X-711-69-384
PREPRINT

NASA TM X- 63730

GRAPHS, NUMBER SYSTEMS, AND ARITHMETIC CODES

OSCAR N. GARCIA

SEPTEMBER 1969

GSFC

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-711-69-384

GRAPHS, NUMBER SYSTEMS, AND ARITHMETIC CODES

Oscar N. Garcia
Department of Electrical Engineering
Old Dominion University
Norfolk, Virginia

September 1969

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

GRAPHS, NUMBER SYSTEMS, AND ARITHMETIC CODES

Oscar N. Garcia
Department of Electrical Engineering
Old Dominion University
Norfolk, Virginia

PREFACE

This paper reports on some preliminary results obtained in using graphs to represent number systems and in the converse question of what class of graphs represents number systems. Because arithmetic codes are finite redundant number systems, the motivation for this work is the improvement of the reliability of digital computer arithmetic through redundancy in the information processed. More questions have been raised in the process of determining the class of graphs which are "equivalent" to rumber systems than have been answered. This document, therefore, is more in the nature of a progress report than of a completed work.

CONTENTS

GRAPHS, NUMBER SYSTEMS, AND ARITHMETIC CODES

INTRODUCTION

We will be concerned here with the representation of the integers forming an algebraic (additive) group by sets of binary strings of a given identical length (usually called "word length"). Let Z_m be the additive group of integers modulo m. We will be interested in the rules that will govern the addition of the elements of the set of binary strings or words v_i which form the binary code V (called "code words") such that the group homomorphism α holds for all z_i, $z_j \in Z_m$ and for v_i, $v_j \in V$.

$$\alpha(z_i \oplus z_j) = \alpha(z_i) + \alpha(z_j) = v_i + v_j \tag{1}$$

where α: $Z_m \to V$ is an onto mapping, and $(Z_m, \oplus)$ and $(V, +)$ are the groups in question.

We will furthermore assume the following rules for addition on the elements of the binary strings:

$$0 + 0 = 0$$

$$1 + 0 = 0 + 1 = 1$$

$1 + 1 = 0 +$ a carry to the next position to the left and possibly other positions to the right. Examples will be given in the following sections.

With each position of the binary string we may associate two weights.

POSITIVELY WEIGHTED ARITHMETIC CODES

When a weight is associated with the elements of the string, the inverse mapping α^{-1}: $V \to Z_m$ corresponding to α as given in (1) is a function which may be represented in a general form:

$$\alpha^{-1}(v_j) = \sum_{i=0}^{n-1}\left[a_i \cdot w(a_i)\right] + K$$

where K is an integer constant usually zero. By $w(a_i)$ we mean that the weight depends not only on the position in the string, but also on the element itself. For the binary case we have double weighted number systems in which a weight is attached to zero and another to one, in each position of the binary string. In that case

$$\alpha^{-1}(v_i) = \sum_{i=0}^{n-1} (b_i w_i + \bar{b}_i \cdot w_i') + K$$

where $\bar{b}_i$ is one when b_i is zero, and vice versa, and w_i and w_i' are the two weights corresponding to that position.

Example 1: Let $n = 4$ in the binary case and consider the usual excess 3 code of ten elements from 0011 to 1100. As a double weighted code with $K = 0$ and

i	w_i	w_i'
0	0	-1
1	0	-2
2	4	0
3	8	0

we have for example:

$$\alpha^{-1}(1000) = \bar{0} \cdot (-1) + \bar{0} \cdot (-2) + \bar{0}(0) + 1 \cdot (8) = 5$$

Double weighted binary systems have some practical applications.[1]*

Most commonly, however, we have a mapping α^{-1} with $w_i' = 0$ for $i = 0, \cdots, n-1$, and $K = 0$. These are single weight systems.

*Numbers in square brackets refer to the Bibliography.

Example 2: Consider the set of triples 000,001,010,011,101 which maps onto Z_5 according to $w_0 = 1$, $w_1 = 2$, $w_2 = 3$, $w_i' = 0$ for $i = 0, 1, 2$ and $K = 0$. Then we have for example:

$$\alpha^{-1}(101) = 1 \cdot 1 + \bar{0} \cdot 0 + 1 \cdot 3 = 4$$

Single weight binary number systems with $K = 0$ have the general mapping:

$$\alpha^{-1}(v_j) = \sum_{i=0}^{n-1} b_i w_i \qquad (2)$$

When $w_i = r^i$ we have a consistently based radix r number system. If $r < 0$ we have a negative radix number system[2].

In what follows we will always assume that $w_i > w_i' = 0$ for all i, and will deal exclusively with binary codes; i.e., number systems as in (2) where b_i is one of 0 or 1.

CODING FOR ERROR CHECKING

To perform addition and subtraction a computer number system must be a finite algebraic group because of the finiteness of the number representations in a computing machine. Garner[3] has shown that a ring model is also possible for the usual complement coded number systems (radix and diminished radix, consistently based, binary) when multiplication is also included.

The improvement of the reliability of computer arithmetic units through the use of codes must naturally start with some method which checks at least the basic operations of addition and complementation. Therefore, we can, from an abstract point of view, raise the question of what are the possible methods of checking the group operation.

Since the mapping given by (2) above will be assumed, an element v of our code V will be represented as an n-tuple

$$v = [b_{n-1}, b_{n-2}, \cdots, b_0]$$

where b_i is either 0 or 1 for all i. The cardinality of the set of all such n-tuples is 2^n. If all n-tuples were included in the number system any error in addition would produce a result which would be in the number system. The strategy for error checking is then to exclude a well chosen set of n-tuples from the number system representations so that certain errors in the process of addition may be identified if they occurred. This of course is equivalent to introducing redundant bits in the representation of the elements of the number system, which now has cardinality smaller than 2^n. If the cardinality of this number system V is m, we must still have the homomorphism $\alpha: Z_m \to V$.

Example 3: Let $m = 5$ and consider the mapping α of Z_5 into (a subgroup of) Z_{15}, given by the relation $\alpha(x) = 3 \cdot x$ for x in Z_5. With a consistently based radix 2 representation we use three bits for Z_5 and four for the subgroup V of Z_{15}.

$$\left.\begin{array}{l} 000 \to 0000 \\ 001 \to 0011 \\ 010 \to 0110 \\ 011 \to 1001 \\ 100 \to 1100 \end{array}\right\} V \subset Z_{15}$$

This is a homomorphism if addition is taken as modulo 5 addition in Z_5 and modulo 15 addition in V.

It should be clear that there is redundancy in the representation of the subgroup V of Z_{15}.

THE DIRECTED GRAPH REPRESENTATION OF A NUMBER SYSTEM

We have said very little, so far, on the way arithmetic is to be performed in the number systems to which we have restricted ourselves. In particular we must clear the matter of how carries flow in a given number system. We will use the following graph representation: each vertex will represent one of the positions in the n-bit block, plus one extra vertex will be used to represent the modulus m of the system. The branches of the graph will represent the carry flows from each one of the positions with only one carry flowing into the modulus vertex. Notice that now our constraints on the flow of carries to the next (left) position in the block, and possibly, to positions to the right is explicitly shown. We will associate a weight with each vertex.

Example 4: Consider the diminished radix, 3-bit number system ($m = 7$) as shown below in Figure 1, with its end-around carry.

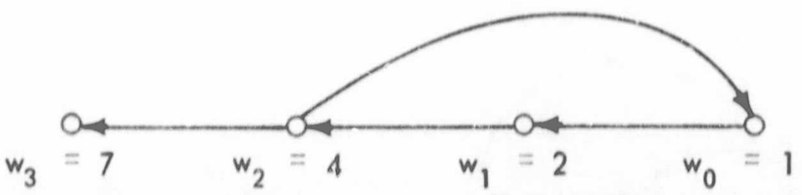

Figure 1.

Example 5: Figure 2 shows a 5-bit radix number system ($m = 32$).

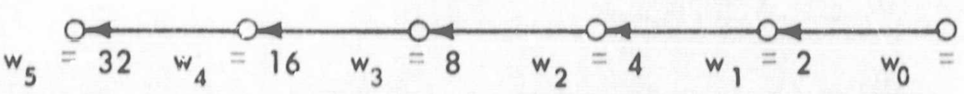

Figure 2.

Notice that this system has no carries flowing back, and that it is also the only n-bit system which can represent exactly 2^n different numbers.

Example 6: Figure 3 shows the graphs representing two possible 3-bit modulo 5 number systems.

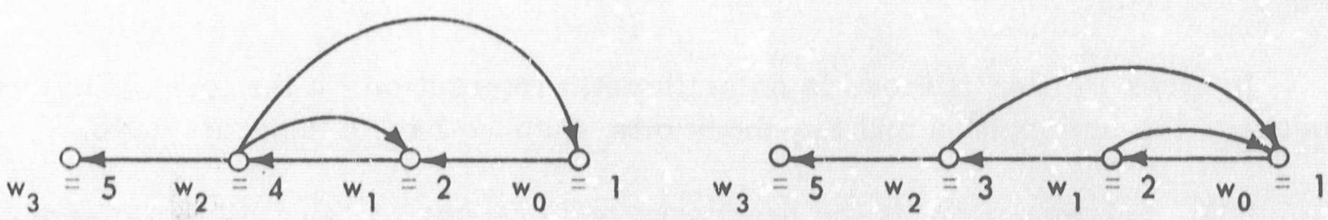

Figure 3.

For the same modulus we may have different systems under our assumptions. The number system represented by the graph on the right is more redundant in the sense that two representations for the integer 3 are possible.

Example 7: A number system may also be formed by the Cartesian product of two or more number systems. Residue number systems are often formed this way. Consider the five-bit number system formed by the concatenation of two number systems representing Z_3 and Z_5. A total of $LCM(3,5) = 15$ representations are possible in $Z_3 \times Z_5$. We show in Figure 4 such a number system. The symbols w_1' and w_4' are used in that figure to emphasize the fact that they do not correspond to weights of actual bit positions.

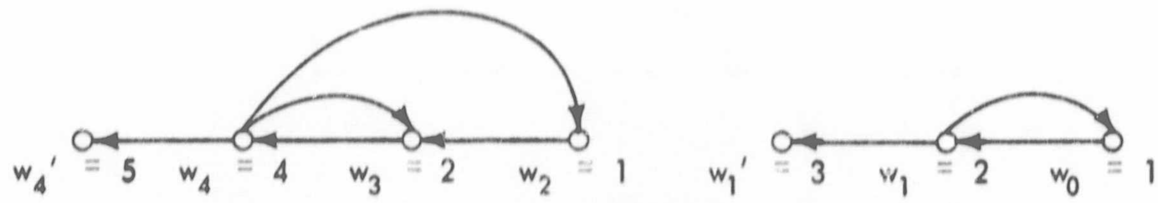

Figure 4.

CLASSIFICATION OF ARITHMETIC CODES

An undirected graph is said to be connected if every pair of distinct vertices may be joined by a sequence of branches. If directions are neglected in a directed graph (di-graph) like the carry-flow graphs described before, and if this now undirected graph is connected, then the di-graph is said to be (at least weakly) connected. A di-graph may consist of one or several connected subgraphs.

We now propose the following classification: if the graph representing the carry rules for a code is such that the information vertices are not connected to any of the check vertices, we have a separate code; otherwise we have a non-separate code.

In other words: if there is no arithmetic interaction (in the form of carries) between the information and the check bits, then we have a separate code.

The implementation of the processor and checker for an arithmetic separate code is given in Figure 5, where $c(a)$ means the check bits associated with a, and similarly for the others.

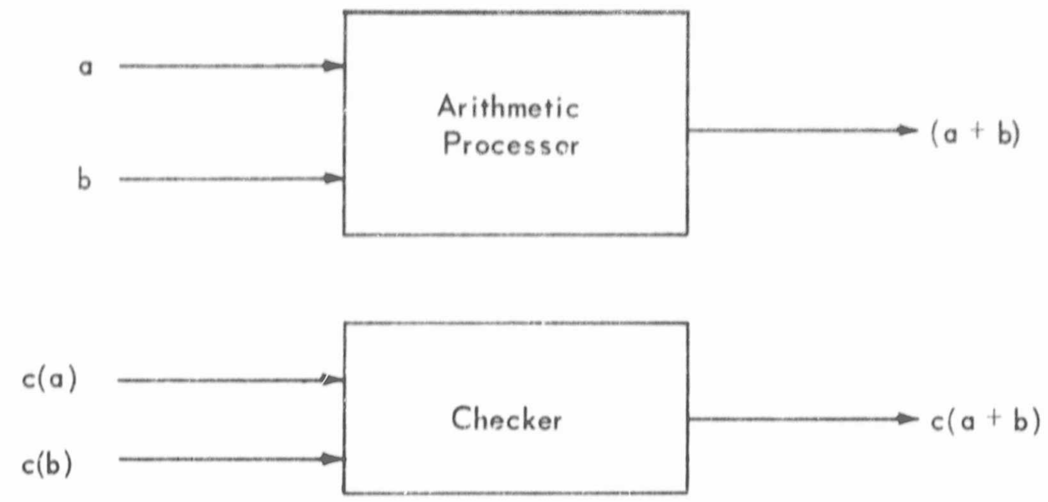

Figure 5.

Example 8: A code formed using the carry rules of Figure 4 as explained in example 7 is separate.

Example 9: A six-bit code generated by the linear combinations of

(100001), (010010), (100100), (011000)

under the arithmetic defined by the carry rules of Figure 6 (for the four right-most positions representing information vertices) is a non separate code since there is a carry flowing from the information vertex associated with w_3 to the check vertex associated with w_4.

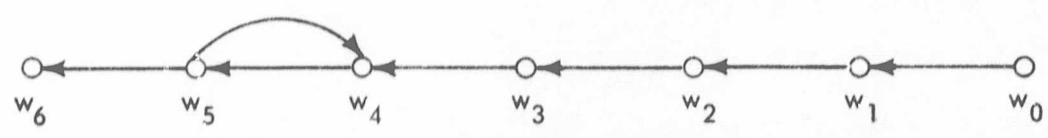

Figure 6.

It is evident that to classify a code as above we must be able to know a priori which are the information bits and which are the check bits. This is not possible in general. A code classification may be made on the basis of whether the information bits (that is, the bits of the binary representation of Z_m) may or may not be directly identified in the ones and zeros representation of the elements of the code V. The following definition accomplishes this:

Definition: If each of the bits of the binary representation of the elements of Z_m may be identified with an identical bit in a fixed position of the corresponding element in V mapped by α, then the code V is said to be systematic.

Notice that we are not asking that the corresponding bits in the elements of V be adjacent, but just that they identically correspond, from fixed positions, to the bits of the elements of Z_m.

Example 10: Consider the four-bit code generated by the multiples (additions of the generator to itself) of (0011) according to the rules of Figure 7. There are (as in all diminished radix systems) redundant representations for the zero of the number system. Arbitrarily, let us choose 0000 for zero. This is the same V of example 3.

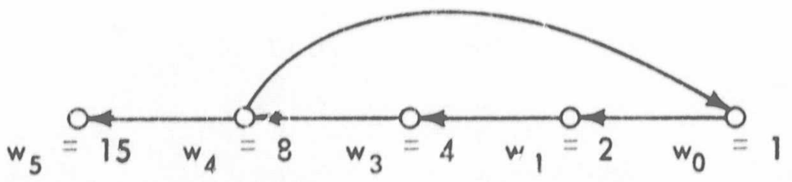

Figure 7.

Let the elements of the code V be {0000, 0011, 0110, 1001, 1100} which will represent {0, 1, 2, 3, 4} in the same order. This code is non-systematic, i.e., there is no possible assignment of the bits of {000, 001, 010, 011, 100} to fixed positions of the bits of the code words so that the information bits remain unchanged.

Example 11: Consider the code

000 → 00011

001 → 00101

010 → 01010

011 → 01111

100 → 10001

101 → 10110

110 → 11011

111 → 11101

defined by the mapping α of Z_8 into the given V of binary quintuples. The left-most three bits of each code word correspond in the same order to the elements of Z_8, and the code is systematic.

If we analyze the carry rules, as is done in Figure 8 we see that this code is actually a separate code.

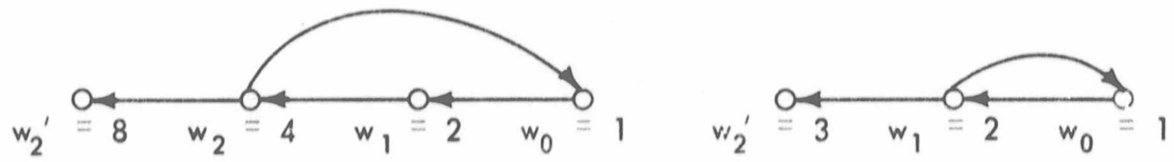

Figure 8.

Actually, all separate codes in which the information is totally contained in a connected graph are trivially systematic.

The code classification is then as follows:

- Arithmetic Codes
  - Systematic
    - Separate
    - Non-separate
  - Non-systematic

Notice that in the non-separate systematic codes there will be carries from the information portion to the check bits but not vice-versa.

THE MATRIX REPRESENTATION OF NUMBER SYSTEMS

Let the graph in Figure 9 represent a weighted number system.

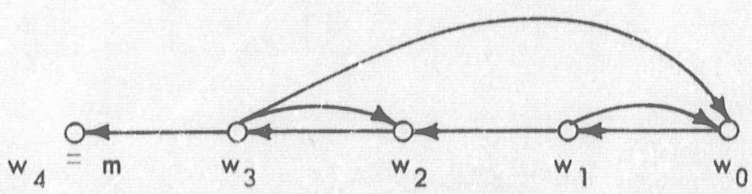

Figure 9.

We can write, taking into account the flow of carries described by the graph, the following system of linear equations

$$
\begin{aligned}
2w_0 - w_1 \quad\quad\quad\quad &= 0 \\
-w_0 + 2w_1 - w_2 \quad\quad &= 0 \\
2w_2 - w_3 &= 0 \\
-w_0 \quad\quad -w_2 + 2w_3 &= w_4
\end{aligned}
$$

each one of which says that when we have 1 + 1 in two corresponding bit positions one or more carries flow out of that position, and that the sum of the weights of the positions into which those carries flow is equal to twice the weight of the position from which they originate. If J_i is the set of positions into which carries originating from the i^{th} position flow, we can have the equations above as

$$2w_i = \sum_{j \in J_i} w_j \qquad (3)$$

for $i = 0, 1, 2, 3$.

Notice that w_4 acts like a "sink" and does not generate an equation.

Writing the system of linear equations in matrix form and letting $w_4 = m$ appear in the constant vector:

$$
\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & 0 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}
\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ m \end{bmatrix}
$$

which may be written

$$A \cdot W = M$$

where A is a matrix which may be called the carry matrix, and W and M are column vectors.

It is interesting to notice that there are $n + 1$ pieces of information not given by the graph associated with the n equations, i.e. we have $n + 1$ unknowns, namely the weights and the modulus of the system.

A CLASS OF GRAPHS WHICH REPRESENT WEIGHTED NUMBER SYSTEMS

In studying the graphs which represent the number systems considered before we find:

1. There exists a unique vertex with out-degree zero and in-degree one; this is labeled w_n or m and its weight is the modulus of the number system.

2. It is possible to order the vertices starting from w_n; the one connected to w_n is labeled w_{n-1} and it also has in-degree 1, but it may have out-degree greater than 1. If we remove all the out going branches from w_{n-1} we will find it connected to w_{n-2}. If we repeat the process with w_{n-2}; i. e., remove all its out-going branches we find w_{n-3}, and so on. (This algorithm was suggested by my research colleague R. A. Cliff.) This is possible because no carries flow forward, except one to the next digit position to the left. There is no restriction on the number of carries flowing backward; i. e., in the branches from w_i to w_j for any $j < i$.

On these graphs so ordered we can state the following:

Lemma. A necessary and sufficient condition for the graphs described before to represent number system with integer weights is that w_0 be an integer.

Proof: The necessity is obvious.

Let w_0 be an integer. The "equilibrium" or carry flow equations

$$2w_i = \sum_{j \in J_i} w_j = w_{i+1} + \sum_{j \in J_i'} w_j \qquad (4)$$

(where the elements of the set J_i are the subscripts of the outgoing branches from position i) will in general include the weight of the following vertex and possibly the weights of some preceeding ones. Then for $i = 0$ it must be that

$w_1 = 2w_0$. Similarly for any i each w_{i+1} is a linear combination of integers and must therefore be an integer.

Remark 1. To guarantee the strict positivity of the weights we must have $w_0 > 0$ and

$$2w_i > \sum_{j \in J_i'} w_j$$

where J_i' is the set of subscripts corresponding to the vertices connected by the outgoing branches from the i^{th} position, except the one to the $(i + 1)^{th}$ position.

Remark 2. In the class of number systems considered $w_1 = 2w_0$. Also, any $w_i = k_i w_0$ for some integer k_i. It is clear that no generality is lost if $w_0 = 1$, since this system will be isomorphic to one with any integer value for w_0.

Theorem: Given a number system with integer weights represented by the graph as described and with a matrix representation

$$A \cdot W = M$$

then

$$\det A = \frac{m}{w_0}$$

where m is the modulus of the system.

Proof: The matrix A is of the form:

$$\begin{bmatrix} 2 & -1 & & & \text{all zeros here} & & \\ a_{2,1} & 2 & -1 & & & & \\ a_{3,1} & a_{3,2} & 2 & -1 & & & \\ a_{4,1} & a_{4,2} & a_{4,3} & 2 & -1 & & \\ \vdots & \vdots & \cdots & \cdots & \ddots & \ddots & -1 \\ a_{n,1} & a_{n,2} & \cdots & \cdots & \cdots & \cdots & 2 \end{bmatrix}$$

where $a_{i,j}$ for $i < j$ is either 0 or -1; $a_{ii} = 2$; $a_{i,i+1} = -1$ and $a_{i,j} = 0$ for $j > i + 1$. Let $\Delta = \det A$. Then

$$w_0 = \frac{1}{\Delta} \det \begin{bmatrix} 0 & -1 & & & & \\ 0 & 2 & -1 & & \text{all zeros here} & \\ 0 & a_{3,2} & 2 & -1 & & \\ 0 & a_{4,2} & a_{4,3} & 2 & \ddots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & -1 \\ m & a_{n,2} & a_{n,3} & a_{n,4} & \cdots & 2 \end{bmatrix}$$

and expanding by minors we get a triangular matrix Δ' whose determinant is $(-1)^n$, where n is the number of vertices. Then

$$w_0 = \frac{(-1)^n m \Delta'}{\Delta} = \frac{(-1)^n m (-1)^n}{\Delta} = \frac{m}{\Delta}$$

$$\Delta = \frac{m}{w_0}$$

Because of the reasons given in the second remark above, we can let $w_0 = 1$ and then $\Delta = m$.

CONCLUSIONS

A number of interesting facts have been discovered in analyzing the relations between graphs, number systems and arithmetic codes. This generality allows a number of conclusions from study of graphs representing number systems. For example; whenever we have the carry flows given in Figure 10 we will have repeated weights; other similar patterns are easily identifiable.

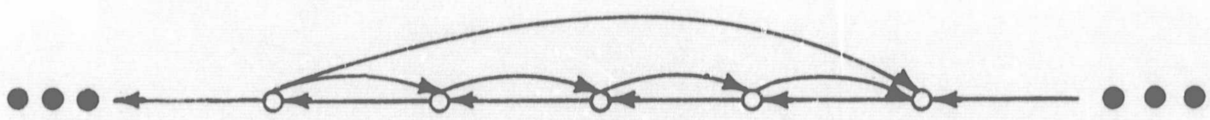

Figure 10.

We have just investigated superficially this topic, and have not even made use of practically any graph theory. It is hoped that this approach may encourage others to investigate arithmetic codes from a graph theory point of view, which seems quite appropriate, and in the process yield a stronger foundation to the theory of finite number systems.

ACKNOWLEDGMENTS

This work was done while the author was participating in the Summer Faculty Fellowship Program (sponsored jointly by NASA and ASEE) with the Flight Data Systems Branch of GSFC. The cooperation and assistance of the members of the Branch and, in particular, of my research colleague Rodger A. Cliff is gratefully acknowledged.

BIBLIOGRAPHY

(1) Walker, M., "A New Approach to Weighted Number Systems", Computer Design, July, 1969.

(2) DeRegt, M. P., "Negative Radix Arithmetic", Computer Design, May, 1967 through January, 1968.

(3) Garner, H. L., "A Ring Model for the Study of Multiplication for Complement Codes", IRE Transactions on Electronic Computers, vol. EC-8, pp. 25-30, March, 1959.