Links between N-Modular Redundancy and the Theory of Error-Correcting Codes

V. Bobin*, S. Whitaker** and G. Maki**

* California Design Center
Hewlett-Packard Co.
5301 Stevens Creek Blvd.
Santa Clara, California 95051

** NASA Space Engineering Research Center
University of New Mexico
2650 Yale SE, Suite # 101
Albuquerque, New Mexico 87106

Abstract – **N-Modular Redundancy (NMR) is one of the best known fault tolerance techniques. Replication of a module to achieve fault tolerance is in some ways analogous to the use of a repetition code where an information symbol is replicated as parity symbols in a codeword. Linear Error-Correcting Codes (ECC) use linear combinations of information symbols as parity symbols which are used to generate syndromes for error patterns. These observations indicate links between the theory of ECC and the use of hardware redundancy for fault tolerance. In this paper, we explore some of these links and show examples of NMR systems where identification of good and failed elements is accomplished in a manner similar to error correction using linear ECC's.**

1 Introduction

In a repetition code, the information symbol is replicated as parity symbols for the purpose of error detection and correction. An NMR system with voting also has the same purpose; ie., tolerate failures in some modules. In an NMR system, voting can be replaced by Galois Field (GF) arithmetic operations to identify the good and failed modules from the results of these operations, just as GF arithmetic operations are used to identify and correct the erroneous symbols in a received codeword, provided the number of errors or failed modules is within the correction capability of the code. Some necessary results from the theory of linear ECCs are reviewed in the next section [1, 2].

2 Useful Principles from Coding Theory

This paper attempts to adapt some results from coding theory for use in fault tolerance using modular redundant systems. Some definitions and results from coding theory [1, 2] are reproduced below.

Definition 1 *A* block code *of size* M *over an alphabet with* q *symbols is a set of* M q*-ary sequences of length* n *called* codewords.

If $q = 2$, the symbols are called *bits* and the code is a binary code. Usually $M = q^k$ for some integer k, and the code is an (n, k) code. Each sequence of k q-ary information symbols is associated with a sequence of n q-ary symbols making a codeword.

Definition 2 *The* Hamming distance d(x, y) *between two q-ary sequences x and y of length n is the number of places in which they differ.*

Definition 3 *Let $C = c_i$, $i = 0, \ldots, M-1$ be a code. Then the* minimum distance *of C is the Hamming distance of the pair of codewords with smallest Hamming distance.*

Suppose that a codeword is transmitted, and some symbols are corrupted by the channel. If t errors occur, and if the distance from the received word to every other codeword is larger than t, then the decoder will properly correct the errors, if it presumes that the closest codeword to the received word was actually transmitted. This always occurs if $d \geq (2t+1)$.

Within the space of all q-ary n-tuples, a set of n-tuples is selected and the elements are designated as codewords. If d is the minimum distance of this code and t is the largest integer satisfying $d \geq 2t+1$, then nonintersecting spheres of radius t can be drawn about each of the codewords. A received word that falls in a sphere is decoded as the codeword at the center of that sphere. If t or fewer errors occur, then the received word is always in the proper sphere, and the decoding is correct. Some received words with more than t errors will be within the decoding spheres of other codewords and will be decoded incorrectly. Other received words with more than t errors will lie in the interstitial space between decoding spheres. (Notice that if the minimum distance is $(2t+1)$, a received word with $2t$ or fewer errors can be recognized as being in error, since it will be a non codeword.)

There is a class of codes called *linear codes* which are defined by imposing strong structural property on the codes. This class includes most of the known good codes. Good codes are those which have good error-correction and detection capabilities. The structure helps in the search for good codes as well as in the design of encoders and decoders. There are good codes that are not linear, but non linear codes are hard to deal with because of their lack of structure. The theory of linear codes involves the concept of *vector spaces*. From group theory, it is known that under componentwise vector addition and componentwise scalar multiplication, the set of n-tuples of elements from *GF(q)* (the Galois Field with q elements), is a vector space called $GF(q^n)$. A special case of major importance is $GF(2^n)$, the vector space of all binary n-tuples with two such vectors added by modulo-2 addition in each component.

Definition 4 *A* linear code *is a non-empty set of n-tuples over $GF(q)$ called codewords such that the sum of two codewords is a codeword and the product of any codeword by a field element is a codeword.*

There are q^k codewords, each n symbols long out of which k are information symbols and $(n-k)$ are parity symbols added for error-correction purposes. Another way of defining a linear code is that it is a subspace of $GF(q^n)$.

It follows from the theory of vector spaces that any (n, k) linear code can be represented by its *generator matrix* which is a k by n matrix with the property that any codeword is a linear combination of the rows of this matrix. The generator matrix is a concise way to describe a linear code. The generator matrix is formed by using any set of basis vectors for the subspace as its rows. Any one-to-one pairing of k-tuples and codewords can be used as the encoding procedure, but the most natural approach is to use the equation

$$\mathbf{c} = \mathbf{iG} \tag{1}$$

where, $\mathbf{i}$, the information word, is a k-tuple of information symbols to be encoded and $\mathbf{c}$ is the codeword n-tuple. $\mathbf{G}$ is the generator matrix.

Obviously the parity symbols can be generated as a linear combination of some of the information symbols. Every generator matrix $\mathbf{G}$ is equivalent to one with a k by k identity in the first k columns. That is,

$$\mathbf{G} = [\mathbf{I}\vdots\mathbf{P}] \tag{2}$$

and $\mathbf{P}$ is a k by $(n-k)$ matrix. We call this the *systematic form* of the generator matrix. This generator matrix leads to systematic codes, where the information symbols appear as a block in the first k positions of the codeword, and the parity symbols appear as a block in the next $(n-k)$ positions.

Associated with every generator matrix is a *parity check* matrix $\mathbf{H}$, defined such that

$$\mathbf{G}\mathbf{H}^{\mathbf{T}} = \mathbf{0}. \tag{3}$$

For the systematic form of $\mathbf{G}$, an appropriate definition of a systematic parity check matrix is

$$\mathbf{H} = [-\mathbf{P}^{\mathbf{T}}\vdots\mathbf{I}] \tag{4}$$

where $\mathbf{I}$ is an identity matrix of dimension $(n-k)$, because then $\mathbf{G}\mathbf{H}^{\mathbf{T}} = \mathbf{0}$.

When a codeword is transmitted through a communication channel, it can become corrupted. The effect is the same as adding an n-tuple called the *error pattern* to the codeword. The $GF(q)$ sum of the codeword and error pattern gives the received word. If it is assumed that the number of errors is within the error-correction capability of the code, each correctable error pattern gives rise to a unique *syndrome*. The syndrome is generated from the received word by $GF(q)$ operations. Thus the syndrome helps to recover the original codeword from its corrupted received version. One way of generating the syndrome from the received word is to take its product with the transpose of the parity check matrix, ie.,

$$\mathbf{S} = \mathbf{r}\mathbf{H}^{\mathbf{T}} \tag{5}$$

where $\mathbf{S}$ is the syndrome corresponding to the received word $\mathbf{r}$. From the structure of $\mathbf{H}$, the structure of $\mathbf{H}^{\mathbf{T}}$ can be seen to be

$$\begin{bmatrix} -\mathbf{P} \\ \cdots \\ \mathbf{I} \end{bmatrix}$$

where, $\mathbf{I}$ is an identity matrix of dimension $(n-k)$. The structure of $\mathbf{H}^{\mathbf{T}}$, shows that the syndrome can be generated as the vector sum of the received parity symbols and the *additive inverse* of the parity symbols recomputed from the received information symbols. For any extension field of GF(2), the additive inverse of an element is the element itself.

3 NMR as a Repetition Code

Consider a repetition code, where there is one information symbol and $(N-1)$ parity symbols, which are copies of the information symbol. This is an $(N,1)$ repetition code. The codewords

are [0 0 ... 0] and [x x ... x], where x is any member from the code alphabet with q symbols. This is a special case of linear codes. This code has a minimum distance of N and hence can correct t errors where $N \geq (2t+1)$. Consider a received word $[r_1\, r_2 \ldots r_N]$. From the theory of linear ECC, we can find the syndrome corresponding to this received word as follows. First, we recompute the parity symbols from the received word. Notice that for a repetition code, the parity symbols are copies of the information symbol itself. Thus the reconstructed parity symbol vector corresponds to the $(N-1)$-tuple $[r_1 r_1 \ldots r_1]$. To generate the syndrome, we find the GF(q) vector sum of the additive inverse of the recomputed parity symbol vector and the received parity symbol vector. That is, the syndrome

$$S = [(-r_1 \oplus r_2)(-r_1 \oplus r_3) \ldots (-r_1 \oplus r_N)] \tag{6}$$

where $\oplus$ represents GF(q) addition. Thus the syndrome of the error pattern is obtained by the GF(q) addition of the additive inverse of the received information symbol with each of the received parity symbols. For all correctable errors, the syndrome uniquely determines the error pattern, ie., it shows which symbols are in error and by how much.

An NMR system resembles a repetition code in that the module is replicated just as the information symbol is replicated in the repetition code. The modules in an NMR system can be either good or faulty. A faulty module can, in general, take an undefined state but the module output is always equivalent to an element from GF(q), where $q = 2^m$, m being the number of module output lines. In other words, we are representing the module outputs as GF(q) symbols. Imagine that the first module in an NMR system corresponds to the information symbol of a repetition code, and the other modules correspond to the parity symbols. Consider that the first module is GF(q) added with every other module using $(N-1)$ two-input GF(q) adder units. Assume that the adder units are fault-free. This system is shown in Figure 1, where the c_i's represent GF(q) addition.

The following theorem can be proved. Some terms are defined first.

Definition 5 *A Module Fault Pattern is an N-bit vector over GF(2), with weight less than or equal to* $\lfloor \frac{N-1}{2} \rfloor$, *where 1's represent faulty modules and 0's represent good modules by position in an NMR system. The most significant position represents the first module and the least significant position represents the last module.*

Definition 6 *A Fault Signature is a vector over GF(q) whose symbols are the outputs of the GF(q) adders for a particular fault pattern. The symbols of the fault signature are assigned by position; they depend on the physical position of the corresponding adder in the system.*

Theorem 1 *Assuming fault-free adders in an NMR system where a particular module output is GF(q) added with the outputs of all other modules as in Figure 1, there is a unique correspondence between all module fault patterns of* $\lfloor \frac{N-1}{2} \rfloor$ *faulty modules or less, and fault signatures, so that the faulty and fault-free modules can be identified.*

Proof:

Consider the equivalent (N,1) repetition code. This code has a minimum distance of N. Therefore it can generate unique syndromes for all error patterns of $\lfloor \frac{(N-1)}{2} \rfloor$ errors or less. The NMR system is configured such that the first module output is GF(q) added with

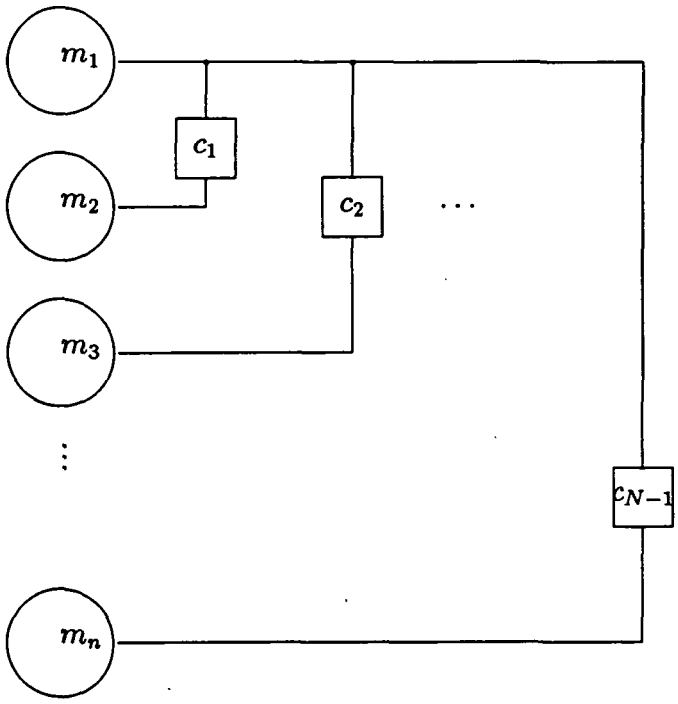

Figure 1: Star-Connected NMR System

every other module output to generate the fault signature. The syndrome of the received word is generated by the GF(q) addition of the additive inverse of the received information symbol with the received parity symbols. Since the module output lines have boolean values (elements of GF(2)), and q is equal to 2^m, GF(q) is always an extension field of GF(2). Therefore, the additive inverse of any element in GF(q) is the element itself. Thus the given configuration of the NMR system exactly determines the syndrome of the fault pattern which in turn gives the position of the failed module and what its output should have been. Therefore, the fault signature is identical to the syndrome of the corresponding error pattern or the module fault pattern. □

Thus diagnosis of an NMR system is exactly similar to error correction using a repetition code if the GF(q) arithmetic modules are fault-free. If multiple module failures are assumed to be common-mode in nature, then the code reduces to a binary code (the only elements of the alphabet are good and faulty, corresponding to 0 and 1 arbitrarily), and the GF(q) adder blocks reduce to comparators. This leads to very simple systems [3]. The same situation results if single element failures alone are considered.

4 Extending the Code to Cover Checker Units

In the previous section, the similarity between an NMR system and an (N,1) repetition code was elaborated. One of the features of the implementation was that the determination of the syndrome had to be error-free, ie., the GF arithmetic units had to be assumed fault-free. This is because the repetition code only covers the modules; it does not cover the checker units. This observation opens up a new possibility. It seems as if an encoding scheme that

somehow includes the checker units also might lead to a system which can also tolerate checker units failures. This would be a very desirable property indeed. In this section, this idea is pursued.

Consider an NMR system capable of tolerating t faulty modules. Let each module have an output bus with m lines. Thus the module outputs can be represented by the elements of GF(q), where $q = 2^m$. In other words GF(q) is an extension field of GF(2) and hence the additive inverse of an element in GF(q) is the element itself. Consider a t error-correcting linear code over GF(q) with N information symbols and p parity symbols. There will be q^N valid code words, and the minimum distance of the code is $(2t + 1)$. Each parity symbol will be a linear combination of some of the information symbols. We will constrain the generation of parity symbols thus; each parity symbol must be a GF(q) sum of an *even* number of information symbols. Assume that in an NMR system, the modules correspond to the information symbols and the checker units roughly correspond to the parity symbols. In the no-fault case for an NMR system, we are dealing with codewords whose information symbols are all the same. Notice that in all the cases, the parity symbols are all zeroes because of the way the code is constructed. Thus we know beforehand what the parity symbols should be when there are no faults. GF(q) adders are connected at the module outputs exactly the way the parity symbols are generated. The outputs of the adders are actually reconstructions of the parity symbols from the received information symbols. We know that the received information symbols (or equivalently, the output set of all the N modules) must be of the N-tuple form $[x\,x \ldots x]$ where x is any element from GF(q). All these N-tuples have the same parity symbols represented by the p-tuple $[0\,0 \ldots 0]$ because of the construction of the code. The GF(q) sum of the outputs of the p GF(q) adders and the p-tuple $[0\,0 \ldots 0]$ gives the "syndrome" of the fault pattern. If the fault pattern is correctable, ie., if t or less modules are faulty, the modules can be identified because the syndrome identifies the fault pattern. Notice that there can be faults in the adder units also (which correspond to faults in received parity symbols) as long as the total number of faulty elements does not exceed the correction capability of the code, t. A total of t faulty elements can be identified unambiguously using this scheme. This will be formalized by the following theorem.

Theorem 2 *Consider an NMR system whose module outputs are considered as elements of GF(q), an extension field of GF(2). Consider also a minimum distance N linear ECC with N information symbols and p parity symbols whose parity symbols are all GF(q) sums of an even number of information symbols. A network of p GF(q) adders is connected at the output of the NMR system, such that each adder takes as input the modules corresponding to the information symbols that make up a single parity symbol. This system gives fault signatures that uniquely identify all possible fault patterns of* $\lfloor \frac{N-1}{2} \rfloor$ *faulty elements or less.*

Proof:
The system described here corresponds to a linear code, where the module outputs correspond to the information symbols and the GF(q) adders roughly correspond to the generation of parity symbols. Also, since GF(q) is an extension field of GF(2), additive inverses of field elements are the elements themselves. Due to the structure of the code, the fault-free case corresponds to code words of the structure $[x\,x \ldots x(information N-tuple)0 \ldots 0(parity p-$

tuple)], where x is an element of GF(q). In this proof, we will assume that the number of faulty elements does not exceed the capability of the code.

Case A: Faults confined to modules

Consider a received word whose information part is identical to the N-tuple formed by the module outputs. We know apriori, that the correct parity p-tuple is the all zero p-tuple. The received (N+p)-tuple obviously is a correctable received word, as far as the code is concerned, as long as the number of corrupted symbols (module outputs) is less than or equal to the error-correction capability of the code. Notice that the parity symbols (adder outputs) recomputed from the received information symbols (module outputs) are identical to the corresponding symbols of the fault signature generated by the GF(q) adders. Since there is no error in the received parity symbols, the fault signature is identical to the syndrome of the error patter. Thus the faulty modules can be uniquely identified.

Case B: Faults confined to GF(q) adders

The parity symbols corresponding to the faulty adders are corrupted and all the other parity symbols are 0's. Consider a received word whose information part is identical to the module outputs (all correct and identical), and parity symbols are all zeroes except the ones that are in error which correspond to the locations of the failed adders. Obviously, the syndrome of the error pattern is the same as the parity part of the received word, which is identical to the fault signature. Thus the faulty GF(q) adders are identified from the syndrome.

Case C: Faults mixed among the modules and GF(q) adders

Consider a received word whose information part is identical to the module outputs and whose parity symbols are zeroes in all positions except the ones corresponding to the faulty adders. The syndrome is generated by recomputing the parity symbols and then GF(q) addition of the recomputed parity symbols with the received parity symbols. On close inspection, it can be seen that the resultant p-tuple is identical to the fault signature. The adders recompute the parity with errors corresponding to the positions of the faulty adders. This is the same as if the parity symbols were recomputed without error but the received parity symbols corresponding to the positions of the faulty adders were in error. Hence the fault signature is identical to the syndrome of the error pattern and the failed elements can be identified from the fault signature.

To conclude the proof, if there are no faulty elements, the fault signature will be the all zero vector. □

It is obvious that we are not using all the redundancy inherent in the coding structure, since we are really dealing with only a few codewords. But this is a necessary evil since we want the effects of error-correction to extend over the computation of the syndrome. This is the main difference from the theory presented in the previous section.

5 A Systematic Way of Generating Required Codes

It has been shown that we can create an NMR system that tolerates $\lfloor \frac{N-1}{2} \rfloor$ faulty elements by creating a special type of linear error-correcting code of minimum distance N. This section introduces one systematic way of doing this. This method is not unique; there could be other ways of doing this.

Theorem 3 *Consider a linear block code over GF(q), where $q = 2^m$, with q^N code words, ie., N information symbols. If ${}_NC_2$ parity symbols are appended to the information symbols such that each parity symbol is the GF(q) sum of a unique pair of information symbols, the resulting code has minimum distance equal to N.*

Proof:
We will first show that the minimum distance is at least N. Let the information symbols be represented by the N symbol vector $\hat{\mathrm{I}}$, and the parity symbols be represented by the ${}_NC_2$ symbol vector $\hat{\mathrm{P}}$. For any two code words i and j, $\hat{\mathrm{I}}_i$ and $\hat{\mathrm{I}}_j$ differ in a positions, where $0 < a \leq N$. Consider one of those differing symbol positions (For example, consider that the left most position in $\hat{\mathrm{I}}_i$ and $\hat{\mathrm{I}}_j$ are different). The parity vectors $\hat{\mathrm{P}}_i$ and $\hat{\mathrm{P}}_j$ have parity symbols corresponding to the GF(q) sum of the each information symbol with each other information symbol. $\hat{\mathrm{I}}_i$ and $\hat{\mathrm{I}}_j$ agree on $(N - a)$ positions. Consider the parity symbols generated by a symbol at a differing symbol position (for example the left most symbol in $\hat{\mathrm{I}}_i$ and $\hat{\mathrm{I}}_j$ in the example being considered) with these $(N - a)$ agreeing positions. It is clear that $\hat{\mathrm{P}}_i$ and $\hat{\mathrm{P}}_j$ differ in at least $(N - a)$ positions. But $\hat{\mathrm{I}}_i$ and $\hat{\mathrm{I}}_j$ differ in a positions. Therefore any two code words differ in at least N positions.

Now we will show that the minimum distance is not more than N. Consider the codewords whose information vectors are the all zero vector and the all x vector, where x is any element of GF(q). These two code words have the same parity vectors equal to the all zero vector (notice that GF(q) is an extension field of GF(2)). Obviously, the distance between these two codewords is N. Therefore, the minimum distance of the code is not greater than N. The two results together prove the theorem. □

An example 5MR system is shown in Figure 2. The X_i's represent GF(q) adders. In this system a maximum of two faulty elements (the code being used is a double error-correcting code) can be identified, irrespective of the positions of the faulty elements, including the adders.

6 An Example; Following a (7,4) Hamming Code

As was mentioned in Section 3, if only a single faulty element needs to be tolerated, the code reduces to a binary code, and results in easy implementation of the NMR system. Additionally, since only one failed element is anticipated, GF(2) addition on module outputs can be simulated by a comparison operation if we arbitrarily assume that the element '1' represents a failed module, and the element '0' represents a good module. In Section 4, we introduced the need to compose a parity symbol from an even number of information symbols. This was to ensure that in a fault-free NMR system following the code, a GF(q) addition on the module outputs would always yield the additive identity (**0**) since an even number of module outputs are being added in GF(q) which is an extension field of GF(2). This results in the advantageous situation where we know apriori, what the parity symbols in an error-free case would be. In the case of a single failed element represented by the element 1 of GF(2), a straight comparison operation always simulates a GF(2) addition since there can be at the most one differing input to the comparator. Also, since the fault-free element is represented by element 0 (the additive identity of GF(2)), no matter how many fault-free

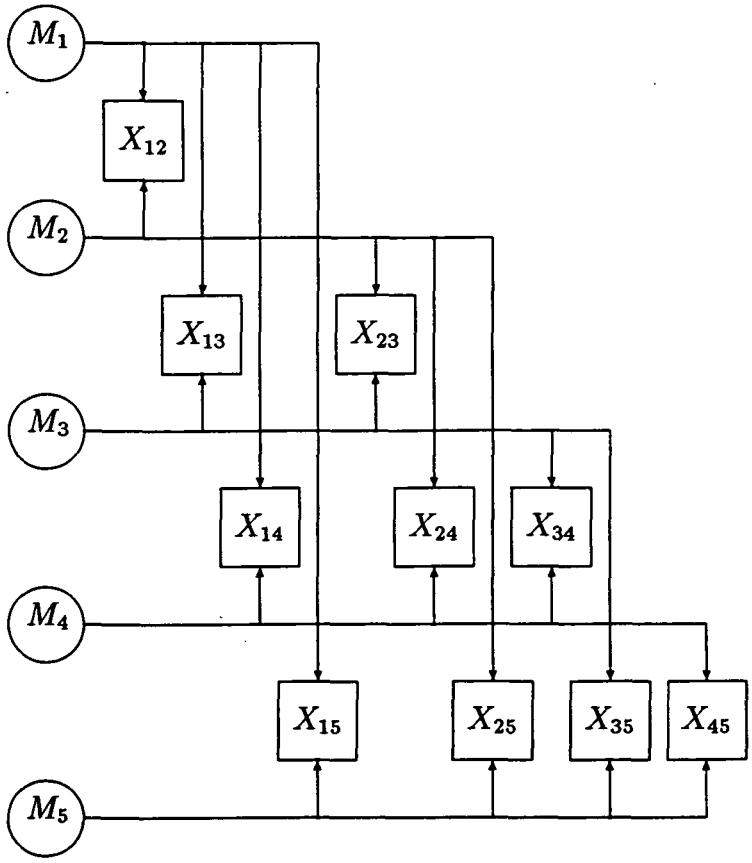

Figure 2: Linear ECC Implementation of 5MR System

Failed Element	Comparator Outputs	Corresponding Error Pattern	
	$C_1\ C_2\ C_3$	Information	Parity
	$p_1\ p_2\ p_3$	$i_1\ i_2\ i_3\ i_4$	$p_1\ p_2\ p_3$
M_1	1 1 0	1 0 0 0	0 0 0
M_2	1 0 1	0 1 0 0	0 0 0
M_3	1 1 1	0 0 1 0	0 0 0
M_4	0 1 1	0 0 0 1	0 0 0
C_1	1 0 0	0 0 0 0	1 0 0
C_2	0 1 0	0 0 0 0	0 1 0
C_3	0 0 1	0 0 0 0	0 0 1

Table 1: Relation to (7,4) Hamming Code

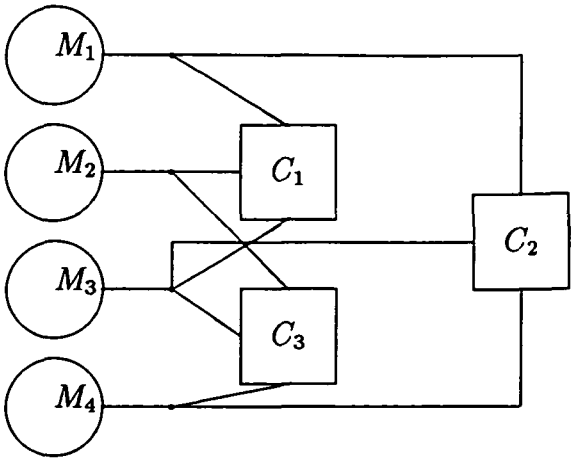

Figure 3: (7,4) Hamming Code Example

elements are compared, the result would be 0. Hence there is no need to compose the parity bits using an even number of information bits, in the corresponding binary linear code. These principles are illustrated in the example.

Consider a (7,4) binary Hamming code. The i's represent information bits and the p's represent parity bits. Note that a Hamming code has single error correction capability. We are trying to isolate a single faulty element. The parity bits are generated as follows.

$$\begin{aligned} p_1 &= i_1 + i_2 + i_3 \\ p_2 &= i_1 + i_3 + i_4 \\ p_3 &= i_2 + i_3 + i_4 \end{aligned}$$

Notice that the parity bits are composed of an odd number of information bits. This is allowed since we are considering the fault-free modules to be represented by the additive identity of GF(2). Let the modules being checked be represented by the information bits. Let bus-bus comparators be connected at the outputs of the modules according to the way the parity bits are generated; ie., one three-way comparator compares the outputs of modules $M_1, M_2, \& M_3$, a second comparator compares the outputs of $M_1, M_3, \& M_4$, and a third comparator compares $M_2, M_3, \& M_4$. The relations are summarized in Table 1 given. The system is shown in Figure 3.

The special feature of this implementation is that the comparators roughly correspond to the parity bits. Thus the comparators are also included as part of the code. The outputs of the comparators indicates which element is faulty. Notice that when a module is faulty, the outputs of the comparators indicate the syndrome of the fault pattern (or the error pattern in the equivalent code). For example, when the module M_1 is failed, the comparators output the vector [1 1 0]. This vector corresponds to the syndrome of the error pattern [1 0 0 0 0 0 0], where the first four positions represent the information bits which are the modules in the equivalent NMR system, and the last three represent the parity bits which are the comparators. Since 1 represents a failed element by our representation, we know that module M_1, corresponding to the first information bit is the failed element. As another example, consider the case where comparator C_1 is failed. Now the comparators output the vector [1 0 0], which is the same as the syndrome of the error pattern [0 0 0 0 1 0 0]. This error

pattern indicates that the first parity bit is in error, or in the equivalent NMR system, the first comparator is failed. Similarly, all possible single element failures are uniquely identified by this system.

7 Conclusions

In this paper, we explored the links between the use of N-Modular Redundancy for hardware fault tolerance, and the use of linear Error-Correcting Codes. We showed how NMR is an application of a (N,1) repetition code, and also showed how special encoding structures can be designed to cover failures in the checker units themselves. We adapted some well-known results from coding theory for use in NMR for fault tolerance. The rich mathematical theory that finds application in error-protection techniques in digital data communication has faces that touch other subjects. The use of modular redundancy for hardware fault tolerance is one of them.

Acknowledgements– This research was supported in part by NASA under Space Engineering Research Center Grant NAGW-1406. The authors would like to express sincere gratitude to Barbara Martin for help in the preparation of the manuscript. The first author acknowledges the help of colleagues, especially Suresh Gopalakrishnan, at the California Design Center of Hewlett-Packard Company.

References

[1] R. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, May 1984.

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

[3] V. Bobin, "Modular Redundancy and the Theory of Linear Codes", Ph D Dissertation, Dept. of Elec. Engr., University of Idaho, 1992.