2004-01-01

A system for the correction of missing or deleted symbols

Martin Tomlinson

Jing Cai

C. J. Tjhai

Marcel Ambroze

Mohammed Zaiki Ahmed

A system for the correction of missing or deleted symbols

Tomlinson, Martin; Cai, Jing; Tjhai, C. J.; Ambroze, Marcel; Ahmed, Mohammed Zaiki

Publication date:
2004

(12) UK Patent Application (19) GB (11) 2 421 599 (13) A

(43) Date of A Publication 28.06.2006

(21) Application No: 0428042.6

(22) Date of Filing: 22.12.2004

(71) Applicant(s):
Martin Tomlinson
The Old Coach House, Tristford,
Harberton, TOTNES, Devon, TQ9 7RZ,
United Kingdom

Jing Cai
No.13A, North Road East, PLYMOUTH,
Devon, PL4 6AS, United Kingdom

(continued on next page)

(51) INT CL:
G06F 11/08 (2006.01) G06F 11/10 (2006.01)

(52) UK CL (Edition X):
G4A AEE APCH

(56) Documents Cited:
EP 0903955 A1 **US 5115436 A**
US 4555784 A

(58) Field of Search:
UK CL (Edition X) **G4A**
INT CL **G06F**
Other:

(54) Abstract Title: **Correction for missing or deleted symbols using parity check equations**

(57) In many communications systems, broadcasting systems and storage systems, information is encoded as a sequence of symbols. The total number of symbols transmitted or stored is denoted by n, and these are obtained by encoding k information symbols using parity check equations from an error correcting code (ECC). The present invention allows the k information symbols to be recovered before all n symbols have been received or read, thereby compensating for missing symbols and allowing faster access to the information. Three decoding methods using parity check equations to recover missing symbols are described, each of a different degree of complexity. It is shown that the present invention can cope with a higher number of missing symbols than the current state of the art.

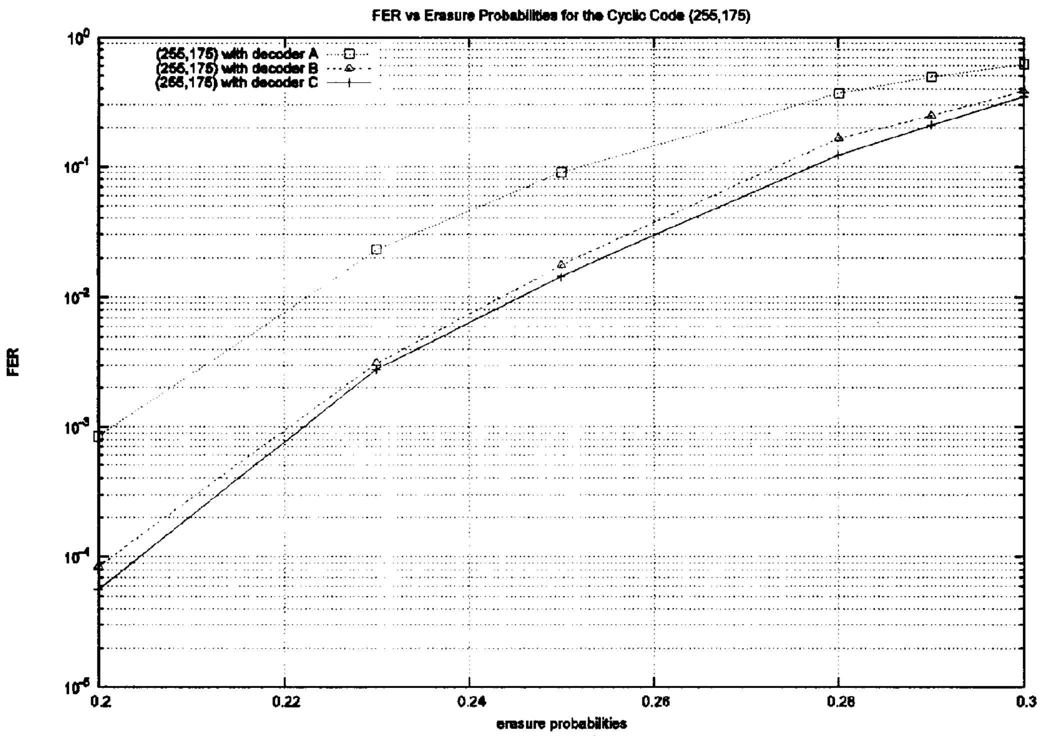

Fig 5 Performance of Methods A,B and C as a function of code and erasure bit probability

GB 2 421 599 A

GB 2421599 A continuation

(71) cont

C.J Tjhai
Room A20IB, St. Thomas Court,
20 Gasking Street, PLYMOUTH, PL4 9AP,
United Kingdom

Marcel Arabroth
4B Pier Street, PLYMOUTH, PL1 3BS,
United Kingdom

Mohammed Zaiki Ahmed
226 Ladysmith Rd, PLYMOUTH, PL4 7NR,
United Kingdom

(72) Inventor(s):

Martin Tomlinson
Jing Cai
C.J Tjhai
Marcel Arabroth
Mohammed Zaiki Ahmed

(74) Agent and/or Address for Service:

Martin Tomlinson
The Coach House, Tristford, Harberton,
TOTNES, Devon, TQ9 7RZ,
United Kingdom

DIAGRAMS

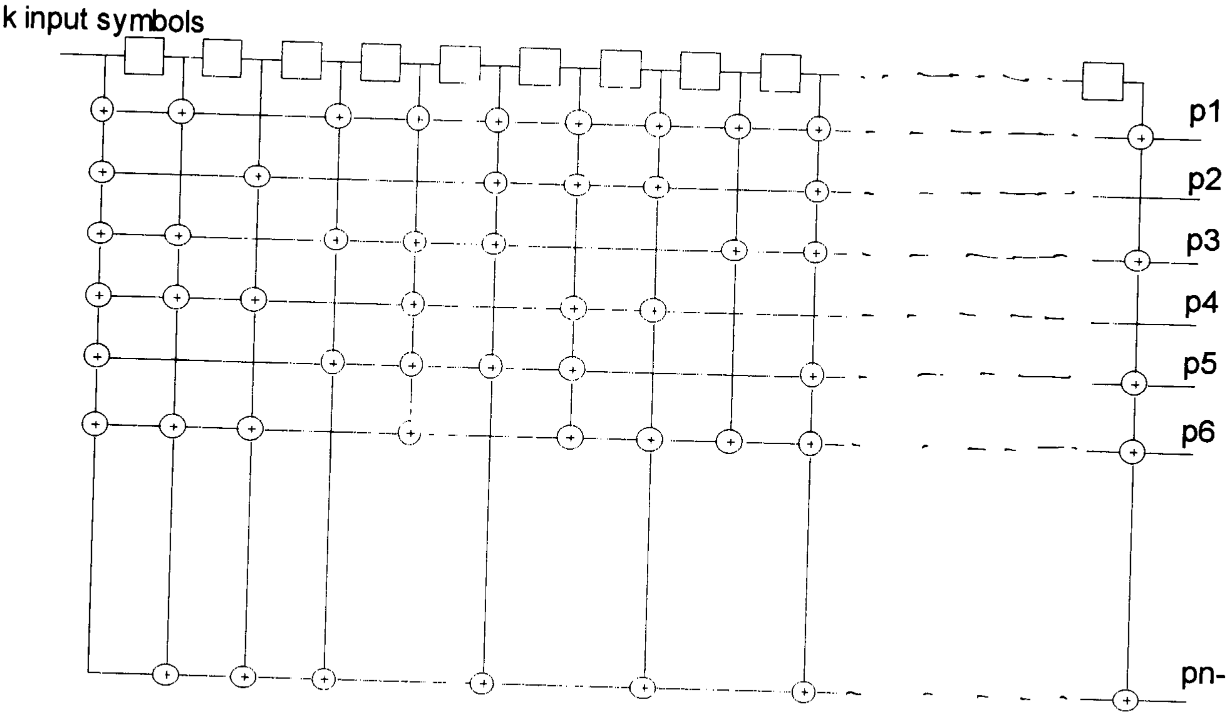

Fig 1 Encoding of n-k parity symbols from input k symbols

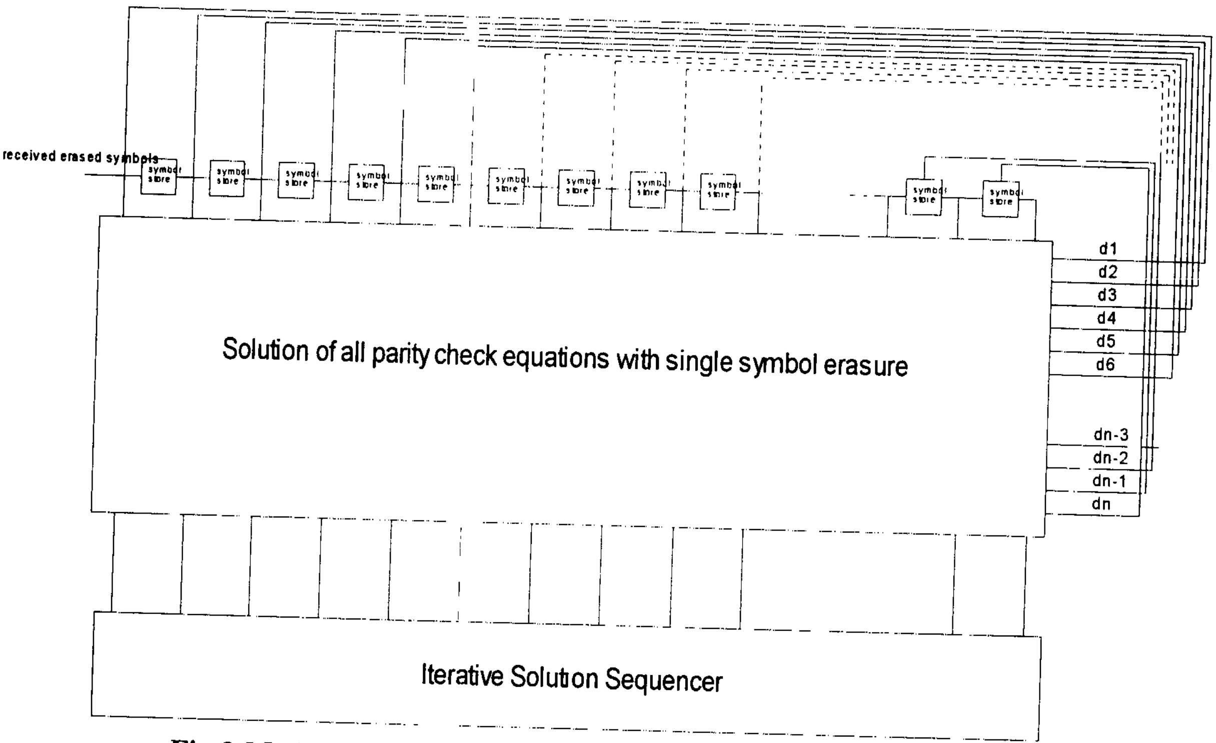

Fig 2 Method A: Correction of Erased Symbols Using Sequential Solution of Single Erased Symbol Parity Check Equations

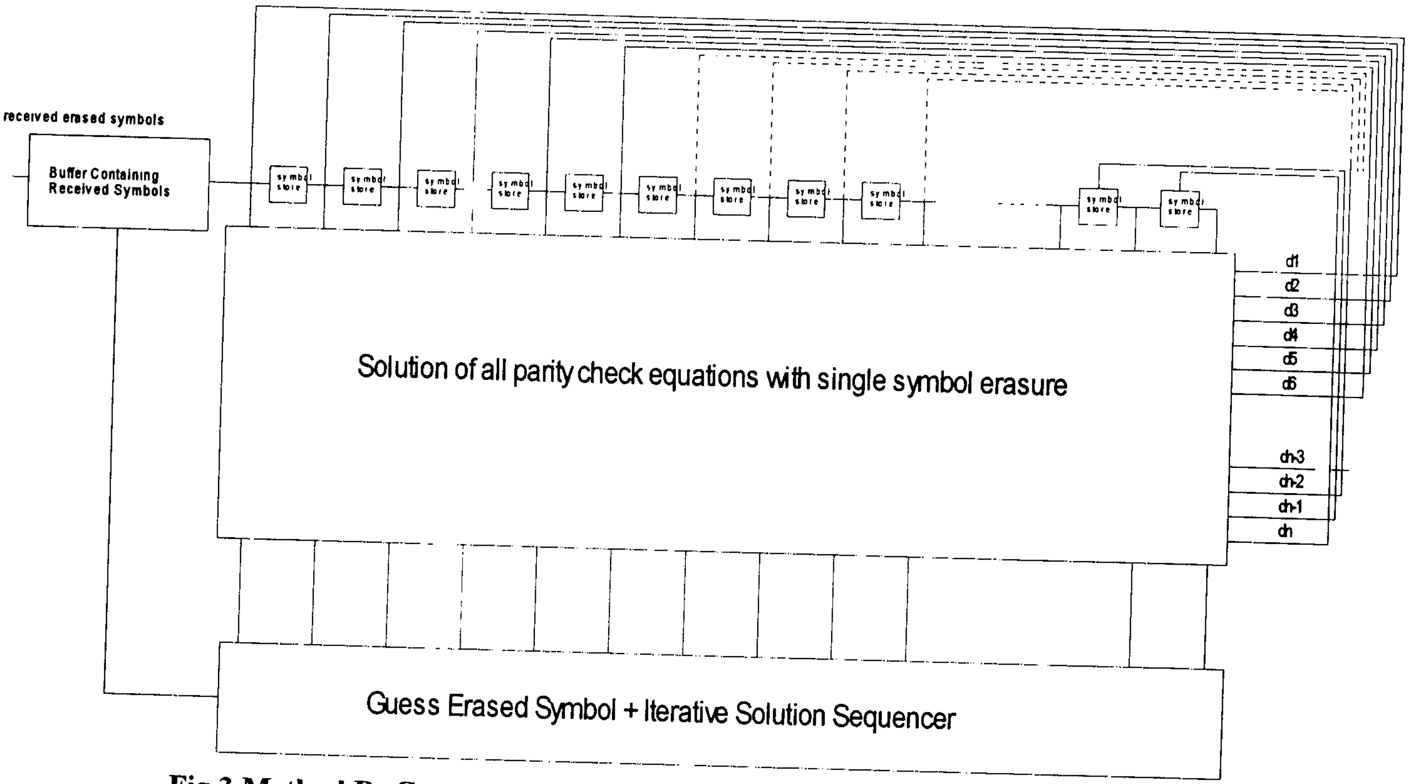

Fig 3 Method B: Correction of Erased Symbols Using Guess Erased Symbols + Sequential Solution of Remaining Single Erased Symbol Parity Check Equations

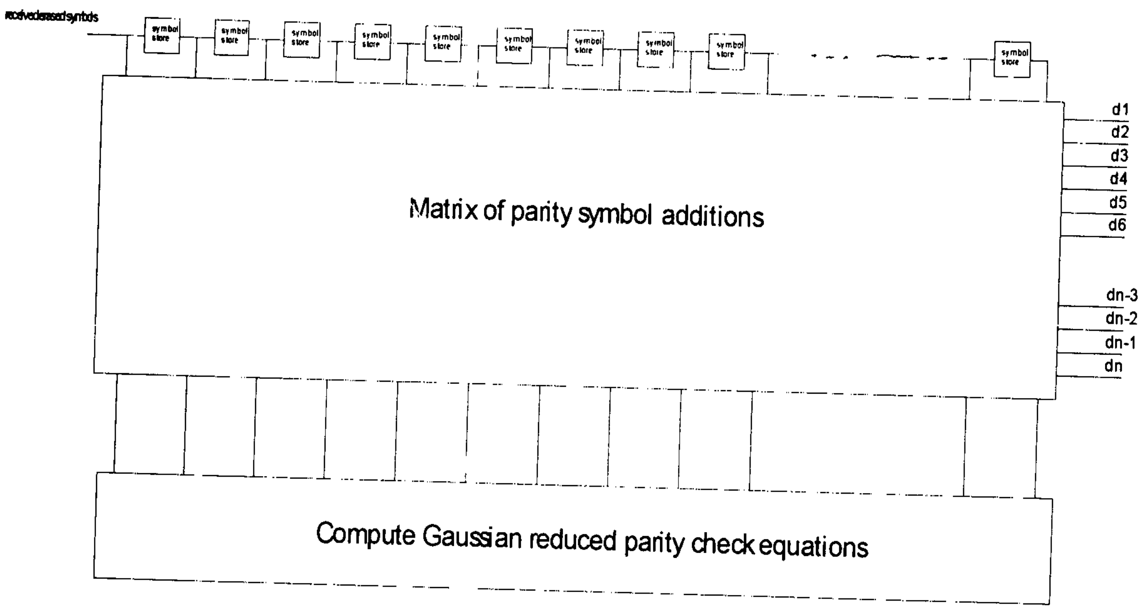

Fig 4 Method C: Erasure Correction using Gaussian Reduction and Solution of Parity Check Equations

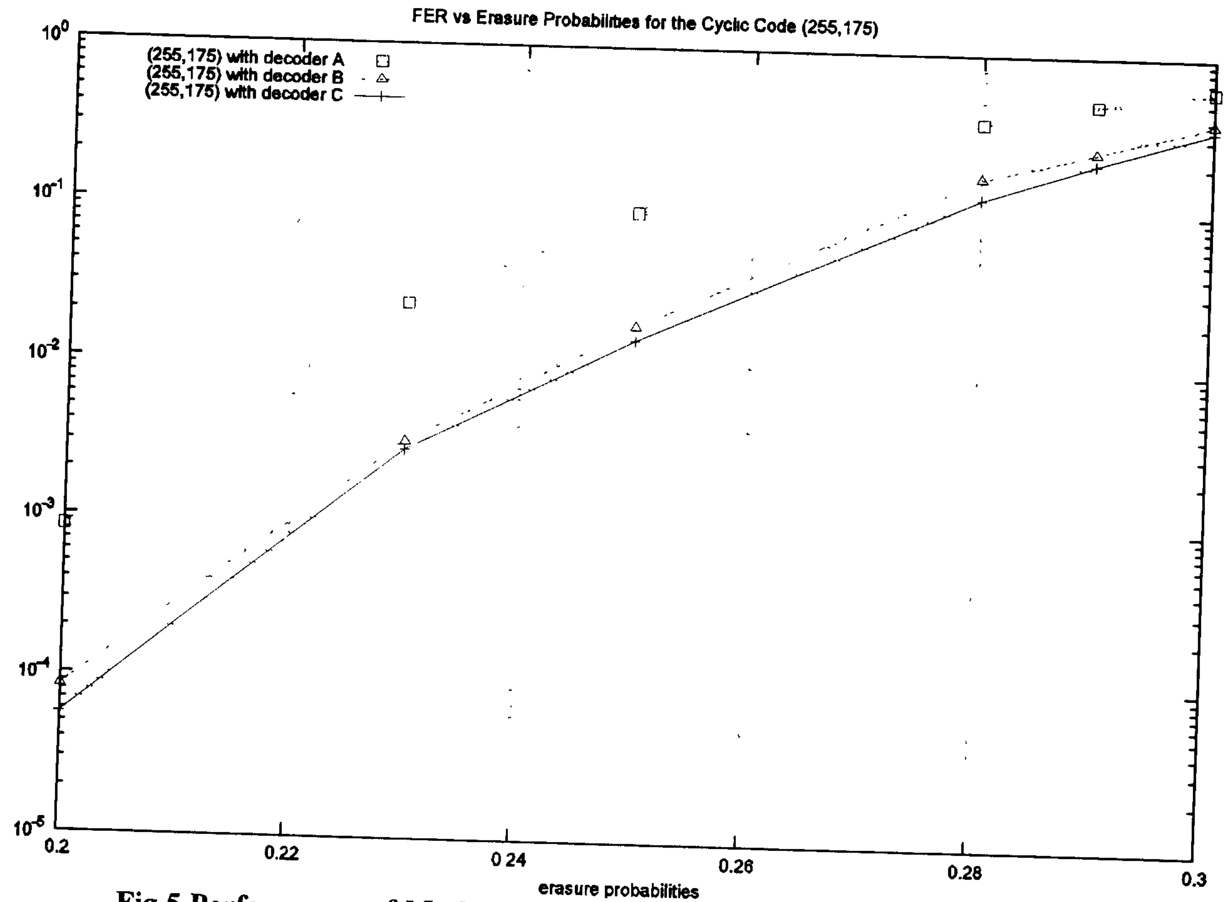

Fig 5 Performance of Methods A,B and C as a function of code and erasure bit probability

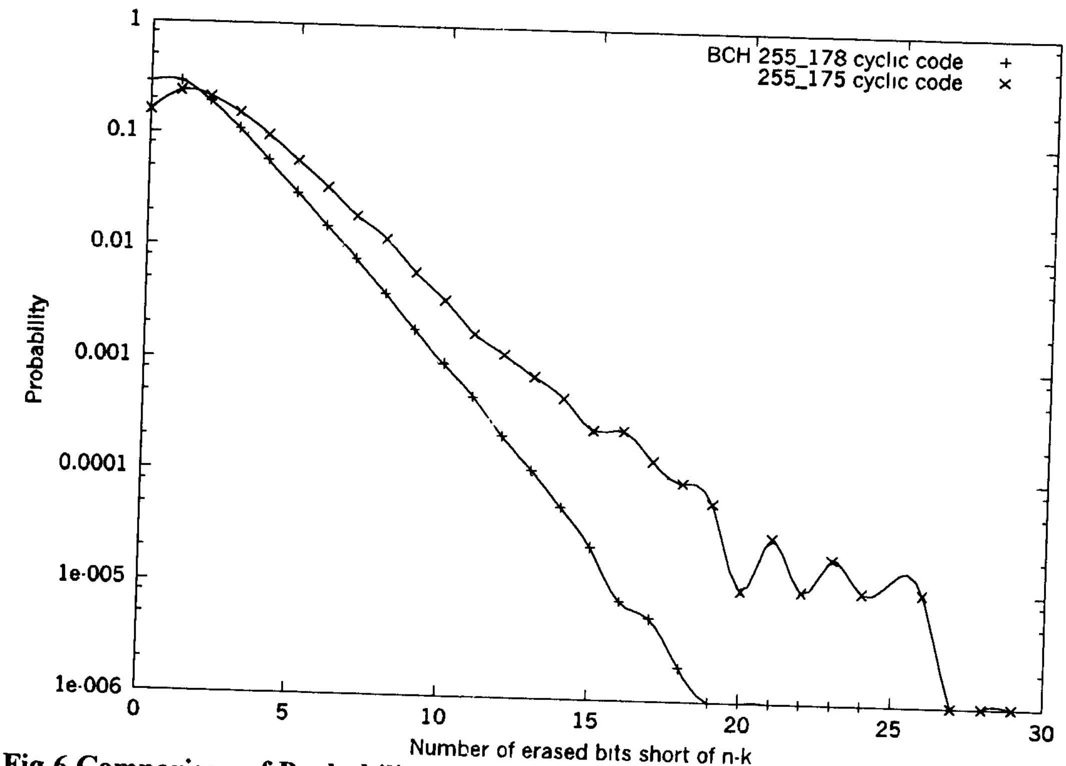

Fig 6 Comparison of Probability Distribution of Number of Erased Bits not Corrected from Maximum Correctible (n-k) for (255,175) code and BCH (255,178) code

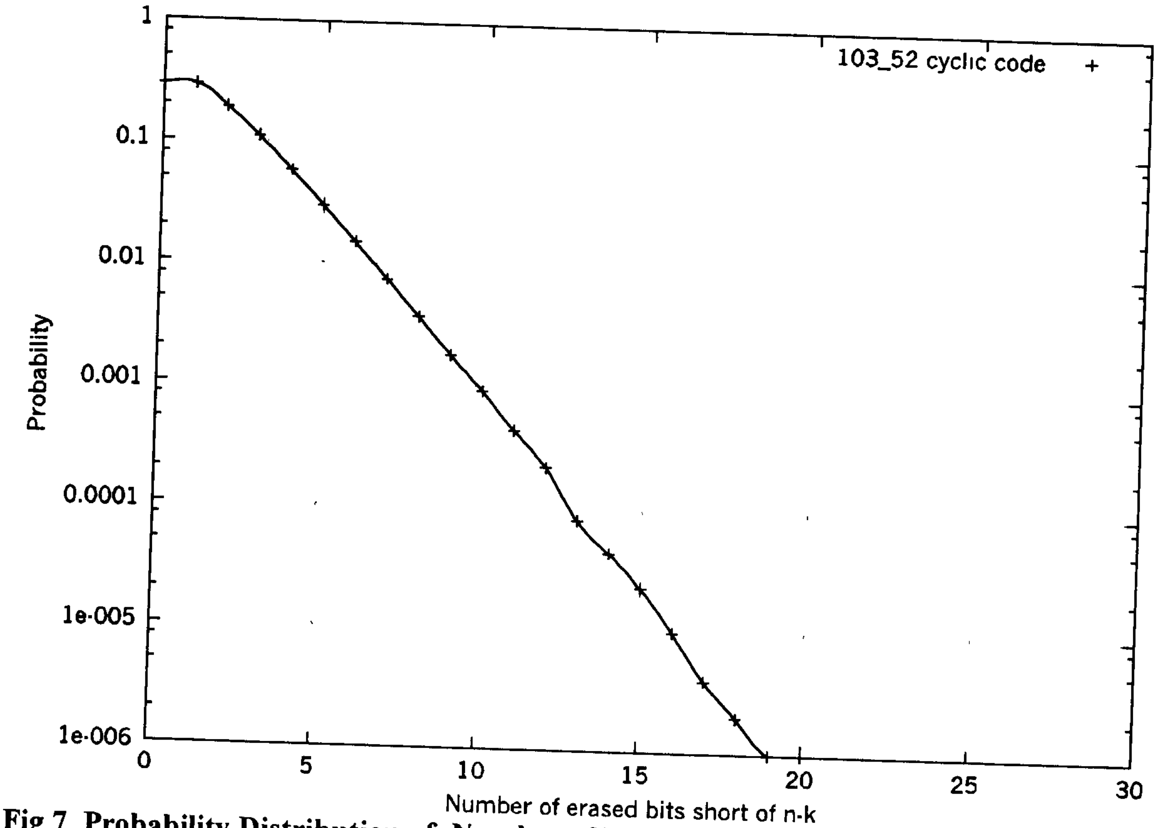

Fig 7 Probability Distribution of Number of Erased Bits not Corrected from Maximum Correctible (n-k) for (103,52) quadratic residue code

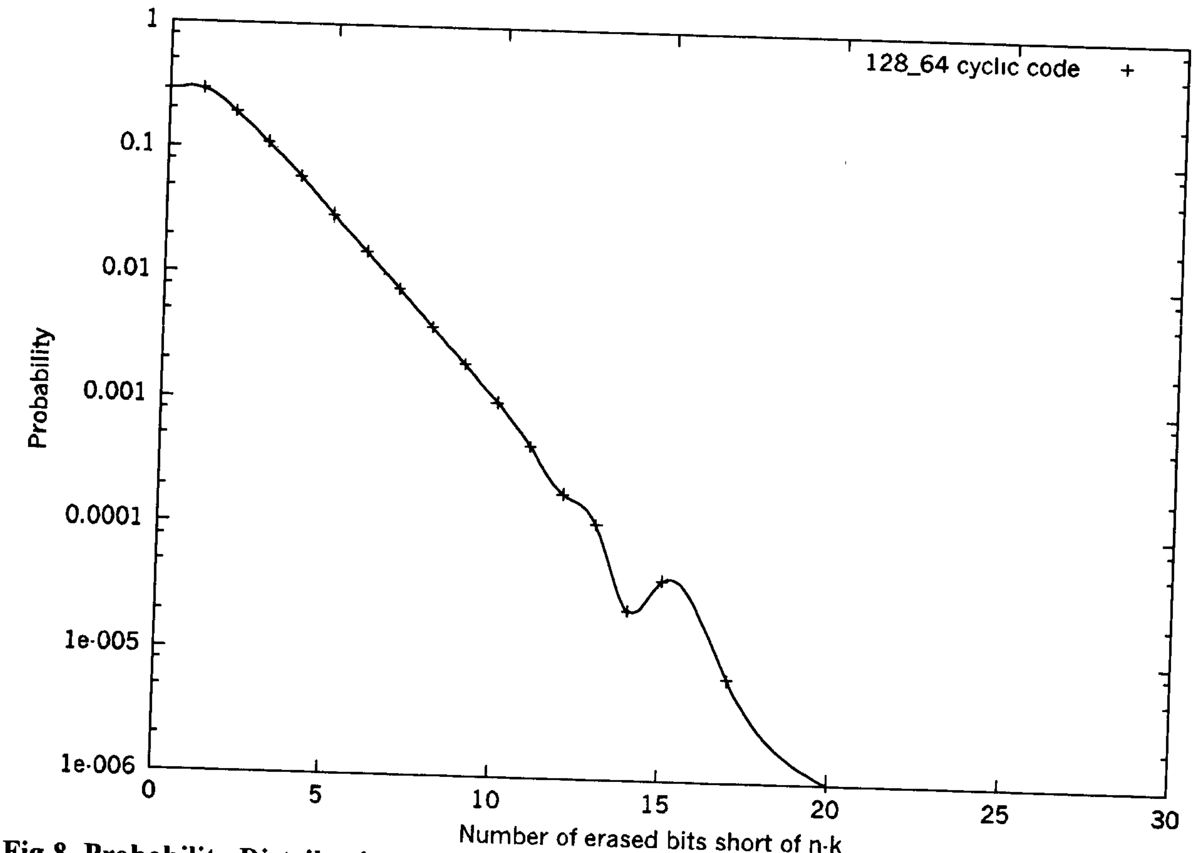

Fig 8 Probability Distribution of Number of Erased Bits not Corrected from Maximum Correctible (n-k) for (128,64) extended BCH code

A SYSTEM FOR THE CORRECTION OF MISSING OR DELETED SYMBOLS

This invention relates to the encoding and decoding of symbols in symbol sequences in order to correct or compensate for deleted or missing symbols All of the symbol information is retrieved from each symbol sequence despite some of the symbols being missing or deleted.

BACKGROUND

In many communication systems, storage systems and broadcasting systems information is encoded as a sequence of symbols from a finite alphabet and binary symbols are the most common. This invention is applicable to any finite alphabet and may be used for binary symbols but is not limited to these In several applications the symbol stream is subject to missing symbols or deleted symbols. Without loss of generality the total number of symbols is denoted by n and the number of information symbols contained in the symbol stream is denoted by k The number of erased symbols is denoted by m. In communication systems these m symbols are referred to as erased symbols [1] In broadcasting systems or in multicasting systems, symbols are transmitted serially. Each symbol could be a data packet of a suitable length in bits In broadcasting systems or multicasting, at any point in time, only r symbols may have been received either because of network congestion or interference or because only r symbols have been transmitted up until now The system described below enables the k information symbols to be retrieved with a given probability of symbol error, provided r is greater than or equal to k In information storage systems, such as hard drives, the invention described below may be used to recover the stored information before all of the n symbols of data have been read (so as to achieve faster access) or to compensate for unread /missing symbols It is shown that the system can cope with a much higher number of missing symbols than the current state of the art, for the same transmitted sequence For example if 64 information bits are encoded into 128 bits by using a code, such as the extended (128,64) BCH code given in the Appendix, then the traditional hard decision decoder [2] will be able to correct only up to 21 missing bits, one bit less than the d_{min} of the code which is 22. With the invention described below, up to 62 missing bits on average can be corrected and 45 missing bits can be corrected with a probability of 0.99999.

DESCRIPTION OF THE INVENTION

The information to be stored or transmitted consists of k information symbols or packets, which are encoded into n code symbols by using a series of parity check equations. This is a well known procedure described generally, for example, in references [1] and [2]. The n code symbols are transmitted or stored A schematic diagram for the encoder is shown in Fig 1.The invention provides a means of recovering the k information symbols or packets from k+r symbols, where r is a small integer between 0 and some limit, r_{lim}.The reasons for wanting to recover the k information symbols or packets from k+r symbols are manifold. Some of the remaining symbols may have been corrupted in storage, and are missing, or in a communication system, congestion, time constraints, distortion, noise or other transmission impairments may have prevented the reception of the remaining symbols or packets.

The code that is used for encoding is not the subject of this invention. The open literature is a rich source of suitable codes using binary or non binary symbols and having various values of k and n. See for example [1] and [2]

The invention uses the received symbols and the parity check equations in three methods, Method A, Method B, and Method C, each with increasing levels of complexity of the decoder so that in the implementation of the invention, performance may be traded off against cost

Method A

In method A the decoder inserts the received symbols into the parity check equations substituting symbols representing unknown variables for the missing symbols The equations are scanned for the number of unknowns in each equation. Those equations with only one unknown symbol, are solved for the unknown symbol and the solved symbols are substituted back into the parity check equations. The procedure then repeats and continues repeating until all missing symbols have been solved and found. A schematic diagram for the decoder is shown in Fig 2 Without loss of generality the invention is described further by way of example using binary symbols and with a short code length of 15.

Consider the parity check matrix for a (15,7) error correcting code with a dmin of 5 The code is guaranteed to correct 4 erasures using syndrome decoding [2] By using the decoding Method A, the average number of erasures corrected is approximately 5 5.The code_length is 15 and there are 7 information bits encoded into a total of 15 bits The sequence to be transmitted or stored is represented as

$$C(x) = c_0 + c_1x^{-1} + c_2x^{-2} + c_3x^{-3} + c_4x^{-4} + c_5x^{-5} + c_6x^{-6} + c_7x^{-7} + c_8x^{-8} + c_9x^{-9} + c_{10}x^{-10} + c_{11}x^{-11} + c_{12}x^{-12} + c_{13}x^{-13} + c_{14}x^{-14}$$

The information is contained in the 15 binary coefficients $c_0, c_1, c_2, c_3, c_4 \ldots c_{14}$ following the encoding according to the parity check equations

There are 8 parity check equations which define 8 of the coefficients c_i ·

$c_0 + c_1 + c_3 + c_7 = 0$ (1)

$c_1 + c_2 + c_4 + c_8 = 0$ (2)

$c_2 + c_3 + c_5 + c_9 = 0$ (3)
$c_3 + c_4 + c_6 + c_{10} = 0$ (4)
$c_4 + c_5 + c_7 + c_{11} = 0$ (5)
$c_5 + c_6 + c_8 + c_{12} = 0$ (6)
$c_6 + c_7 + c_9 + c_{13} = 0$ (7)
$c_7 + c_8 + c_{10} + c_{14} = 0$ (8)

The 7 information bits may be distributed amongst the 15 coefficients in several ways, the usual convention is that the coefficients c_0 through to c_6 are set equal to the information bits and c_7 through to c_{14} are parity check bits derived from c_0 through to c_6.

As an alternative representation, the parity check equations may be represented as a parity check matrix:

110100010000000
011010001000000
001101000100000
000110100010000
000011010001000
000001101000100
000000110100010
000000011010001

where in each row of the matrix the position of the 1's indicate the coefficients used in the parity check equation corresponding to that row

Consider a sequence of missing bits or erasures in positions 3, 12, 1, 9, 2, 0
The received or the available sequence is therefore represented as

$$c_4x^4 + c_5x^5 + c_6x^6 + c_7x^7 + c_8x^8 + c_{10}x^{10} + c_{11}x^{11} + c_{13}x^{13} + c_{14}x^{14}$$

The 6 coefficients c_0, c_1, c_2, c_3, c_9, and c_{12} are unknown and need to be determined unambiguously This is impossible with the conventional syndrome decoder as only 4 coefficients can be determined as the code has a d_{min} of 5.

For each erasure, the erasure is represented as an unknown in each of the parity check equations that it appears as z_i, where i is the position of the erasure

The set of parity check equations become
$z_0 + z_1 + z_3 + c_7 = 0$ (A1)
$z_1 + z_2 + c_4 + c_8 = 0$ (A2)
$z_2 + z_3 + c_5 + z_9 = 0$ (A3)
$z_3 + c_4 + c_6 + c_{10} = 0$ (A4)
$c_4 + c_5 + c_7 + c_{11} = 0$ (A5)
$c_5 + c_6 + c_8 + z_{12} = 0$ (A6)
$c_6 + c_7 + z_9 + c_{13} = 0$ (A7)
$c_7 + c_8 + c_{10} + c_{14} = 0$ (A8)

All the equations are scanned to determine the number of unknowns z_i in each equation Those equations with only one unknown are solved. These are equations (A4), (A6) and (A7) and z_3, z_9 and z_{12} are solved to produce c_3, c_9 and c_{12}. That is
$c_3 = c_4 + c_6 + c_{10}$
$c_9 = c_6 + c_7 + c_{13}$ and
$c_{12} = c_5 + c_6 + c_8$

These are then substituted into equations (A1) through to (A8) to produce the following equations
$z_0 + z_1 + c_3 + c_7 = 0$ (B1)
$z_1 + z_2 + c_4 + c_8 = 0$ (B2)
$z_2 + c_3 + c_5 + c_9 = 0$ (B3)
$c_3 + c_4 + c_6 + c_{10} = 0$ (B4)
$c_4 + c_5 + c_7 + c_{11} = 0$ (B5)
$c_5 + c_6 + c_8 + c_{12} = 0$ (B6)
$c_6 + c_7 + c_9 + c_{13} = 0$ (B7)
$c_7 + c_8 + c_{10} + c_{14} = 0$ (B8)

The procedure then repeats, scanning the equations to determine the number of unknowns z_i in each equation. Those equations with only one unknown are solved. In this case it is equation (B3) for the unknown z_2 in order to find c_2. The solution (s) is substituted to produce a new set of equations

$z_0 + z_1 + c_3 + c_7 = 0$ (C1)
$z_1 + c_2 + c_4 + c_8 = 0$ (C2)
$c_2 + c_3 + c_5 + c_9 = 0$ (C3)
$c_3 + c_4 + c_6 + c_{10} = 0$ (C4)
$c_4 + c_5 + c_7 + c_{11} = 0$ (C5)
$c_5 + c_6 + c_8 + c_{12} = 0$ (C6)
$c_6 + c_7 + c_9 + c_{13} = 0$ (C7)
$c_7 + c_8 + c_{10} + c_{14} = 0$ (C8)

The procedure then repeats, scanning the equations to determine the number of unknowns z_i in each equation. Those equations with only one unknown are solved which is now equation (C2) for the unknown z_1 in order to find c_1. The solution (s) is substituted in the equations to produce a new set of equations of which there is only one unknown z_0 and the new equation (D0) solved to find c_0. In this way all of the erasures have been corrected As shown in Fig 2 the received codeword is clocked into the n stage, tri-state, shift register, with each stage i storing one of 3 states, either 0 or 1 or z_i to represent an erasure in that position. Each shift register stage feeds the set of parity check equations and those with only one z entry are solved and the solutions fed back to the respective shift register stage as shown in Fig 2. The procedure then repeats until the shift register contains no z states or until decoding fails due to an excessive number of erasures being present in the parity check equations.

Method B

In the event that each parity check equation contains two or more erasures then the procedure of Method A will fail. Method B is the same as Method A except that in the event of all parity check equations containing two or more erasures, one or more erased bits are selected and their states are systematically guessed The basis of the selection, is that these erased bits are in the maximum number of equations containing only two erasures The states of these selected bits are set to all possible symbol states, one at a time, substituted back into the parity check equations, and Method A invoked A schematic diagram of the decoder is shown in Fig 3. If the procedure progresses with all the parity check equations solved then decoding is declared complete. In the event that all of the parity check equations cannot be solved, then the received symbols are input again from the receive buffer, and new guesses are made for the selected bits If all possible guesses have been made for the selected bits, then a new selection of bits to be guessed is made and the procedure repeated.

Method C

For the ultimate performance all of the information contained in the parity check equations needs to be used with a consequent increase in decoder complexity The codeword is received and unknowns z_i substituted in positions of erased symbols in the parity check equations Starting with one of the erased symbols, z_s, the first equation containing this symbol is flagged that it will be used for the solution of z_s and then this equation is subtracted from all other equations containing z_s and not yet flagged, to produce a new set of equations The procedure repeats with the next of the non flagged equations containing the next erased symbol z_{s+1} flagged and subtracted from all of the remaining non flagged equations containing z_{s+1}. The procedure is a form of Gaussian reduction of the parity check equations

The procedure repeats until either no non flagged equations remain containing the erased symbol z_{s+fail} (in which case a decoder failure is declared) or no erased symbols remain that are not in flagged equations In this case starting with the last flagged equation with erased symbol z_{last} this equation is solved to find c_{last} and this equation is unflagged This coefficient is substituted back into the remaining flagged equations containing z_{last} The procedure now repeats with the second from last flagged equation now being solved for z_{last-1}; this equation is unflagged and followed by back substitution of c_{last-1} for z_{last-1} in the remaining flagged equations. A block schematic of the decoder is shown in Fig 4 The received symbols are stored in the shift register with the erased symbols being replaced by the unknowns z_i The Gaussian reduced equations are computed and used to define the connection of symbol adders from each respective shift register stage to compute the outputs d_1 through to d_n The non erased symbols contained in the shift register are switched directly through to their respective outputs so that overall, the decoded codeword containing no erased symbols is present at the outputs d_1 through to d_n

As an example of the method consider the (15,7) code with erasures in positions c_1, c_2, c_3, c_4

After substitution with the unknowns in the parity check equations, the following set of equations are obtained:

$c_0 + z_1 + z_3 + c_7 = 0$ (D1)
$z_1 + z_2 + z_4 + c_8 = 0$ (D2)
$z_2 + z_3 + c_5 + c_9 = 0$ (D3)
$z_3 + z_4 + c_6 + c_{10} = 0$ (D4)
$z_4 + c_5 + c_7 + c_{11} = 0$ (D5)
$c_5 + c_6 + c_8 + c_{12} = 0$ (D6)
$c_6 + c_7 + c_9 + c_{13} = 0$ (D7)
$c_7 + c_8 + c_{10} + c_{14} = 0$ (D8)

(This simple example could have the erasures corrected by Method A , but Method C will be applied as an example of the procedure)

Starting with z_1 , equation (D1) is flagged and subtracted from equation (D2) only because z_1 is not contained in the other equations. The new set of equations obtained is as follows

$c_0 + z_1 + z_3 + c_7$	$= 0$	(E1)	* for z_1
$c_0 + z_3 + c_7 + z_2 + z_4 + c_8$	$= 0$	(E2)	
$z_2 + z_3 + c_5 + c_9$	$= 0$	(E3)	
$z_3 + z_4 + c_6 + c_{10}$	$= 0$	(E4)	
$z_4 + c_5 + c_7 + c_{11}$	$= 0$	(E5)	
$c_5 + c_6 + c_8 + c_{12}$	$= 0$	(E6)	
$c_6 + c_7 + c_9 + c_{13}$	$= 0$	(E7)	
$c_7 + c_8 + c_{10} + c_{14}$	$= 0$	(E8)	

The * represents the flagging of equation (E1) meaning that this equation will be fixed and used to solve for z_1

The next unknown is z_2 contained in unflagged equation (E2) This equation is flagged and subtracted from the non flagged equations containing z_2 to produce the next set of equations.

$c_0 + z_1 + z_3 + c_7$	$= 0$	(F1)	* for z_1
$c_0 + z_3 + c_7 + z_2 + z_4 + c_8$	$= 0$	(F2)	* for z_2
$c_0 + c_7 + z_4 + c_8 + c_5 + c_9$	$= 0$	(F3)	
$z_3 + z_4 + c_6 + c_{10}$	$= 0$	(F4)	
$z_4 + c_5 + c_7 + c_{11}$	$= 0$	(F5)	
$c_5 + c_6 + c_8 + c_{12}$	$= 0$	(F6)	
$c_6 + c_7 + c_9 + c_{13}$	$= 0$	(F7)	
$c_7 + c_8 + c_{10} + c_{14}$	$= 0$	(F8)	

The next unknown is z_4 contained in equation (F3) This equation is flagged and subtracted from the non flagged equations containing z_4 to produce the next set of equations

$c_0 + z_1 + z_3 + c_7$	$= 0$	(G1)	*for z_1
$c_0 + z_3 + c_7 + z_2 + z_4 + c_8$	$= 0$	(G2)	* for z_2
$c_0 + c_7 + z_4 + c_8 + c_5 + c_9$	$= 0$	(G3)	* for z_4
$c_0 + c_7 + c_8 + c_5 + c_9 + z_3 + c_6 + c_{10}$	$= 0$	(G4)	
$c_0 + c_8 + c_9 + c_{11}$	$= 0$	(G5)	
$c_5 + c_6 + c_8 + c_{12}$	$= 0$	(G6)	
$c_6 + c_7 + c_9 + c_{13}$	$= 0$	(G7)	
$c_7 + c_8 + c_{10} + c_{14}$	$= 0$	(G8)	

There is now only one unknown remaining in an unflagged equation , which is z_3 in(G4) This is solved first to find c_3 .which is substituted into all flagged equations that z_3 appears, ie (G1) and (G2). Equation (G3) is solved next for z_4 to determine c_4 which is then substituted into (G2). Equation (G2) is solved next for z_2 and finally (G1) is solved for z_1

Multicast and Broadcast

In multicast and broadcast applications, information is transmitted in data packets with typical packet lengths from 30 bits to 1000 bits. These packets could define a symbol from a Galois field [1], viz GF(2^m) but with m equal to 30 or more up to and beyond 1000 bits this is impracticable and it is more convenient to use a matrix approach with the packets forming the rows of the matrix. The columns of bits (or symbols) are encoded using an error correcting code Usually, but not essentially, the same error correcting code would be used to encode each column of symbols The matrix of symbols may be defined as ·

$b_{00}\, b_{01}\, b_{02}\, b_{03}\, b_{04}\, b_{05}\, b_{06}\, b_{07}$	b_{0s}	= packet 1
$b_{10}\, b_{11}\, b_{12}\, b_{13}\, b_{14}\, b_{15}\, b_{16}\, b_{17}$	b_{1s}	= packet 2
$b_{20}\, b_{21}\, b_{22}\, b_{23}\, b_{24}\, b_{25}\, b_{26}\, b_{27}$	b_{2s}	= packet 3
... ...		
... .		
... .		
$b_{n-10}\, b_{n-11}\, b_{n-12}\, b_{n-13}\, b_{n-14}\, b_{n-15}\, b_{n-16}\, b_{n-17}$	b_{n-1s}	= packet n

There are a total of (s+1) k information symbols which are encoded using the parity check equations of a selected code into a total number of transmitted symbols equal to (s+1) n The symbols are transmitted in a series of packets with each packet corresponding to a row of the matrix as indicated above For example the row:

$b_{20}\,b_{21}\,b_{22}\,b_{23}\,b_{24}\,b_{25}\,b_{26}\,b_{27}$ b_{2s}

is transmitted as a single packet.

Self contained codewords are encoded from each column of k symbols For example $b_{00}\,b_{10}\,b_{20}\,b_{30}\,b_{40}$ b_{k-10} form the k information symbols of one codeword and the remaining symbols $b_{k0}\,b_{k+10}\,b_{k+20}$ b_{n-10} are the n-k parity symbols of that codeword and these are these are the result of encoding the k information symbols

As a result of network congestion, drop outs, loss of radio links or other multifarious reasons, not all of the transmitted packets are received. The effect is that some rows above may be considered as erased rows. The decoding procedure is that codewords are assembled from the received packets with missing symbols corresponding to the missing packets marked as z_{ij} corresponding to their position in the matrix. For example if the second packet only is missing above

The first received codeword corresponds to the first column above and is

$b_{00}\,z_{10}\,b_{20}\,b_{30}\,b_{40}\,b_{50}\,b_{60}\,b_{70}$ b_{n-10}

The second codeword corresponding to the second column above is

$b_{01}\,z_{11}\,b_{21}\,b_{31}\,b_{41}\,b_{51}\,b_{61}\,b_{71}$ $b_{n\,11}$

and so on

All of the three methods outlined above may be used to solve for the erased symbol z_{10} in the first received codeword, and for the erased symbol z_{11} in the second received codeword and so on up to the s'th codeword (column) solving for symbol z_{1s-1}

As an example the binary, extended (128,64) BCH code given in the Appendix could be used to encode the information data. The packet length is chosen to be 100 bits, and the total transmission could consist of 128 transmitted packets (12,800 bits total) containing 6,400 bits of information On average as soon as any 66 packets from the original 128 packets have been received, the remaining 62 packets are treated as if they are erased The 100 codewords are assembled, and decoded with the results that the erased symbols are solved and the 6,400 bits of information retrieved One additional advantage is that a user does not have to wait until the entire transmission has been received in order to recover the 6,400 bits of information even if there have been no erasures. For this code, on average, only 66 packets have to be received to recover all 6,400 bits of information. (see results below for this code's performance)

Results for Some Typical Codes

The applicability of the decoding methods above depends upon the error correcting code being used and specifically on the parity check matrix being used The parity check matrix should be sparse (each row of the matrix having a small number of non zero entries) for Methods A and B. The sparseness of the parity check matrix does not affect the performance of Method C

A particularly strong binary code and one which has a sparse parity check matrix is the (255,175) binary code given in the Appendix. This code has a length of 255 bits after encoding of 175 information bits

The performance of this code for the three methods above is shown in Fig 5 in terms of the probability of decoder error (FER) as a function of the erasure probability for every transmitted bit An erasure probability of 0.2 means that on average 1 bit in 5 is erased or lost. Method C has the best performance but at the expense of decoder complexity The ultimate performance of this method as a function of error correcting code is shown in Fig 6 for the example (255,175) code which can correct a maximum of 80 erased bits. Fig 6 shows the probability density function of the number of erased bits short of the maximum correctible which is n-k. The results were obtained by computer simulations. The probability of being able to correct only 68 bits, a shortfall of 12 bits, is 1.1×10^{-3} Simulations indicate that on average 77 6 erased bits may be corrected for this code. In comparison the BCH (255,178) code having similar rate is also shown in Fig 6 The BCH code has similar a similar rate but a higher minimum Hamming distance of 22 (compared to 17). It can be seen that it has better performance than the (255,175) code but it has a less sparse parity check matrix and consequently it is less suitable for the decoding Methods A and B Moreover the average shortfall in erasures not being able to becorrected is virtually identical for the two codes

The simulation results of using Method C for the (103,52) quadratic residue binary code [3] are shown in Fig 7. The minimum Hamming distance for this code is 19 and the results are similar to that of the (255,178) BCH code above. It is found from the simulations that on average 49 1 erasure bits are corrected (out of a maximum of 51) and the average shortfall from the maximum is 1.59 bits.

Similarly the results for the extended BCH (128,64) code are shown in Fig 8. This code has a minimum Hamming distance of 22 and has a similar probability density function to the other BCH code above. On average 62 39 erasure bits are corrected (out of a maximum of 64) and the average shortfall is . 61 bits from the maximum

References

[1] W.W.Peterson, Error Correcting Codes, The MIT Press 1961

[2] S.Lin and D.J Costello, Error Control Coding. Fundamentals and Applications, Prentice-Hall 1983

[3] F.J. MacWilliams and N.J A Sloane, The Theory of Error Correcting Codes, North-Holland 1977.

Appendix

The parity check (H) matrix below is for the (255,175) binary code having a minimum Hamming distance of 17. It is a sparse matrix because a code of this length and performance would usually have approximately 80 entries per row instead of 16. Consequently it is particularly suitable for use in Method A and Method B. The -1 symbol at the end of each row is only there to enable the matrix to be easily machine readable, it is not part of the code. The numbers represent the bit positions of the bits involved in each parity check equation.

```
0 18 26 32 56 61 97 106 110 113 125 150 152 172 173 183 -1
1 19 27 33 57 62 98 107 111 114 126 151 153 173 174 184 -1
2 20 28 34 58 63 99 108 112 115 127 152 154 174 175 185 -1
3 21 29 35 59 64 100 109 113 116 128 153 155 175 176 186 -1
4 22 30 36 60 65 101 110 114 117 129 154 156 176 177 187 -1
5 23 31 37 61 66 102 111 115 118 130 155 157 177 178 188 -1
6 24 32 38 62 67 103 112 116 119 131 156 158 178 179 189 -1
7 25 33 39 63 68 104 113 117 120 132 157 159 179 180 190 -1
8 26 34 40 64 69 105 114 118 121 133 158 160 180 181 191 -1
9 27 35 41 65 70 106 115 119 122 134 159 161 181 182 192 -1
10 28 36 42 66 71 107 116 120 123 135 160 162 182 183 193 -1
11 29 37 43 67 72 108 117 121 124 136 161 163 183 184 194 -1
12 30 38 44 68 73 109 118 122 125 137 162 164 184 185 195 -1
13 31 39 45 69 74 110 119 123 126 138 163 165 185 186 196 -1
14 32 40 46 70 75 111 120 124 127 139 164 166 186 187 197 -1
15 33 41 47 71 76 112 121 125 128 140 165 167 187 188 198 -1
16 34 42 48 72 77 113 122 126 129 141 166 168 188 189 199 -1
17 35 43 49 73 78 114 123 127 130 142 167 169 189 190 200 -1
18 36 44 50 74 79 115 124 128 131 143 168 170 190 191 201 -1
19 37 45 51 75 80 116 125 129 132 144 169 171 191 192 202 -1
20 38 46 52 76 81 117 126 130 133 145 170 172 192 193 203 -1
21 39 47 53 77 82 118 127 131 134 146 171 173 193 194 204 -1
22 40 48 54 78 83 119 128 132 135 147 172 174 194 195 205 -1
23 41 49 55 79 84 120 129 133 136 148 173 175 195 196 206 -1
24 42 50 56 80 85 121 130 134 137 149 174 176 196 197 207 -1
25 43 51 57 81 86 122 131 135 138 150 175 177 197 198 208 -1
26 44 52 58 82 87 123 132 136 139 151 176 178 198 199 209 -1
27 45 53 59 83 88 124 133 137 140 152 177 179 199 200 210 -1
28 46 54 60 84 89 125 134 138 141 153 178 180 200 201 211 -1
29 47 55 61 85 90 126 135 139 142 154 179 181 201 202 212 -1
30 48 56 62 86 91 127 136 140 143 155 180 182 202 203 213 -1
31 49 57 63 87 92 128 137 141 144 156 181 183 203 204 214 -1
32 50 58 64 88 93 129 138 142 145 157 182 184 204 205 215 -1
33 51 59 65 89 94 130 139 143 146 158 183 185 205 206 216 -1
34 52 60 66 90 95 131 140 144 147 159 184 186 206 207 217 -1
35 53 61 67 91 96 132 141 145 148 160 185 187 207 208 218 -1
36 54 62 68 92 97 133 142 146 149 161 186 188 208 209 219 -1
37 55 63 69 93 98 134 143 147 150 162 187 189 209 210 220 -1
38 56 64 70 94 99 135 144 148 151 163 188 190 210 211 221 -1
39 57 65 71 95 100 136 145 149 152 164 189 191 211 212 222 -1
40 58 66 72 96 101 137 146 150 153 165 190 192 212 213 223 -1
41 59 67 73 97 102 138 147 151 154 166 191 193 213 214 224 -1
42 60 68 74 98 103 139 148 152 155 167 192 194 214 215 225 -1
43 61 69 75 99 104 140 149 153 156 168 193 195 215 216 226 -1
44 62 70 76 100 105 141 150 154 157 169 194 196 216 217 227 -1
45 63 71 77 101 106 142 151 155 158 170 195 197 217 218 228 -1
46 64 72 78 102 107 143 152 156 159 171 196 198 218 219 229 -1
47 65 73 79 103 108 144 153 157 160 172 197 199 219 220 230 -1
48 66 74 80 104 109 145 154 158 161 173 198 200 220 221 231 -1
49 67 75 81 105 110 146 155 159 162 174 199 201 221 222 232 -1
50 68 76 82 106 111 147 156 160 163 175 200 202 222 223 233 -1
51 69 77 83 107 112 148 157 161 164 176 201 203 223 224 234 -1
52 70 78 84 108 113 149 158 162 165 177 202 204 224 225 235 -1
53 71 79 85 109 114 150 159 163 166 178 203 205 225 226 236 -1
54 72 80 86 110 115 151 160 164 167 179 204 206 226 227 237 -1
55 73 81 87 111 116 152 161 165 168 180 205 207 227 228 238 -1
56 74 82 88 112 117 153 162 166 169 181 206 208 228 229 239 -1
57 75 83 89 113 118 154 163 167 170 182 207 209 229 230 240 -1
58 76 84 90 114 119 155 164 168 171 183 208 210 230 231 241 -1
59 77 85 91 115 120 156 165 169 172 184 209 211 231 232 242 -1
60 78 86 92 116 121 157 166 170 173 185 210 212 232 233 243 -1
61 79 87 93 117 122 158 167 171 174 186 211 213 233 234 244 -1
62 80 88 94 118 123 159 168 172 175 187 212 214 234 235 245 -1
63 81 89 95 119 124 160 169 173 176 188 213 215 235 236 246 -1
64 82 90 96 120 125 161 170 174 177 189 214 216 236 237 247 -1
65 83 91 97 121 126 162 171 175 178 190 215 217 237 238 248 -1
66 84 92 98 122 127 163 172 176 179 191 216 218 238 239 249 -1
```

67 85 93 99 123 128 164 173 177 180 192 217 219 239 240 250 -1
68 86 94 100 124 129 165 174 178 181 193 218 220 240 241 251 -1
69 87 95 101 125 130 166 175 179 182 194 219 221 241 242 252 -1
70 88 96 102 126 131 167 176 180 183 195 220 222 242 243 253 -1
71 89 97 103 127 132 168 177 181 184 196 221 223 243 244 254 -1
0 72 90 98 104 128 133 169 178 182 185 197 222 224 244 245 -1
1 73 91 99 105 129 134 170 179 183 186 198 223 225 245 246 -1
2 74 92 100 106 130 135 171 180 184 187 199 224 226 246 247 -1
3 75 93 101 107 131 136 172 181 185 188 200 225 227 247 248 -1
4 76 94 102 108 132 137 173 182 186 189 201 226 228 248 249 -1
5 77 95 103 109 133 138 174 183 187 190 202 227 229 249 250 -1
6 78 96 104 110 134 139 175 184 188 191 203 228 230 250 251 -1
7 79 97 105 111 135 140 176 185 189 192 204 229 231 251 252 -1

The extended BCH (128,64) code[2], is represented by its parity check (H) matrix describing the 64 parity check equations. This matrix is:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 -1
1 3 5 6 7 9 10 14 15 17 18 19 21 25 26 28 30 31 35 36 37 38 39 40 41 44 45 53 54 58 59 61 63 65 -1
2 4 6 7 8 10 11 15 16 18 19 20 22 26 27 29 31 32 36 37 38 39 40 41 42 45 46 54 55 59 60 62 64 66 -1
3 5 7 8 9 11 12 16 17 19 20 21 23 27 28 30 32 33 37 38 39 40 41 42 43 46 47 55 56 60 61 63 65 67 -1
4 6 8 9 10 12 13 17 18 20 21 22 24 28 29 31 33 34 38 39 40 41 42 43 44 47 48 56 57 61 62 64 66 68 -1
5 7 9 10 11 13 14 18 19 21 22 23 25 29 30 32 34 35 39 40 41 42 43 44 45 48 49 57 58 62 63 65 67 69 -1
6 8 10 11 12 14 15 19 20 22 23 24 26 30 31 33 35 36 40 41 42 43 44 45 46 49 50 58 59 63 64 66 68 70 -1
7 9 11 12 13 15 16 20 21 23 24 25 27 31 32 34 36 37 41 42 43 44 45 46 47 50 51 59 60 64 65 67 69 71 -1
8 10 12 13 14 16 17 21 22 24 25 26 28 32 33 35 37 38 42 43 44 45 46 47 48 51 52 60 61 65 66 68 70 72 -1
9 11 13 14 15 17 18 22 23 25 26 27 29 33 34 36 38 39 43 44 45 46 47 48 49 52 53 61 62 66 67 69 71 73 -1
10 12 14 15 16 18 19 23 24 26 27 28 30 34 35 37 39 40 44 45 46 47 48 49 50 53 54 62 63 67 68 70 72 74 -1
11 13 15 16 17 19 20 24 25 27 28 29 31 35 36 38 40 41 45 46 47 48 49 50 51 54 55 63 64 68 69 71 73 75 -1
12 14 16 17 18 20 21 25 26 28 29 30 32 36 37 39 41 42 46 47 48 49 50 51 52 55 56 64 65 69 70 72 74 76 -1
13 15 17 18 19 21 22 26 27 29 30 31 33 37 38 40 42 43 47 48 49 50 51 52 53 56 57 65 66 70 71 73 75 77 -1
14 16 18 19 20 22 23 27 28 30 31 32 34 38 39 41 43 44 48 49 50 51 52 53 54 57 58 66 67 71 72 74 76 78 -1
15 17 19 20 21 23 24 28 29 31 32 33 35 39 40 42 44 45 49 50 51 52 53 54 55 58 59 67 68 72 73 75 77 79 -1
16 18 20 21 22 24 25 29 30 32 33 34 36 40 41 43 45 46 50 51 52 53 54 55 56 59 60 68 69 73 74 76 78 80 -1
17 19 21 22 23 25 26 30 31 33 34 35 37 41 42 44 46 47 51 52 53 54 55 56 57 60 61 69 70 74 75 77 79 81 -1
18 20 22 23 24 26 27 31 32 34 35 36 38 42 43 45 47 48 52 53 54 55 56 57 58 61 62 70 71 75 76 78 80 82 -1
19 21 23 24 25 27 28 32 33 35 36 37 39 43 44 46 48 49 53 54 55 56 57 58 59 62 63 71 72 76 77 79 81 83 -1
20 22 24 25 26 28 29 33 34 36 37 38 40 44 45 47 49 50 54 55 56 57 58 59 60 63 64 72 73 77 78 80 82 84 -1
21 23 25 26 27 29 30 34 35 37 38 39 41 45 46 48 50 51 55 56 57 58 59 60 61 64 65 73 74 78 79 81 83 85 -1
22 24 26 27 28 30 31 35 36 38 39 40 42 46 47 49 51 52 56 57 58 59 60 61 62 65 66 74 75 79 80 82 84 86 -1
23 25 27 28 29 31 32 36 37 39 40 41 43 47 48 50 52 53 57 58 59 60 61 62 63 66 67 75 76 80 81 83 85 87 -1
24 26 28 29 30 32 33 37 38 40 41 42 44 48 49 51 53 54 58 59 60 61 62 63 64 67 68 76 77 81 82 84 86 88 -1
25 27 29 30 31 33 34 38 39 41 42 43 45 49 50 52 54 55 59 60 61 62 63 64 65 68 69 77 78 82 83 85 87 89 -1
26 28 30 31 32 34 35 39 40 42 43 44 46 50 51 53 55 56 60 61 62 63 64 65 66 69 70 78 79 83 84 86 88 90 -1
27 29 31 32 33 35 36 40 41 43 44 45 47 51 52 54 56 57 61 62 63 64 65 66 67 70 71 79 80 84 85 87 89 91 -1
28 30 32 33 34 36 37 41 42 44 45 46 48 52 53 55 57 58 62 63 64 65 66 67 68 71 72 80 81 85 86 88 90 92 -1
29 31 33 34 35 37 38 42 43 45 46 47 49 53 54 56 58 59 63 64 65 66 67 68 69 72 73 81 82 86 87 89 91 93 -1
30 32 34 35 36 38 39 43 44 46 47 48 50 54 55 57 59 60 64 65 66 67 68 69 70 73 74 82 83 87 88 90 92 94 -1
31 33 35 36 37 39 40 44 45 47 48 49 51 55 56 58 60 61 65 66 67 68 69 70 71 74 75 83 84 88 89 91 93 95 -1
32 34 36 37 38 40 41 45 46 48 49 50 52 56 57 59 61 62 66 67 68 69 70 71 72 75 76 84 85 89 90 92 94 96 -1
33 35 37 38 39 41 42 46 47 49 50 51 53 57 58 60 62 63 67 68 69 70 71 72 73 76 77 85 86 90 91 93 95 97 -1
34 36 38 39 40 42 43 47 48 50 51 52 54 58 59 61 63 64 68 69 70 71 72 73 74 77 78 86 87 91 92 94 96 98 -1
35 37 39 40 41 43 44 48 49 51 52 53 55 59 60 62 64 65 69 70 71 72 73 74 75 78 79 87 88 92 93 95 97 99 -1
36 38 40 41 42 44 45 49 50 52 53 54 56 60 61 63 65 66 70 71 72 73 74 75 76 79 80 88 89 93 94 96 98 100 -1
37 39 41 42 43 45 46 50 51 53 54 55 57 61 62 64 66 67 71 72 73 74 75 76 77 80 81 89 90 94 95 97 99 101 -1
38 40 42 43 44 46 47 51 52 54 55 56 58 62 63 65 67 68 72 73 74 75 76 77 78 81 82 90 91 95 96 98 100 102 -1
39 41 43 44 45 47 48 52 53 55 56 57 59 63 64 66 68 69 73 74 75 76 77 78 79 82 83 91 92 96 97 99 101 103 -1
40 42 44 45 46 48 49 53 54 56 57 58 60 64 65 67 69 70 74 75 76 77 78 79 80 83 84 92 93 97 98 100 102 104 -1
41 43 45 46 47 49 50 54 55 57 58 59 61 65 66 68 70 71 75 76 77 78 79 80 81 84 85 93 94 98 99 101 103 105 -1
42 44 46 47 48 50 51 55 56 58 59 60 62 66 67 69 71 72 76 77 78 79 80 81 82 85 86 94 95 99 100 102 104 106 -1
43 45 47 48 49 51 52 56 57 59 60 61 63 67 68 70 72 73 77 78 79 80 81 82 83 86 87 95 96 100 101 103 105 107 -1
44 46 48 49 50 52 53 57 58 60 61 62 64 68 69 71 73 74 78 79 80 81 82 83 84 87 88 96 97 101 102 104 106 108 -1
45 47 49 50 51 53 54 58 59 61 62 63 65 69 70 72 74 75 79 80 81 82 83 84 85 88 89 97 98 102 103 105 107 109 -1
46 48 50 51 52 54 55 59 60 62 63 64 66 70 71 73 75 76 80 81 82 83 84 85 86 89 90 98 99 103 104 106 108 110 -1
47 49 51 52 53 55 56 60 61 63 64 65 67 71 72 74 76 77 81 82 83 84 85 86 87 90 91 99 100 104 105 107 109 111 -1
48 50 52 53 54 56 57 61 62 64 65 66 68 72 73 75 77 78 82 83 84 85 86 87 88 91 92 100 101 105 106 108 110 112 -1
49 51 53 54 55 57 58 62 63 65 66 67 69 73 74 76 78 79 83 84 85 86 87 88 89 92 93 101 102 106 107 109 111 113 -1
50 52 54 55 56 58 59 63 64 66 67 68 70 74 75 77 79 80 84 85 86 87 88 89 90 93 94 102 103 107 108 110 112 114 -1
51 53 55 56 57 59 60 64 65 67 68 69 71 75 76 78 80 81 85 86 87 88 89 90 91 94 95 103 104 108 109 111 113 115 -1
52 54 56 57 58 60 61 65 66 68 69 70 72 76 77 79 81 82 86 87 88 89 90 91 92 95 96 104 105 109 110 112 114 116 -1
53 55 57 58 59 61 62 66 67 69 70 71 73 77 78 80 82 83 87 88 89 90 91 92 93 96 97 105 106 110 111 113 115 117 -1
54 56 58 59 60 62 63 67 68 70 71 72 74 78 79 81 83 84 88 89 90 91 92 93 94 97 98 106 107 111 112 114 116 118 -1
55 57 59 60 61 63 64 68 69 71 72 73 75 79 80 82 84 85 89 90 91 92 93 94 95 98 99 107 108 112 113 115 117 119 -1

56 58 60 61 62 64 65 69 70 72 73 74 76 80 81 83 85 86 90 91 92 93 94 95 96 99 100 108 109 113 114 116 118 120 -1
57 59 61 62 63 65 66 70 71 73 74 75 77 81 82 84 86 87 91 92 93 94 95 96 97 100 101 109 110 114 115 117 119 121 -1
58 60 62 63 64 66 67 71 72 74 75 76 78 82 83 85 87 88 92 93 94 95 96 97 98 101 102 110 111 115 116 118 120 122 -1
59 61 63 64 65 67 68 72 73 75 76 77 79 83 84 86 88 89 93 94 95 96 97 98 99 102 103 111 112 116 117 119 121 123 -1
60 62 64 65 66 68 69 73 74 76 77 78 80 84 85 87 89 90 94 95 96 97 98 99 100 103 104 112 113 117 118 120 122 124 -1
61 63 65 66 67 69 70 74 75 77 78 79 81 85 86 88 90 91 95 96 97 98 99 100 101 104 105 113 114 118 119 121 123 125 -1
62 64 66 67 68 70 71 75 76 78 79 80 82 86 87 89 91 92 96 97 98 99 100 101 102 105 106 114 115 119 120 122 124 126 -1
0 63 65 67 68 69 71 72 76 77 79 80 81 83 87 88 90 92 93 97 98 99 100 101 102 103 106 107 115 116 120 121 123 125 -1

As before the notation is that each row contains the positions of bits in that equation. There are 64 rows because there are 64 equations. The k information bits (also 64) may be in any position but traditionally these are in positions 0 to 63.

CLAIMS

Claim 1. A system in which k information symbols are encoded into n symbols using parity check equations from an error correcting code and some of the n symbols are marked as erased symbols The k information symbols are retrieved from the remaining symbols based on a decoder (Method A above) which examines the number of erasures in each parity check equation and solves for the erased symbols in those parity check equations that only contain one erased symbol All of these erased symbols are determined and substituted back into the parity check equations and the procedure is repeated over and over again until all erased symbols have been determined and all k information symbols retrieved

Claim 2. A system according to Claim 1and in which the n symbols containing marked erasure symbols are stored in a buffer memory. Under the condition that all of the parity check equations contain two or more erased symbols then one or more of these symbols are guessed as to their respective states using all combinations of their respective states, one state at a time, and the guesses substituted into the parity check equations as described in Method B. Each parity check equation which has only one erased symbol is solved for that symbol and all the solved symbols substituted into the equations and the procedure repeated as in Claim 1. In the event that not all equations are solved the original n symbols are retrieved from the buffer memory and the whole procedure repeated with new guesses for one or more of the erased symbols until all parity check equations are solved or a decoder failure is declared.

Claim 3. A system in which k information symbols are encoded into n symbols using parity check equations from an error correcting code and in which some of the n symbols are marked as erased symbols The k information symbols are retrieved based on a decoder (Method C above) which selects one erased symbol at a time The parity check equations are examined and the first equation containing this symbol is flagged that it will be used to find this symbol Each unflagged equation containing this symbol is replaced with the result of that unflagged equation minus the equation just flagged The procedure is repeated examining all unflagged equations for the presence of the next selected erased symbol The first unflagged equation found is flagged and subtracted from all other unflagged equations containing that symbol The procedure is repeated over and over again until each erased symbols has a corresponding flagged equation and either there are no erased symbols left that have not been selected or there are no unflagged equations containing the currently selected erased symbol. In this latter event a decoder failure is declared. The last flagged equation is used to solve for its respectively selected erased symbol and the solved symbol is substituted into all equations in which it is present. The next to last flagged equation is solved for its selected erased symbol and then the solved symbol is substituted into all equations in which it is present The procedure is repeated over and over again working through the flagged equations, in last to be flagged order until all erased symbols have been solved

Claim 4. A system in which the product of k and s (k s) information symbols are encoded into n s symbols using parity check equations from an error correcting code and transmitted or stored as packets of length s symbols. The encoding is carried out so that each packet contains a single coordinate symbol from each of s encoded codewords With k or more packets received or recovered the remaining packets are marked as being erased The symbols within these packets are marked as erased symbols and the corresponding s codewords each decoded using one of the three Methods A, B or C If successful decoding is not possible either an additional non erased packet is obtained and the decoding procedure attempted again or a decoding failure is declared.

Claim 5. A system of multicasting or broadcasting in which the information to be transmitted or stored is partitioned into blocks of k packets of fixed length or of variable length equal to s symbols and encoded according to Claim 4 into n packets of length s symbols. As soon as k or more packets have been received or recovered the k s information symbols corresponding to that partition are decoded using one of the Methods A, B or C In this way a system is provided in which information may be multicast or broadcast in minimum time and also be resilient to lost packets r

CLAIMS **Amendments to the claims have been filed as follows**

Claim 1. A system in which k information symbols are encoded into n symbols using parity check equations from an error correcting code and in which some of the n symbols are marked as erased symbols with k information symbols retrieved based on a decoder which selects one erased symbol at a time and flags the first parity check equation containing this symbol and replaces each unflagged equation containing this symbol with that unflagged equation minus the equation just flagged continuing the procedure until each erased symbol has a corresponding flagged equation and then the last flagged equation is used to solve for its respectively selected erased symbol and the solved symbol is substituted into all equations in which it is present followed by the next to last flagged equation which is solved for its selected erased symbol with this solved symbol substituted into all equations in which it is present followed by working through the remaining flagged equations, in last to be flagged order, until all erased symbols have been solved.

Claim 2. A system according to Claim 1, in which the product of k and s+1 ,(k.s+k) information symbols are encoded into n.s+n symbols using parity check equations from an error correcting code and transmitted or stored as packets of length s+1symbols with the encoding carried out so that each packet contains a single coordinate symbol from each of s+1 encoded codewords and once k or more packets have been received or recovered the remaining packets are marked as being erased with the symbols contained in these packets marked as erased symbols and the corresponding s+1 codewords each decoded such that the ks +k information symbols are retrieved.

Claim 3 A system according to Claim 2 , for application in multicasting, or broadcasting , in which the information to be transmitted or stored is partitioned into blocks of k packets of fixed length equal to s+1 symbols and encoded into n packets of length s+1 symbols and in which as soon as k or more packets have been received or recovered the k.s +k information symbols are decoded such that the ks+k information symbols are retrieved

The Patent Office

PATENTS DESIGNS COPYRIGHT TRADE MARKS

11

Application No: GB0428042.6
Claims searched: 1, 4 & 5

Examiner: Dr Mark Shawcross
Date of search: 8 February 2006

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	1 & 4-5	US 5115436 A (McAULEY) in particular, see col.3 line 31 to col.4 line 68 and figures 1 & 6
X	1 & 4-5	EP 0903955 A1 (STMicroelectronics) in particular, see paragraphs [0026-0042].
X	1 & 4-5	US 4555784 A (WOOD) col.6 line 28 to col.12 line 41

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^{X} :

G4A

Worldwide search of patent documents classified in the following areas of the IPC

G06F

The following online and other databases have been used in the preparation of this search report

Online: EPODOC, WPI, INSPEC