Three and Four-dimensional Parity-check Codes for Correction and Detection of Multiple Errors

Naveen Babu Anne, Utthaman Thirunavukkarasu, Dr. Shahram Latifi
Department of Electrical and Computer Engineering
University of Nevada-Las Vegas
Las Vegas, Nevada-89119
{ naveenba, utthaman, latifi}@egr.unlv.edu

Abstract

We examine two different schemes of three dimensional parity checking codes that can be obtained by arranging the information and parity bits as a combination of rows and columns or by arranging them as two dimensional planes to obtain a three dimensional cube. Finding the number of errors detected and the numbers of errors corrected has been the main aim of this work. The code rate and the over head of each scheme has been calculated and compared with that of the standard parity schemes available. Some general equations have been derived to represent these families of codes. A four dimensional scheme that can detect and correct more multiple errors has also been discussed.

1. Introduction

A message is represented in coded form as a sequence of binary digits. The coded message may be uniform or non uniform in length. For convenience, in this paper we have arranged all the binary digits in equi-dimensional arrays.

The ability to detect errors in a message requires parity checking schemes, so that erroneous word will be recognizable as nonstandard or different from the expected result. When a codeword is received, the receiver always receives all the bits, but some of them may be bad .A bad bit does not disappear, nor does it change into something other than a bit. A bad bit simply changes its value either from 0 to 1 or from 1 to 0. This is what we call as an error in the message and various parity schemes are used to detect and correct the error. The parity schemes are not completely reliable but can be made more and more powerful by adding more number of parity check bits and by selecting valid code words carefully.

Greater capability of error detection and correction can be obtained by increased redundancy in the coding. This can be achieved by two ways: 1. more than one bit can be used as parity check bit with correspondingly greater detection power; and, 2.the group length may be decreased .In this case, the probability of errors is decreased and the effectiveness of the parity check codes is increased.

This paper addresses itself to the first procedure, namely, by increasing the number of parity check bits and explores the possibility of arranging them in various dimensions. This paper is organized as follows: Section 2 describes three-dimensional (horizontal, vertical and one main diagonal) parity checking codes and compares its performance with currently existent two-dimensional and diagonal codes. Section 3 describes a three-dimensional parity checking scheme in the shape of a cube which comes with greater error detection and correction capability. In section 4, we propose a new four dimensional parity scheme with five error detection and two error correction capabilities. Conclusions and remarks follow in section5.

2. Three-dimensional parity checking matrix (horizontal, vertical and diagonal parity)

A parity bit can be added to a group of information bits to complete the total number of '1' bits to an odd number. Even parity can also be used and the only difference between odd and even parity schemes is that, in the case of even parity, a group of all zeros is valid, whereas, with odd parity, any group of bits with a parity bit added cannot be all zeros.

Parity bits can be used to design simple and efficient error correcting and detecting codes. By arranging the message bits as a rectangle and adding a parity check bit at the end of each row and column, we obtain a two dimensional parity checking matrix. A similar, but slightly more efficient code is a triangular configuration, where the information bits are arranged in a triangle, with the parity bits at the diagonal .Each parity bit is the parity of all the bits in its row and column.

2.1 Diagonal from left bottom to right top

In this paper, we used both the above schemes and devised a new parity checking matrix with the ability to check the parity horizontally, vertically and along the main diagonal. We have worked out the scheme for both odd and even parities. Figure 1 shows the three dimensional parity checking matrix with the bottom row acting as horizontal parity row, and the rightmost column acting as vertical parity column and the diagonal running from left bottom corner to right top corner as the diagonal parity.

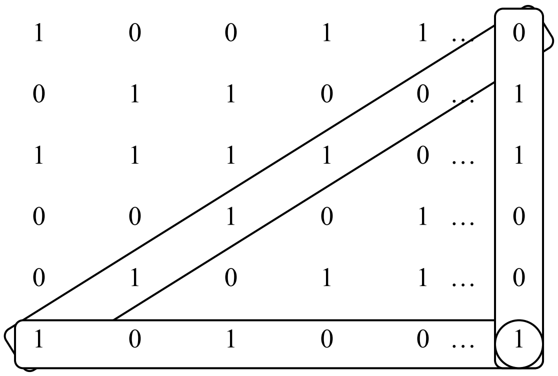

Fig.1. Horizontal, vertical and diagonal parity scheme.

Here, we assume that the length and height of the rectangle are equal. We also consider the bit common to the diagonal and the either the horizontal or vertical parity line as only horizontal or vertical parity bit. It is not considered in the diagonal parity line. And the bit common to horizontal and vertical parity lines checks the errors occurring in the parity line itself and so , we call this bit as the 'System parity bit'.

Let the length be 'h' and height be 'v' and the length of the diagonal be'd'. We assume that

$$h=v=d.$$

Then we have

Number of parity bits =

$$(d-1) + (d-1) + (d-2) = 3d-4 \quad (1)$$

Number of information bits =

$$[d*(d-1) + (d-2)(d-3)]/2 \quad (2)$$

Total bits (information bits+parity bits) =

$$[d*(d-1) + (d-2)(d-3)]/2 + 3d-4 \quad (3)$$

$$\text{Overhead} = \frac{\text{No. of parity bits}}{\text{No. of information bits}} = \frac{3d-4}{d^2-3d+3} \quad (4)$$

$$\text{Code Rate} = \frac{\text{No. of information bits}}{\text{No. of information bits+parity bits}} = \frac{d^2-3d+3}{d^2-1} \quad (5)$$

This parity checking scheme has the ability to correct a single error and detect up to three errors. If an error occurs in four bits in the pattern of a square, such errors go undetected by this scheme.

The horizontal row of parity bits checks the parity for corresponding columns and the vertical row of parity bits checks the parity for the corresponding rows for errors. Each bit in the diagonal parity line checks the parity for the respective row and the column. The diagonal checks the errors in both the upper as well as the lower halves. The system parity bit shows an error if there is an error in either the horizontal parity or vertical parity bits.

This scheme is illustrated with the following examples. Figure 2 shows a three dimensional odd parity checking matrix.

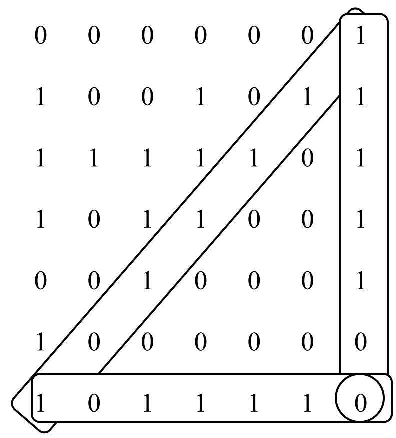

Fig.2. Horizontal, vertical and diagonal odd parity scheme.

This scheme checks odd parity and the above derived general equations for overhead and code rate can be verified for this example. Here, we can observe that we can correct up to a single error and detect up to a maximum of three errors.

2.2 Diagonal from left top to right bottom

Another three dimensional scheme similar to the former has been devised with the direction of the diagonal being from left top corner to right bottom corner. The difference between these two schemes is that the number of parity bits in the former model is 1 less than the present model. Thus the present model has a higher for the present scheme whereas the code rate is lesser than the former scheme.

Figure3 shows the three dimensional parity checking matrix with the last row as the horizontal parity, last column as the vertical parity and the diagonal running from the left top element to the right bottom element. The bit common to horizontal and vertical parity rows is termed as 'System parity bit'. One more fact to note here

0-7695-2108-8/04 $ 20.00 © 2004 IEEE

is that there are no bits common to the diagonal parity and either of the other two dimensions.

0	1	1	0	0…	0
1	0	1	0	1…	1
0	1	0	1	0…	1
1	0	0	0	1…	0
0	0	1	0	1…	1
1	1	1	1	0…	1

Fig.3. Horizontal, vertical and diagonal parity

Let the length be 'h', height be 'v' and length of the diagonal be'd'. We assume that

$$h = v = d$$

Then we have

Number of parity bits = (d-1) + (d-1) + (d-1)

$$= 3(d-1) \tag{6}$$

Number of information bits =

$$\frac{(d-1)*(d-2)}{2} + \frac{(d-1)*(d-2)}{2}$$

$$= (d-1)*(d-2) \tag{7}$$

Total bits (information bits + parity bits) =

$$(d-1)*(d-2) + 3(d-1)$$

$$= d^2 - 1 \tag{8}$$

$$\text{Overhead} = \frac{\text{No. of parity bits}}{\text{No. of information bits}}$$

$$= \frac{3}{(d-2)} \tag{9}$$

$$\text{Code Rate} = \frac{\text{No. of information bits}}{\text{No.of information bits+parity bits}}$$

$$= \frac{(d-2)}{(d+1)} \tag{10}$$

This scheme has the same error correction and detection capabilities as the previously described three dimensional parity schemes. Four bit error patterns in the form of a square cannot be detected by this scheme. The horizontal parity bits check the parity for all the columns in the matrix and the vertical parity bits check the parity for all the rows. Each element in the parity diagonal checks the parity for the message bits to its corresponding row and column and this diagonal works for elements above and below it.

Figure 4 shows an example of the three dimensional odd parity with the third dimension being from left top corner to the right bottom corner. The diagonal parity bit will check the even parity in all its four directions i.e. to the left, to the right, to the top and to the bottom.

1	1	0	1	0	0	1
1	0	0	0	1	0	1
0	0	1	1	1	0	1
0	1	1	1	1	1	1
0	0	1	0	0	1	1
1	0	0	0	1	1	1
1	1	1	1	1	1	1

Fig.4. Three dimensional odd parity scheme.

3. Three dimensional parity checking scheme using parity planes

In this method the information bits are arranged in different layers one over the other to obtain a cube. The parity check bits make up three outer surface of the cube. Therefore we get a three dimensional structure of bits. The most prominent advantage of this scheme is that it can not only detect seven errors but can also correct two errors. any single error in the cube of bits can be easily located as the corresponding parity bits will show error(x,y,z axis).By looking at figure5, we can see that this scheme of error correction easily detects two errors because whenever two bit errors occur one dimension of the parity scheme wouldn't come into the picture as the error will get masked in the dimension which is common to both the errors ,but the other two dimensions will detect the error and the their co-ordinates (x,y,z) can be used to correct the two errors. Errors more than two and less than eight can only be detected.

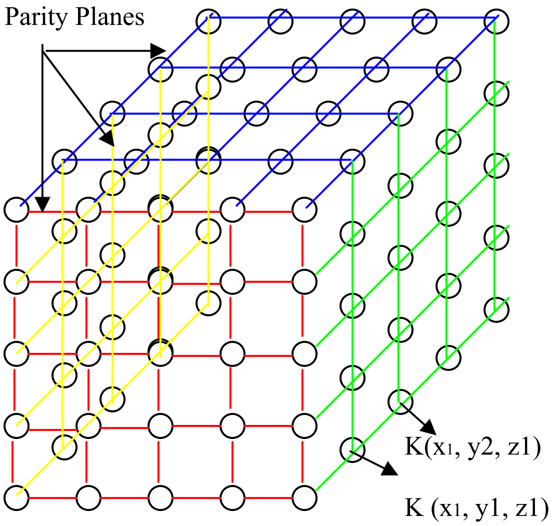

Fig.5. Combinations of parity planes forming a three dimensional cube.

In this parity check scheme, a pattern of eight bit errors will go undetected if the errors are in the shape of a cube i.e. the eight bits form the vertices of a cube.

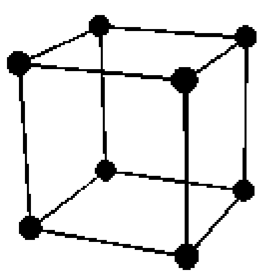

Fig.6. Undetectable 8-bit error pattern.

We can easily notice that all the errors will get masked in the above pattern. There are many other patterns of errors that will go undetected but all these error patterns consist of more than eight bits. For example, an error pattern with twelve bits is shown in figure 7.
The above said scheme corrects one error in the parity bits and can detect up to three errors in the parity bits .The parity bits at the three edges are used to correct and detect any errors in the parity bits.
Consider Figure5 if there are 'l' bits along the length, 'b' bits along the breadth and 'h' bits along the height then the total number of message bits is given by ' l*b*h'
And if we add three planes or parity bits on the three sides of this cube we get the following results:

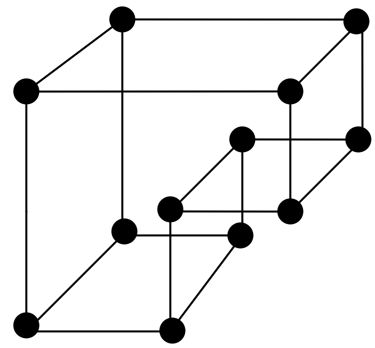

Fig.7. Twelve bit undetectable error pattern

$$k = l*b*h \tag{11}$$

$$N = (l+1)*(b+1)*(h+1) \tag{12}$$

$$N-k = l*h+b*l+h*b+l+b+h+1 \tag{13}$$

Where N is the total number of bits.
k is the number of message bits.
(N-k) is the number parity bits.

$$\text{Code rate} = \frac{l*b*h}{(l+1)\,(b+1)\,(h+1)}$$

$$= \frac{l^3}{(l+1)^3} \quad (l=b=h) \tag{14}$$

$$\text{Overhead} = \frac{l*h+b*l+h*b+l+b+h+1}{l*b*h}$$

$$= \frac{3l*(l+1)+1}{l^3} \quad (l=b=h) \tag{15}$$

Single error can occur in 'l*b*h' ways and the second error can occur in '(l*b*h-1)'ways therefore a double error can occur in (l*b*h)(l*b*h-1) ways ,as this is a double error correcting code, the relation between the total number of bits and the parity bits is given as

$$2^{N-k} \geq (l*b*h)*(l*b*h-1). \tag{16}$$

Assuming that all the bits are equally likely and occur independently, it is possible to write the probability of errors occurring in a sequence of 'N' symbols as

$$P\,(j*N) = {}_N C_{j*}\, P_e^{\,j*} \cdot (1-P_e)^{N-j} \tag{17}$$

Where P_e is the probability that a bit is received in error.

As we have mentioned above the number of errors which go undetected is 8 and above. The expression for the probability of an eight bit error going undetected can be derived as shown:

Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)
0-7695-2108-8/04 $ 20.00 © 2004 IEEE

Total number of ways in which an eight bit error can occur

$$\frac{{}_{N}C_{8}}{2} \tag{18}$$

The total number of ways an eight bit error can occur in the pattern of a cube

$$l*(l-1)*b*(b-1)*h*(h-1) \tag{19}$$

Therefore the probability of an undetected eight bit error is given as

$$P_{un} = \frac{2*\, l*(l-1)*b*(b-1)*h*(h-1)}{{}_{N}C_{8}} \tag{20}$$

In order to see the efficiency of the code we have compared it with the simple overlapping parity check scheme which can correct two errors. For an overlapping parity check error correcting code to correct two errors the minimum requirement is given as

$$2^{N-k} \geq \frac{N + N*(N-1)}{2} \tag{21}$$

$$2^{N-k} \geq \frac{N^2+N+2}{2} \tag{22}$$

This means for a 10 bit information bit you can have only 4 message bits. These results are based on the minimum requirement needed for a overlapping parity check scheme. We find that the overlapping parity scheme requires less number of parity check bits as compare to the three dimensional parity check scheme but the major drawback of the overlapping scheme is that it requires a very complex design procedure and till now no code has been designed for using the overlapping parity check scheme to correct two errors and implementation might also cause a problem.

4. Four Dimensional Parity Checking Scheme

Here, we introduce a new parity checking scheme which has parity bits laid out in four dimensions. This scheme is shown in figure 8.

This scheme has four bit parity bit lines at its four sides forming a four dimensional parity. The bottom row acts as the vertical parity and the rightmost column acts as the horizontal parity while the leftmost column and the top row act as the diagonal parity, checking the parity as shown with the lines in figure 8.

If an error occurs in two bits, the corresponding diagonal parity bits which check the number of 1s in their respective diagonal indicate an error and either the horizontal or vertical parity bits show an error, thereby correcting two errors.

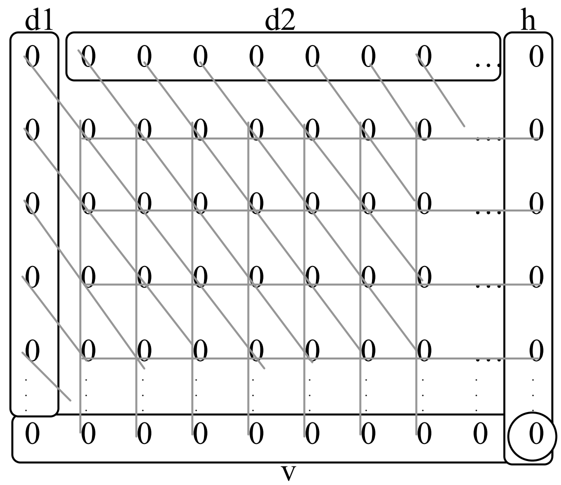

Fig.8. A four dimensional parity check matrix

Two bit errors in any pattern can be corrected with this scheme. This parity scheme has the capability of correcting two errors and detecting up to five errors. A pattern of six error bits go undetected in this parity check scheme. One such pattern is shown in figure 9.

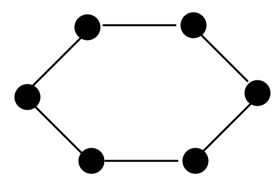

Fig.9. Undetectable 6-bit error pattern

If the pattern of the error takes the form of a hexagon i.e. all the error bits are at the vertices of a hexagon, the errors will go undetected.

Let the number of horizontal information bits = 'h'.

Let the number of vertical information bits = 'v'.

Total number of information bits = h*v

Total number of parity bits = $2*(h+1) + 2*(v+1)$

$$\text{Code rate} = \frac{h*v}{h*v + 2*(h+1) + 2*(v+1)}$$

When h = v,

$$\text{Code rate} = \frac{h^2}{h^2 + 4*(h+1)}$$

$$\text{Overhead} = \frac{2*(h+1) + 2*(v+1)}{h*v}$$

When h = v,

$$\text{Overhead} = \frac{4*(h+1)}{h^2}$$

As compared to the three dimension cube parity scheme, this method is simple and gives a better code rate. As we can see, the general equations are very simple and hence, this scheme can be designed easily.
Figure 10 shows the histograms comparing the code rate of the 2-D and triangle parity schemes with the cube parity, overlapping parity and the 4-D parity schemes.

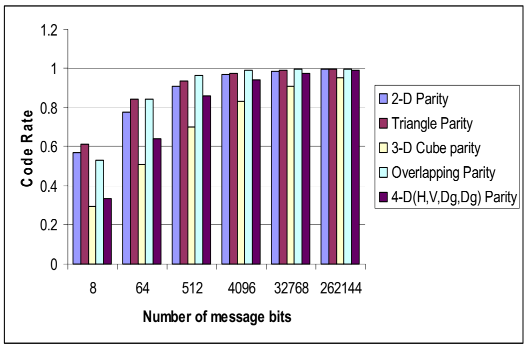

Fig.12. Comparison of code rates of various parity schemes for different number of message bits

5. Conclusion

Two different classes of three dimensional families of codes for detecting and correcting multiple errors have been discussed in detail. The efficiency of a single plane three dimensional parity scheme has been found to be the same as any other two dimensional parity schemes. But, a cube consisting of parity planes at the edges can correct two and detect up to seven errors. A new four dimensional parity scheme has been proposed which can correct two errors and detect up to five errors. Clearly, codes of greater dimensions have greater error correction and detection capabilities, but the complexity of the code and its design increases with the number of dimensions.

6. References

[1]. Morris Rubinoff, "N-Dimensional Codes for Detecting and Correcting Multiple Errors", Communications of the ACM, pp.545-551.
[2].Tan F.Wong and John M.Shea, "Multi-dimensional parity- check codes for bursty channels", Proceedings of the IEEE international Symposium on Information theory (ISIT 2001), pp.123, Washington.D.C. , June 2001.
[3]. David Salomon, Data Compression: The complete Reference, Springer, 2000.
[4]. Kanad Chakraborty and Pinaki Mazumder, Fault-Tolerance and Reliability Techniques for High-Density Random-Access Memories, Prentice Hall PTR, 2002.