510. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

Input/output or I/O devices (including but not limited to keyboards 6508, displays 6506, pointing devices, and the like) can be coupled to the system either directly (such as via bus 6510) or through intervening I/O controllers (omitted for clarity).

Network adapters such as network interface 6514 may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

In any case, it should be understood that the components illustrated herein may be implemented in various forms of hardware, software, or combinations thereof, e.g., application specific integrated circuit(s) (ASICs), functional circuitry, one or more appropriately programmed general purpose digital computers with associated memory, one or more programmable logic arrays (PLAs), combinational logic as described herein, and the like. Given the teachings of the invention provided herein, one of ordinary skill in the related art will be able to contemplate other implementations of the components of the invention. It should of course be noted that an encoding scheme can be implemented via a look-up table.

Although illustrative embodiments of the present invention have been described herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various other changes and modifications may be made by one skilled in the art without departing from the scope or spirit of the invention.

Claims

What is claimed is:

1. A method of encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, $M > N > 0$, comprising the steps of:

obtaining a plurality of NB source data vectors; and

encoding said NB source data vectors into a plurality of MB encoded vectors according to an encoding scheme, said encoding scheme mapping at least a first portion of said NB source data vectors into MB encoded vectors comprising disparity independent encoded vectors, said encoding scheme mapping at least a second portion of said NB source data vectors into MB encoded vectors comprising disparity dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said MB encoded vectors having M-N binary symbols appended thereto by said encoding scheme, a fraction of said MB encoded vectors having binary symbol changes, other than whole-vector complementation, compared to corresponding ones of said NB source data vectors;

wherein said encoding scheme is preselected to perform at least one of:

(i) reducing, compared to at least some other possible NB to MB encoding schemes, and

(ii) substantially eliminating

an amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors.

2. The method of claim 1, wherein $N=8$, $M=10$, and said encoding scheme substantially eliminates said amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol

changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors.

3. The method of claim 2, wherein:

said encoding scheme produces a DC-balanced transmission code; and

said first portion of said NB source data vectors comprises at least 60 balanced source data vectors having a leading run-length no greater than 2 that are appended with a complementary pair of appended binary symbols, corresponding to said M-N binary symbols, when encoded into corresponding ones of said disparity independent encoded vectors.

4. The method of claim 3, wherein said fraction of said MB encoded vectors having binary symbol changes compared to said corresponding ones of said NB source data vectors correspond to those of said disparity independent encoded vectors having 8 binary symbols identical to said corresponding ones of said disparity independent encoded vectors and 2 binary symbols that are complements of said pair of appended binary symbols.

5. The method of claim 4, wherein:

said encoding scheme assigns a majority of said NB source data vectors corresponding to said fraction of said MB encoded vectors having binary symbol changes, in complementary pairs of said majority, to corresponding pairs of said disparity independent MB encoded vectors that are complementary in all binary symbol positions except said appended binary symbols;

said encoding scheme assigns at least eight vectors of said first portion of said NB source data vectors, having a leading run of four followed by two pairs of complementary binary symbols, to selected ones of said fraction of said MB encoded vectors having two leading binary symbols complemented with respect to corresponding ones of said eight vectors;

said encoding scheme assigns at least eight vectors of said first portion of said NB source data vectors, having a leading run of four followed by a single binary symbol in four subsequent binary symbols, where said single binary symbol matches

binary symbols in said leading run of four, to selected ones of said fraction of said MB encoded vectors having a third binary symbol complemented with respect to corresponding ones of said eight vectors; and

said encoding scheme assigns at least eight vectors of said first portion of said NB source data vectors, having a leading run of four followed by a single binary symbol in four subsequent binary symbols, where said single binary symbol does not match binary symbols in said leading run of four, to selected ones of said fraction of said MB encoded vectors having a first, third and fourth binary symbols complemented with respect to corresponding ones of said eight vectors.

6. The method of claim 2, wherein:

said encoding scheme produces a DC-balanced transmission code;

said first portion of said NB source data vectors comprises:

27 source data vectors having an overall disparity of plus 2, and

27 source data vectors having an overall disparity of minus 2, and having a running disparity of at most 2 after any given binary symbol position, that are appended with binary symbols 00, for those 27 source data vectors having an overall disparity of plus 2, and binary symbols 11, for those of said 27 source data vectors having an overall disparity of minus 2, when encoded into corresponding ones of said disparity independent encoded vectors; and

said second portion of said NB source data vectors comprises:

at least 19 source data vectors having a disparity of plus 2 and satisfying at least one of:

ending in binary symbols 11001;

starting with binary symbols 1110 and ending with binary symbol 1; and

ending with binary symbols 10 and having one of: (i) at most three leading ones, and (ii) at most one leading zero;

said 19 NB source data vectors being mapped to 19 disparity dependent balanced encoded vectors having a required negative starting disparity, said encoded

vectors having binary symbols identical to said source data vectors with two zeroes appended thereto;

at least 4 balanced source data vectors starting with binary symbols 1110, said 4 balanced source data vectors starting with binary symbols 1110 being mapped to 4 disparity dependent balanced encoded vectors having a required negative starting disparity and having binary symbols identical to said 4 balanced source data vectors with binary symbols 01 appended thereto; and

at least 4 balanced source data vectors starting with binary symbols 0001, said 4 balanced source data vectors starting with binary symbols 0001 being mapped to 4 disparity dependent balanced encoded vectors having a required positive starting disparity and having binary symbols identical to said source data vectors with binary symbols 01 appended thereto.

7. The method of claim 6, wherein said second portion of said NB source data vectors further comprises:

at least 18 source data vectors having a disparity of plus 4 and satisfying one of:
trailing 4 binary symbols comprise a complementary pair of binary symbols followed by binary symbols 11;

trailing 4 bay symbols are 1111 and leading 4 binary symbols are one of: (i) two pairs of complementary binary symbols, and (ii) 1100;

leading binary symbol is 1 and trailing 4 binary symbols are 1101; and
leading two binary symbols are 1 and trailing 4 binary symbols are 1110;

said 18 NB source data vectors being mapped to 18 encoded vectors having a disparity of plus 4 and a required negative starting disparity, said encoded vectors having binary symbols identical to said 18 source data vectors with 01 appended thereto;

at least one source data vector in the form 11101111, said one source data vector being mapped to one encoded vector having a disparity of plus 4 and a required negative starting disparity, said encoded vector having the form 1110111100;

at least 21 vectors having a disparity of minus 2 and satisfying one of:
 trailing 4 binary symbols comprise 01 followed by a pair of
 complementary binary symbols;
 trailing binary symbols are one of 1001 and 11010;
 leading binary symbol is 0 and trailing 4 binary symbols are 0011;
 trailing 5 binary symbols are 10001; and
 trailing 6 binary symbols are 100001;
 said 21 NB source data vectors being mapped to 21 encoded vectors having a
 disparity of minus 4 and a required positive starting disparity, said 21 encoded vectors
 having binary symbols identical to said 21 source data vectors with binary symbols 00
 appended thereto; and
 at least 15 vectors having a disparity of minus 4 and satisfying one of:
 trailing 4 binary symbols comprise a pair of complementary binary
 symbols followed by binary symbols 00;
 first binary symbol is 0 and last 5 binary symbols are 10000;
 leading binary symbol is 0 and trailing 4 binary symbols are 0010; and
 leading two binary symbols are 00 and trailing 4 binary symbols are
 0001;
 said 15 NB source data vectors being mapped to 15 encoded vectors having a
 disparity of minus 4 and a required positive starting disparity, said 15 encoded vectors
 having binary symbols identical to said 15 source data vectors with binary symbols 01
 appended thereto.

8. The method of claim 2, wherein:
 - said encoding scheme produces a DC-balanced transmission code; and
 - said encoding scheme further assigns at least 7 NB vectors, as control vectors,
 to at least 7 MB disparity dependent corresponding encoded control vectors having
 primary representations and alternate representations complementary to said primary
 representations, said encoded control vectors comprising said 7 NB vectors plus two

appended binary symbols, said primary representations of said 7 MD corresponding encoded control vectors satisfying at least one of:

(i) disparity equals minus four, (ii) required starting disparity is positive, (iii) third, fifth, sixth, seventh and ninth binary symbols have a value of zero and first binary symbol has a value of one, (iv) fourth and eighth binary symbols are complementary, and (v) eighth and tenth binary symbols are complementary;

(i) disparity equals minus four, (ii) required starting disparity is positive, (iii) fourth, sixth, eighth, ninth and tenth binary symbols have a value of zero and fifth and seventh binary symbols have a value of one;

(i) disparity equals zero, (ii) required starting disparity is negative, (iii) first two binary symbols and last two binary symbols have a value of zero and third, fourth, fifth, sixth and seventh binary symbols have a value of one; and

(i) disparity equals zero, (ii) required starting disparity is positive, (iii) first two binary symbols and last two binary symbols have a value of zero and fourth, fifth, sixth and seventh and eighth binary symbols have a value of one.

9. The method of claim 8, wherein each of said 7 NB control vectors and each of said 7 MB corresponding encoded control vectors has a complementary alternate vector.

10. The method of claim 2, wherein:

said encoding scheme produces a DC-balanced transmission code; and

said primary representations of said disparity dependent encoded vectors end with one of two 2-binary symbol patterns.

11. The method of claim 1, wherein said encoding step comprises the sub-steps of:
appending M-N binary symbols to said NB source data vectors to obtain augmented vectors;

complementing M binary symbols of a given one of said augmented vectors to obtain one of said alternate representations of one of said MB encoded vectors

corresponding to a given one of said NB source data vectors, of said second portion of said NB source data vectors, from which said given one of said augmented vectors was obtained, said complementation of said M binary symbols being at least in part responsive to a determination that said one of said MB encoded vectors comprises one of said disparity dependent encoded vectors and a current running disparity does not match a required starting disparity for said one of said MB encoded vectors; and

complementing less than N binary symbols of another given one of said NB source data vectors to obtain a corresponding portion of another given one of said MB encoded vectors corresponding to said other given one of said NB source data vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said MB encoded vectors comprises one of said disparity-independent encoded vectors, said less than N binary symbol complementing and said N binary symbol complementing steps being performed substantially in parallel.

12. A method of decoding M-binary symbol (MB) encoded vectors into decoded N-binary symbol (NB) source data vectors, $M > N > 0$, comprising the steps of:

obtaining a plurality of MB encoded vectors that were encoded from a plurality of NB source data vectors according to an encoding scheme, said encoding scheme mapping at least a first portion of said NB source data vectors into MB encoded vectors comprising disparity independent encoded vectors, said encoding scheme mapping at least a second portion of said NB source data vectors into MB encoded vectors comprising disparity dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said MB encoded vectors having M-N binary symbols appended thereto by said encoding scheme, a fraction of said MB encoded vectors having binary symbol changes, other than whole-vector complementation, compared to corresponding ones of said NB source data vectors, said encoding scheme being preselected to perform at least one of:

(i) reducing, compared to at least some other possible NB to MB encoding schemes, and

(ii) substantially eliminating
an amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors; and

decoding said MB encoded vectors into a plurality of NB source data vectors according to decoding rules of said encoding scheme.

13. The method of claim 12, wherein $N=8$, $M=10$, and said encoding scheme substantially eliminates said amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors.

14. The method of claim 13, wherein:
said encoding scheme produces a DC-balanced transmission code; and
said primary representations of said disparity dependent encoded vectors end with one of two 2-binary symbol patterns.

15. The method of claim 12, wherein said decoding step comprises the sub-steps of:

complementing at least N binary symbols of a given one of said MB encoded vectors to recover a given one of said NB source data vectors, of said second portion of said NB source data vectors, corresponding to said given one of said MB encoded vectors, said complementation of said at least N binary symbols being at least in part responsive to a determination that said given one of said MB encoded vectors comprises one of said alternate versions of one of said disparity-dependent encoded vectors; and

complementing less than N binary symbols of another given one of said MB encoded vectors to recover a corresponding portion of another given one of said NB source data vectors, of said first portion of said NB source data vectors, corresponding to said other given one of said MB encoded vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said MB encoded vectors comprises one of said disparity-independent encoded vectors, said less than N binary symbol complementing and said N binary symbol complementing being performed substantially in parallel.

16. An apparatus for encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, $M > N > 0$, at least some of said MB encoded vectors comprising disparity-dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said apparatus comprising:

- a binary symbol appending module configured to append M-N binary symbols to said NB source data vectors to obtain augmented vectors;

- a full vector complementing module configured to complement M binary symbols of a given one of said augmented vectors to obtain one of said alternate representations of one of said MB encoded vectors corresponding to a given one of said NB source data vectors from which said given one of said augmented vectors was obtained, said complementation of said M binary symbols being at least in part responsive to a determination that said one of said MB encoded vectors comprises one of said disparity dependent encoded vectors and a current running disparity does not match a required starting disparity for said one of said MB encoded vectors; and

- a binary symbol complementing module configured to complement less than N binary symbols of another given one of said NB source data vectors to obtain a corresponding portion of another given one of said MB encoded vectors corresponding to said other given one of said NB source data vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said MB encoded vectors comprises a

disparity-independent encoded vector, said binary symbol complementing module and said full vector complementing module being configured to operate substantially in parallel, said binary symbol complementing module and said full vector complementing module being coupled to each other and being configured to implement an encoding scheme wherein, in assignment of MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity-independent.

17. The apparatus of claim 16, wherein $N=8$, $M=10$, and said encoding scheme assigns substantially all of said MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors to MB encoded vectors that are balanced and disparity independent.

18. The apparatus of claim 16, further comprising a disparity monitoring module coupled to said full vector complementing module, said disparity classifier being configured to determine current running disparity for use in assigning proper ones of said disparity dependent encoded vectors to given ones of said NB source data vectors.

19. The apparatus of claim 16, wherein said apparatus is implemented as an array of logic gates.

20. An apparatus for decoding M-binary symbol (MB) encoded vectors into N-binary symbol (NB) source data vectors, $M>N>0$, at least some of said MB encoded vectors comprising disparity-dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said apparatus comprising:

a full vector complementing module configured to complement at least N binary symbols of a given one of said MB encoded vectors to recover a given one of

said NB source data vectors corresponding to said given one of said MB encoded vectors, said complementation of said at least N binary symbols being at least in part responsive to a determination that said given one of said MB encoded vectors comprises one of said alternate versions of one of said disparity-dependent encoded vectors; and

a binary symbol complementing module coupled to said full vector complementing module and configured to complement less than N binary symbols of another given one of said MB encoded vectors to recover a corresponding portion of another given one of said NB source data vectors corresponding to said other given one of said MB encoded vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said MB encoded vectors comprises a disparity-independent encoded vector, said binary symbol complementing module and said full vector complementing module being configured to operate substantially in parallel, said binary symbol complementing module and said full vector complementing module being configured to implement an encoding scheme wherein, in assignment of MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity-independent.

21. The apparatus of claim 20, further comprising a validity check module coupled to said full vector complementing module and said binary symbol complementing module, said validity checker being configured to obtain putative encoded vectors and to determine whether given ones of said putative encoded vectors are valid ones of said MB encoded vectors.

22. The apparatus of claim 20, wherein $N=8$, $M=10$, and said encoding scheme assigns substantially all of said MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors to MB encoded vectors that are balanced and disparity independent.

23. The apparatus of claim 20, wherein said apparatus is implemented as an array of logic gates.

24. A method for encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, $M > N > 0$, at least some of said MB encoded vectors comprising disparity-dependent encoded vectors having primary representations and alternate representations complementary to said primary, said method comprising the steps of:

appending M-N binary symbols to said NB source data vectors to obtain augmented vectors;

complementing M binary symbols of a given one of said augmented vectors to obtain one of said alternate representations of one of said NB encoded vectors corresponding to a given one of said NB source data vectors from which said given one of said augmented vectors was obtained, said complementation of said M binary symbols being at least in part responsive to a determination that said one of said MB encoded vectors comprises one of said disparity dependent encoded vectors and a current running disparity does not match a required starting disparity for said one of said MB encoded vectors; and

complementing less than N binary symbols of another given one of said NB source data vectors to obtain a corresponding portion of another given one of said MB encoded vectors corresponding to said other given one of said NB source data vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said MB encoded vectors comprises a disparity-independent encoded vector, said less than N binary symbol complementing and said N binary symbol complementing steps being performed substantially in parallel and implementing an encoding scheme wherein, in assignment of MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity-independent.

25. The method of claim 24, wherein $N=8$, $M=10$, and said encoding scheme assigns substantially all of said MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors to MB encoded vectors that are balanced and disparity independent.

26. A method for decoding M-binary symbol (MB) encoded vectors into N-binary symbol (NB) source data vectors, $M>N>0$, at least some of said MB encoded vectors comprising disparity-dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said method comprising:

complementing at least N binary symbols of a given one of said MB encoded vectors to recover a given one of said NB source data vectors corresponding to said given one of said MB encoded vectors, said complementation of said at least N binary symbols being at least in part responsive to a determination that said given one of said MB encoded vectors comprises one of said alternate versions of one of said disparity-dependent encoded vectors; and

complementing less than N binary symbols of another given one of said MB encoded vectors to recover a corresponding portion of another given one of said NB source data vectors corresponding to said other given one of said MB encoded vectors, said complementation of said less than N binary symbols being at least in part responsive to a determination that said other given one of said NB encoded vectors comprises a disparity-independent encoded vector, said less than N binary symbol complementing and said N binary symbol complementing being performed substantially in parallel and implementing an encoding scheme wherein, in assignment of MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors, preference is given to MB encoded vectors that are balanced and disparity-independent.

27. The method of claim 26, wherein $N=8$, $M=10$, and said encoding scheme assigns substantially all of said MB encoded vectors having binary symbol changes compared to corresponding ones of said NB source data vectors to MB encoded vectors that are balanced and disparity independent.

28. A computer program product comprising a computer useable medium including computer usable program code for encoding N-binary symbol (NB) source data vectors into M-binary symbol (MB) encoded vectors, $M>N>0$, said computer program product including:

computer usable program code for obtaining a plurality of NB source data vectors; and

computer usable program code for encoding said NB source data vectors into a plurality of MB encoded vectors according to an encoding scheme, said encoding scheme mapping at least a first portion of said NB source data vectors into MB encoded vectors comprising disparity independent encoded vectors, said encoding scheme mapping at least a second portion of said NB source data vectors into MB encoded vectors comprising disparity dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said MB encoded vectors having M-N binary symbols appended thereto by said encoding scheme, a fraction of said MB encoded vectors having binary symbol changes, other than whole-vector complementation, compared to corresponding ones of said NB source data vectors;

wherein said encoding scheme is preselected to perform at least one of:

(i) reducing, compared to at least some other possible NB to MB encoding schemes, and

(ii) substantially eliminating

an amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors.

29. A computer program product comprising a computer useable medium including computer usable program code for decoding M-binary symbol (MB) encoded vectors into decoded N-binary symbol (NB) source data vectors, $M > N > 0$, said computer program product including:

computer usable program code for obtaining a plurality of MB encoded vectors that were encoded from a plurality of NB source data vectors according to an encoding scheme, said encoding scheme mapping at least a first portion of said NB source data vectors into MB encoded vectors comprising disparity independent encoded vectors, said encoding scheme mapping at least a second portion of said NB source data vectors into MB encoded vectors comprising disparity dependent encoded vectors having primary representations and alternate representations complementary to said primary representations, said MB encoded vectors having M-N binary symbols appended thereto by said encoding scheme, a fraction of said MB encoded vectors having binary symbol changes, other than whole-vector complementation, compared to corresponding ones of said NB source data vectors, said encoding scheme being preselected to perform at least one of:

(i) reducing, compared to at least some other possible NB to MB encoding schemes, and

(ii) substantially eliminating

an amount of said disparity dependent encoded vectors that fall within said fraction of said MB encoded vectors having said binary symbol changes, other than whole-vector complementation, compared to said corresponding ones of said NB source data vectors; and

computer usable program code for decoding said MB encoded vectors into a plurality of NB source data vectors according to decoding rules of said encoding scheme.