C. V. Freiman
R. T. Chien

# Further Results in Polynomial Addressing

The problem of efficiently retrieving documents and other information from large-scale memory and file systems has generated much renewed interest in the comparatively old area of "hash-" or key-addressing. Among the more promising approaches taken has been the application of some results of the theory of group codes, as typified by the papers of Schay and Raver[1] and of Hanan and Palermo.[2] Common to the work of these authors and some of the earlier work of Muroga[3] is the idea of choosing a group code of the minimum-distance type and assigning addresses to keys according to the group-code coset in which particular keys fall. The most interesting class of group codes from an implementation point of view is that in which encoding and decoding is accomplished by means of shift-registers for polynomial multiplication and division. We shall restrict our attention to this class and shall refer to the use of such codes as *polynomial addressing.*

Partly for clarity of presentation, but primarily because of far greater ease of implementation, we shall restrict our discussion to binary systems and shall not consider higher order systems such as that presented in Ref. 1. It will also be assumed that the reader is familiar with Ref. 1 and hence no general discussion of polynomial addressing will be included.

In this Letter we first point out that all polynomials used to form $m$-bit addresses will insure that no pair of keys whose differing positions are within a span of $m$ bits will ever be assigned the same address. We will then illustrate that, for fixed $m$, a burst-error-correcting code may often handle more significant classes of key clusters than the minimum-distance codes used in Refs. 1 to 3.

### Polynomial addressing

Let us consider a set of keys, each of which is specified by a sequence of $n$ binary digits. We may then use polynomials with binary coefficients to represent these sequences in a systematic manner. For instance, if $n = 10$ and the ten binary digits are 1 1 1 0 0 0 1 1 0 1, the polynomial associated with this sequence will be $x^9 + x^8 + x^7 + x^3 + x^2 + 1$. In general, the $n$-bit binary sequence may be written as $A_{n-1}, A_{n-2} \cdots A_1, A_0$, and its polynomial representation as

$$A(x) = A_{n-1}X^{n-1} + A_{n-2}X^{n-2} + \cdots + A_1x + A_0.$$

These coefficients are considered to belong to the field of two elements. In this field, addition and multiplication are specified by the following addition and multiplication tables:

| $+$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| $\cdot$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

In this manner, each key has its associated polynomial, and different keys have different polynomial representations.

Now suppose there is another polynomial $M(x)$ of degree $m$, where $m < n$; we may divide each polynomial $A(x)$ by $M(x)$ and obtain a remainder $R(x)$. The division process can be carried out in ordinary manner. For instance, if $A(x) = x^9 + x^8 + x^7 + x^3 + x^2 + 1$ and $M(x) = x^5 + 1$, then

$$x^9 + x^8 + x^7 + x^3 + x^2 + 1$$
$$= (x^4 + x^3 + x^2)(x^5 + 1) + (x^4 + 1).$$

In general, $A(x) = Q(x)M(x) + R(x)$, where the degree $r$ of $R(x)$ is less than the degree $m$ of $M(x)$. The key-to-address transformation we wish to explore in detail is the transformation $A(x) \rightarrow R(x)$.

It is clear that among all such transformations, choosing $M(x) = x^m + 1$, will permit the simplest implementation. Shown in Fig. 1 is a circuit utilizing a single delay line and a single half-adder which achieves this interlacing of $m$ parity checks. Control circuitry is not shown.

In choosing among polynomials of the form

$$x^m + C_{m-1}x^{m-1} + C_{m-2}x^{m-2} + \cdots + C_0(C_i = 0, 1),$$

some thought should be given as to the types of key clusters they will "break up". It can be shown that, if $C_0 = 1$, the addressing system will break up clusters in which the difference between any pair of keys can be spanned by $m$ bits. This is equivalent to the statement that, if $C_0 = 1$, then $M(x)$ is a generator polynomial of an error-detecting code that detects all single-burst errors of length not greater than $m$. A proof of the latter statement is given in Ref. 4.

### Use of burst-error-correcting codes

Previous workers[1,2] have considered polynomials which correspond to random-error correcting codes and which, therefore, have the property that they will break up clusters in which two keys differ in no more than some specified number of positions. We now strongly suggest the consideration of polynomials which correspond to codes that permit the correction of burst errors. These will have the property that all clusters will be broken up provided the positions in which a pair of keys differs can be covered by *two* spans of a specified number of bits.

For example, let us consider the following case. The memory system to be addressed consists of $2^{24}$ locations. Using a Bose-Chaudhuri code[5] which permits the correction of four random errors, we obtain the polynomial

$$M_1(x) = (x^6 + x + 1)(x^6 + x^4 + x^2 + x + 1) \cdot (x^6 + x^5 + x^2 + x + 1)(x^6 + x^3 + 1).$$

Using $M_1(x)$, keys of up to 63 bits can be accommodated with pairs of keys differing in 8 or fewer positions never being assigned the same address. Use of a Fire burst-error correcting code[6] in this case enables keys of effectively unlimited length (up to 7665 bits) to be handled without duplication of address whenever a pair of keys differs in at most 16 positions, spannable by two blocks of 8 bits. A generator polynomial for such a code is

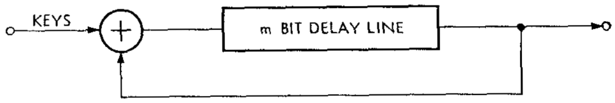

*Figure 1* **A simple circuit for polynomial addressing.**

$$M_2(x) = (x^{15} + 1)(x^9 + x^4 + 1).$$

Instrumentation of either case can be accomplished through the use of a binary shift register with $m$ stages and several logic units.[7] The cost of any binary polynomial addressing circuit is thus quite reasonable, the circuit of Fig. 1 being the most inexpensive of all.

### References

1. G. Schay and N. Raver, "A Method for Key-to-Address Transformation", *IBM Journal* 7, 121 (1963).
2. M. Hanan and F. P. Palermo, "An Application of Coding Theory to a File Addressing Problem", *IBM Journal* 7, 127 (1963).
3. S. Muroga, unpublished report, April, 1961.
4. W. W. Peterson and D. T. Brown, "Cyclic Codes for Error-Detection", *Proc. IRE* **49**, 228 (1961).
5. R. C. Bose and D. K. Ray-Chaudhuri, "Further Results on Error Correcting Binary Group Codes", *Information and Control* **3**, 279 (1960).
6. P. Fire, "A Class of Multiple-Error-Correcting Binary Codes for Non-Independent Errors", Sylvania Report RSL-E-2 (1959).
7. W. W. Peterson, *Error-Correcting Codes*, John Wiley and Sons, New York, 1961, Chap. 8.

*Received May 10, 1963*