course, and unless some of the research is done by endowed organizations, the immediate costs are likely to be high.

There must also be a closer liaison between the digital data processing engineers and the communications engineers. As electronic data systems become more responsive, communicating and transmitting devices will be needed to connect the data processing center with the various segments of the system. Terminal data transfer and translation problems must be solved to permit the ultimate automation of data manipulation that is logically feasible.

Conclusion

The demise of the medium and large-scale general purpose electronic data processor or computer for business purposes is in sight. A sufficient number of industrial and commercial procedural analysts are now able to specify their data system requirements with cognizance of the speed and ability of electronic devices so as to build what is needed—not use just what is available. Small general purpose computers and large capacity computers for scientific calculation will continue in long usage.

Many large companies with special electronic data handling problems have found the traditional large manufacturers of business machines unwilling to do more than tie together existing standard lines of equipment. Often unwilling to entrust the smaller electronic manufacturers with their problems, several companies have embarked on their projects of tailor-made electronic systems.

I predict that this trend will continue until, or unless, some better-known companies enter the field.

The Residue Number System

HARVEY L. GARNER†

Introduction

IN THIS PAPER we develop and investigate the properties of a novel system, called the residue code or residue number system. The residue number system is of particular interest because the arithmetic operations of addition and multiplication may be executed in the same time as required for an addition operation. The main difficulty of the residue code relative to arithmetic operations is the determination of the relative magnitude of two numbers expressed in the residue code. The residue code is probably of little utility for general-purpose computation, but the code has many characteristics which recommend its use for special-purpose computations.

The residue code is most easily developed in terms of linear congruences. A brief discussion of the pertinent properties of congruences is presented in the next section.

Congruences

The congruence relationship is expressed as

$$A \equiv \alpha \bmod b$$

which is read, A is congruent to α modulo b. The congruence states that

$$A = \alpha + bt$$

is valid for some value of t, where A, α, b, and t are integers, α is called the residue, and b the base or modulus of the number A.

As examples of congruences, consider

$$10 \equiv 7 \bmod 3$$
$$10 \equiv 4 \bmod 3$$
$$10 \equiv 1 \bmod 3.$$

In these examples the integers 7, 4, and 1 form a residue class of 10 mod 3. Of particular importance is the least positive residue of the class which in this example is one. The least positive residue is that residue for which $0 \leq \alpha \leq b$.[1]

Consider the following set of congruences:

Given

$$A_1 \equiv \alpha_1 \bmod b$$
$$\vdots$$
$$A_n \equiv \alpha_n \bmod b.$$

Then

1) Congruences with respect to the same modulus may be added and the result is a valid congruence.

$$\sum_{i=1}^{n} A_i \equiv \left(\sum_{i=1}^{n} \alpha_i \right) \bmod b.$$

† University of Michigan, Ann Arbor, Mich.

[1] The equality sign may exist on only one side of the expression.

It follows that terms may be transferred from one side of a congruence to the other by a change of sign and also that congruences may be subtracted and the result is a valid congruence.

2) Congruences with respect to the same modulus may be multiplied and the result is a valid congruence.

$$\prod_{i=1}^{n} A_i \equiv \left(\prod_{i=1}^{n} \alpha_i \right) \bmod b.$$

It follows that both sides of the congruence may be raised to the same power or multiplied by a constant and the result is a valid congruence.

3) Congruences are transitive. If $A \equiv B$ and $B \equiv C$, then $A \equiv C$.

4) A valid congruence relationship is obtained if the number, the residue, and the modulus are divided by a common factor.

5) A valid congruence relationship is obtained if the number and the residue are divided by some common factor relatively prime to the modulus.

The material of this section has presented briefly, without proof, the pertinent concepts of congruences. Additional material on the subject may be found in any standard text on number theory.[2]

Development of the Residue Code

A residue code associated with a particular natural number is formed from the least positive residues of the particular number with respect to different bases. The first requirement for an efficient residue number system is that the bases of the different digits of the representation must be relatively prime. If a pair of bases are not relatively prime, the effect is the introduction of redundancy. The following example will illustrate this fact. Contrast the residues associated with bases of magnitude 2 and 6 against the residues associated with bases of magnitude 3 and 4. In the first case, the bases are not relatively prime while in the second case the bases are relatively prime. The residues associated with the bases of magnitude 2 and 6 are unique for only 6 states while the residues associated with the bases of magnitude 3 and 4 provide a unique residue representation for 12 states. This is further clarified by Table I.

An example of a residue number system is presented in Table II. The number system shown in Table II uses the prime bases 2, 3, 5, and 7. The number system, therefore, contains 210 states. The 210 states may correspond to the positive integers 0 to 209. Table II shows the residue number representation corresponding to the positive integers 0 to 29. Additional integers of the number system may be found by congruence operations. Let a, b, c, and d be the digits associated with the bases 2, 3, 5, and 7, respectively. The following congruences define a, b, c, and d for the residue representation of the number N:

$$N \equiv a \bmod 2$$

$$N \equiv b \bmod 3$$

$$N \equiv c \bmod 5$$

$$N \equiv d \bmod 7.$$

The residue number system is readily extended to include more states. For example, if a base 11 is added to the representation, it is then possible to represent 2310 states. Table III shows the product and sum of the first nine consecutive primes greater than or equal to 2.

[2] G. H. Hardy and E. M. Wright, "An Introduction to the Theory of Numbers," Oxford University Press, London, Eng.; 1956.

TABLE I

Redundancy of a Nonrelatively Primed Base Representation

	Least Postive Residue			
Number	Mod 2	Mod 6	Mod 3	Mod 4
0	0	0	0	0
1	1	1	1	1
2	0	2	2	2
3	1	3	0	3
4	0	4	1	0
5	1	5	2	1
6	0	0	0	2
7	1	1	1	3
8	0	2	2	0
9	1	3	0	1
10	0	4	1	2
11	1	5	2	3
12	0	0	0	0
13	1	1	1	1
14	0	2	2	2

TABLE II

Natural Numbers and Corresponding Residue Numbers

Natural Numbers	2357	Natural Numbers	2357	Natural Numbers	2357
0	0000	10	0103	20	0206
1	1111	11	1214	21	1010
2	0222	12	0025	22	0121
3	1033	13	1136	23	1232
4	0144	14	0240	24	0043
5	1205	15	1001	25	1104
6	0016	16	0112	26	0215
7	1120	17	1223	27	1026
8	0231	18	0034	28	0130
9	1042	19	1145	29	1241

TABLE III

Number of States and Digits Associated with a Residue Representation

i	p_i	$\sum_{i=1}^{n} p_i$	$\prod_{i=1}^{n} p_i$	p_i bits	$\sum p_i$ bits
1	2	2	2	1	1
2	3	5	6	2	3
3	5	10	30	3	6
4	7	17	210	3	9
5	11	28	2,310	4	13
6	13	41	30,030	4	17
7	17	58	510,510	5	22
8	19	77	9,699,690	5	27
9	23	100	223,092,670	5	32

The product of the primes indicates the number of states of the number system, while the sum of the primes is a measure of the size of the representation in terms of digits. Table III also includes the number of bits required to represent each prime base in the binary number system.

Residue Addition and Multiplication

The residue number representation consists of several digits and is assumed to be in one-to-one correspondence with some positive integers of the real number system. The digits of the residue representation are the least positive residues of these real positive integers with respect to the different moduli which form the bases of the residue representation. It follows as a direct consequence of the structure of the residue number system and the properties of linear congruences that operations of addition and multiplication are valid in the residue number system subject to one proviso: the residue system must possess a number of states sufficient to represent the generated sum or product. If the residue number system does not have a sufficient number of states to represent the sums and the products generated by a particular finite set of real integers, then the residue system will overflow and more than one sum or product of the real number system may correspond to one residue representation. For a residue number with a sufficient number of states, an isomorphic relation exists with respect to the operations of addition and multiplication in the residue system and a finite system of real positive integers.

Each digit of the residue number system is obtained with respect to a different base or modulus. It follows, therefore, that the rules of arithmetic associated with each digit will be different. For example, the addition and multiplication of the digits associated with moduli 2 and 3 follow rules specified in Table IV. No carry tables are necessary since the residue number system does not have a carry mechanism. Addition of two residue representations is effected by the modulo addition of corresponding digits of the two representations. Corresponding digits must have the same base or modulus. Modulo addition of digits which have different bases is not defined. Multiplication in the residue system is effected by obtaining the modulo product of corresponding digits. The operations of addition and multiplication of two residue numbers are indicated by the following notation:

$$S = A \oplus B$$

$$p = A \odot B.$$

Consider a residue number representation with bases 2, 3, 5, and 7. We assume an isomorphic relation between the residue number system and the real positive numbers 0 to 209. An isomorphic relation then exists for the operations of multiplication and addition only if the product or sum is less than 210. The following examples

TABLE IV

Mod 2 and Mod 3 Sums and Products

⊕ | 0 1
0 | 0 1
1 | 1 0

sum mod 2

⊙ | 0 1
0 | 0 0
1 | 0 1

product mod 2

⊕ | 0 1 2
0 | 0 1 2
1 | 1 2 0
2 | 2 0 1

sum mod 3

⊙ | 0 1 2
0 | 0 0 0
1 | 0 1 2
2 | 0 2 1

product mod 3

employing residue numbers illustrate the addition and multiplication operations and the presence of an isomorphism or the lack of isomorphism in the case of overflow. Residue numbers will be distinguished by the use of parentheses.

$$29 + 27 = S = 56$$

$$29 \leftrightarrow (1\ 2\ 4\ 1)$$

$$27 \leftrightarrow (1\ 0\ 2\ 6)$$

$$56 \leftrightarrow (0\ 2\ 1\ 0)$$

$$\begin{array}{rl} & (1\ 2\ 4\ 1) \\ \oplus & (1\ 0\ 2\ 6) \\ \hline & (0\ 2\ 1\ 0) \end{array}.$$

The following operations are considered in performing the addition of the two residue representations:

$$1 + 1 \equiv 0 \bmod 2$$

$$2 + 0 \equiv 2 \bmod 3$$

$$4 + 2 \equiv 1 \bmod 5$$

$$1 + 6 \equiv 0 \bmod 7.$$

Consider the addition of two numbers which produce a sum greater than 209.

$$\begin{array}{rl} S = 100 + 200 & \\ & (0\ 1\ 0\ 2) \\ \oplus & (0\ 2\ 0\ 4) \\ \hline & (0\ 0\ 0\ 6) \end{array}.$$

The residue representation (0 0 0 6) corresponds to the real positive number 90. In this particular example, the sum has overflowed the residue representation. The resulting sum is the correct sum modulo 210.

$$300 \equiv 90 \text{ modulo } 210.$$

Finite real number systems and residue number systems have the same overflow characteristics. The sum which remains after the overflow is the correct sum with respect to a modulus numerically equal to the number of states in the finite number system.

The following is presented as an example of the process of residue multiplication:

$$\begin{aligned} p &= 10 \times 17 = 170 \\ 10 &\leftrightarrow (0\ 1\ 0\ 3) \\ 17 &\leftrightarrow (1\ 2\ 2\ 3) \\ 170 &\leftrightarrow (0\ 2\ 0\ 2) \end{aligned} \qquad \begin{array}{r} (0\ 1\ 0\ 3) \\ \odot\ (1\ 2\ 2\ 3) \\ \hline (0\ 2\ 0\ 2) \end{array}.$$

The process of multiplication involved consideration of the following relations for each digit:

$$1 \times 0 \equiv 0 \bmod 2$$
$$1 \times 2 \equiv 2 \bmod 3$$
$$0 \times 2 \equiv 0 \bmod 5$$
$$3 \times 3 \equiv 2 \bmod 7.$$

An overflow resulting from a multiplication is no different from the overflow resulting from an addition. Consider the product obtained from the residue multiplication of the numbers 10 and 100. The result in the modulo 210 number system is 160, since

$$1000 \equiv 160 \bmod 210.$$

Subtraction and the Representation of Negative Numbers

The process of subtraction is obtainable in the residue number system by employing a complement representation consisting of the additive inverses of the positive residue representation. The additive inverse always exists, since each of the elements of the residue representation is an element of a field. There is no basic problem associated with the subtraction operation. There is, however, a problem associated with the representation of negative numbers. In particular, some mechanism must be included in the number system which will permit the representation of positive and negative numbers. This problem is discussed here and in the following section.

The additive inverse of a residue number is defined by the following:

$$a \oplus a' = 0.$$

The formula may be considered to apply to a digit of the residue system or equally well to the whole residue representation. Consider the following examples with reference to the modulo 210 residue number system:

$$a = (1\ 2\ 4\ 1)$$

then

$$a' = (1\ 1\ 1\ 6),$$

since

$$\begin{array}{r} (1\ 2\ 4\ 1) \\ \oplus\ (1\ 1\ 1\ 6) \\ \hline (0\ 0\ 0\ 0) \end{array}.$$

The following examples have been chosen to illustrate the subtraction process and to some extent the difficulties associated with the sign of the difference:

$$D = A \ominus B = A \oplus B'.$$

We consider first the case where the magnitude of A is greater than B.

$$\text{Let } A = 200 \qquad B = 100.$$

In residue representation,

$$B' = (0\ 2\ 0\ 5)$$

and

$$\begin{array}{r} (0\ 2\ 0\ 4) \\ \oplus\ (0\ 2\ 0\ 5) \\ \hline (0\ 1\ 0\ 2) \end{array}.$$

The residue representation of the difference corresponds to positive 100 in the real number domain. We consider next the case where the magnitude of B is greater than the magnitude of A.

$$A' = (0\ 1\ 0\ 3)$$

then

$$D = A' \oplus B$$

and

$$\begin{array}{r} (0\ 1\ 0\ 3) \\ \oplus\ (0\ 1\ 0\ 2) \\ \hline (0\ 2\ 0\ 5) \end{array}.$$

The difference (0 2 0 5) is the additive inverse of (0 1 0 2). Unless additional information is supplied, the correct interpretation of the representation (0 2 0 5) is in doubt. (0 2 0 5) may correspond to either +110 or −100.

The difficulties associated with whether a residue representation corresponds to a positive or negative integer can be partially removed by the division of the residue number range into two parts. This is exactly the scheme that is employed to obtain a machine representation of positive and negative natural numbers. For the system of natural numbers, two different machine representations of the negative numbers may be obtained and are commonly designated the radix complement representation of negative numbers and the diminished radix complement representation of negative numbers.

The complement representation for a residue code is defined in terms of the additive inverse. Thus, the representation of negative A is A' where $A \oplus A' = 0$, and the range of A is restricted to approximately one half of the total possible range of the residue representation. This can be illustrated by consideration of a specific residue code. This residue representation employing bases of magnitude 2, 3, 5, and 7, is divided into two parts. The residue representations corresponding

to the natural numbers 0 to 104 are considered positive. The residue representations corresponding to the natural numbers 105 to 209 are considered inverse representations and associated with the negative integers from -1 to -105. The range of this particular number system is from -105 to $+104$. The arithmetic rules pertaining to sign and overflow conventions for this particular number system are the same rules normally associated with radix complement arithmetic.

The complement representation does eliminate in principle any ambiguity concerning the sign of the result of an arithmetic operation. However, there is a practical difficulty. The determination of the sign associated with a particular residue representation requires the establishment of the magnitude of the representation relative to the magnitude which separates the positive and negative representations. The determination of relative magnitude for a residue representation is discussed in the next section, where it is shown that the determination of relative magnitude is not a simple problem.

Conversion from a Residue Code to a Normal Number Representation

It is frequently desirable to determine the natural number associated with a particular residue representation. The need for this conversion occurs frequently in investigation of the properties of the residue system. The residue representation is constructed in such a manner that magnitude is not readily obtainable. The presence of digit weights in the normal polynomial type number representation greatly facilitates the determination of magnitude. However, it is possible to assign a weight to each digit of the residue representation in such a manner that the modulo m sum of the digit-weight products is the real natural number in a consistently weighted representation. m is the product of all the bases employed in the residue representation. The conversion technique is known as the "Chinese Remainder Theorem." The material which follows describes the conversion technique but omits the proof. A simple and straightforward proof is found in Dickson.[3] The proof does not refer specifically to residue number systems, but rather to a system of linear congruences. If so regarded, a system of congruences defines a component of a residue number system.

Consider a residue number system with bases $m \cdots m_t$. The corresponding digits are labeled $a_1 \cdots a_t$. The following equations define the conversion process:

$$a_1 A_1 \frac{m}{m_1} + \cdots + a_t A_t \frac{m}{m_t} \equiv S \bmod m$$

where

$$A_i \frac{m}{m_i} \equiv 1 \bmod m_i$$

and

$$m = \prod_{j=1}^{t} m_j.$$

The conversion formula for a particular residue number system is now obtained.

$m_1 = 2 \quad m_2 = 3 \quad m_3 = 5 \quad m_4 = 7$

$105\ A_1 \equiv 1 \bmod 2 \quad$ so $A_1 = 1$

$70\ A_2 \equiv 1 \bmod 3 \quad$ so $A_2 = 1$

$42\ A_3 \equiv 1 \bmod 5$

$2\ A_3 \equiv 1 \bmod 5 \quad$ so $A_3 = 3$

$30\ A_4 \equiv 1 \bmod 7$

$2\ A_4 \equiv 1 \bmod 7 \quad$ so $A_4 = 4$

$$105\ a_1 + 70\ a_2 + 126\ a_3 + 120\ a_4 \equiv S \bmod 210.$$

The conversion formula is now used to determine the natural number corresponding to the residue representation (1 2 0 4).

$$105\ (1) + 70\ (2) + 126\ (0) + 120\ (4) = 725$$

$$725 \equiv S \bmod 210$$

$$S = 95.$$

The conversion process described above requires conventional multiplication and modulo addition.

Other conversion techniques exist. In particular it is possible by means of a deductive process to determine the magnitude of a particular residue representation. This requires both a knowledge of the nature of the residue system and the natural number representation associated with at least one residue representation.

Due to the deductive nature of the process, it is more suitable for human computation than for machine computation. The process is explained using the residue number of the previous example (1 2 0 4). The knowledge of the residue representation for unity which is (1 1 1 1) is assumed. Consider the effect of changing the second digit from one to two. The change adds the product $m_1 m_3 m_4 = 70$ to the number, since 70 is congruent 1, modulo 3. The resulting residue representation (1 2 1 1) corresponds to 71. The effect of changing the third digit is to change the magnitude by some multiple of the product $m_1 m_2 m_4 = 42$. The correct change in magnitude is $42x$ where $42x \equiv 4 \bmod 5$; so $42x = 84$ and the residue representation (1 2 0 1) corresponds to 155. The fourth digit is modified by the addition of a three. The effect of this change is determined by $30x \equiv 3 \bmod 7$. The magnitude change is 150. The sum of 150 and 155 modulo 210 yields the correct result 95, in correspondence with (1 2 0 4).

[3] L. E. Dickson, "Modern Elementary Theory of Numbers," University of Chicago Press, Chicago, Ill., p. 16; 1939.

Sign determination for the residue code is dependent on the determination of a greater than or less than relationship. A possible method might involve the conversion techniques described previously. Such a scheme would involve the standard comparison techniques associated with the determination of the relative magnitude of two numbers represented in a weighted representation. An alternate conversion procedure yields a conversion from the residue code to a nonconsistently based polynomial number representation by means of residue arithmetic. Consider a residue code consisting of t digits. The t digits of the residue code are associated with t congruence relationships as follows:

$$S \equiv a_i \bmod m_i \qquad 1 \leq i \leq t.$$

S is the magnitude of the number expressed in normal representation. It is also possible to express the number S as

$$S = a_i + A_i m_i.$$

A_i is the integer part of the quotient of S divided by m_i. In regard to a greater or less than relationship, the determination of A_i divides the range of the residue representation into m/m_i parts. We proceed to calculate A_i from the set of t equations given above. Let

$$S = a_t + A_t m_t \qquad A_t < \frac{m}{m_t}.$$

This equation is then used to replace S in the remaining $t-1$ equations, yielding $t-1$ equations of the form

$$A_t m_t \equiv (a_i + a_t') \bmod m_i \qquad 1 \leq i \leq t-1$$

or

$$A_t \equiv (a_i + a_t')/m_t{}^i \bmod m_i$$
$$A_t \equiv d_t{}^i \bmod m_i$$

where $/m_t{}^i$ is the multiplicative inverse of m_t with respect to base m_i. The multiplicative inverse is defined as[4]

$$x_t/x_t{}^i \equiv 1 \bmod m_i.$$

$d_t{}^i$ is the least positive residue of $(a_i+a_t')/m_t$ with respect to base i.

a_t' is the additive inverse of a_t.

Let A_t be expressed as

$$A_t = d_t{}^{t-1} + A_{t-1}m_{t-1} \qquad A_{t-1} < \frac{m}{m_t m_{t-1}}.$$

If this expression is substituted for A_t a set of $t-2$ equations remain. The equations are of the form

$$A_{t-1} \equiv [d_t{}^i + (d_t{}^{t-1})']/m_{t-1} \bmod m_i \qquad 1 \leq i \leq t-2$$
$$A_{t-1} \equiv d^i{}_{t-1} \bmod m_i.$$

[4] The existence of the multiplicative inverse requires that x_t and m_i be relatively prime.

The system of equations shown below is generated by repetition of the above substitution process until no equations remain.

$$S = a_t + A_t m_t$$
$$A_t = d_t^{t-1} + A_{t-1}m_{t-1}$$
$$A_{t-1} = d_{t-1}^{t-2} + A_{t-2}m_{t-2}$$
$$\vdots$$
$$A_3 = d_3{}^2 + A_2 m_2$$
$$A_2 \equiv d_2{}^1 \bmod m_1$$

The equations are combined to yield

$$S = a_t + m_t\{d_t^{t-1} + m_{t-1}[d_{t-1}^{t-2} + m_{t-2}(d_{t-2}^{t-3} + \cdots$$
$$= a_t + m_t d_t^{t-1} + m_t m_{t-1} d_{t-1}^{t-2} + m_t m_{t-1} m_{t-2} d_{t-2}^{t-3} + \cdots$$
$$+ \frac{m}{m_1} d_2{}^1$$

where

$$A_t < m_t$$
$$d_{t-n}^{t-n-1} < m_{t-n-1}.$$

Therefore, S is never equal to or greater than m and $d_2{}^1$ divides the range into m_1 parts, $d_3{}^2$ divides each of the m_1 parts into m_2 parts, $d_4{}^3$ divides each of the m_2 parts into m_3 parts, etc.

The determination of the less than or greater than relationship consists of the successive comparison of the d_{t-n}^{t-n-1} constants corresponding to two residue representations. Let the representations be designated E and F. The first step of the greater than or less than determination is the comparison of $d_2{}^1(E)$ and $d_2{}^1(F)$. If the two constants are different the process may be terminated and the larger number is associated with the larger constant. If the constants are equal in value, the comparison process must consider the pair of constants $d_3{}^2(E)$ and $d_3{}^2(F)$. The process is continued in this manner until a set of nonidentical constants is found. If all of the d constants are identical, a final comparison is made on the basis of the pair of tth digits of the two residue representations.

The formulas which define the greater than, less than process may be applied recursively to obtain a formula for a greater number of digits. The process has been extended to five variables and the results are shown in Fig. 1.

Admittedly, the process required to obtain a greater than or less than relationship leaves much to be desired. One presumed advantage of the residue number system was the absence of a carry process. The greater or less than process is essentially sequential and is in many ways similar to the carry process of ordinary arithmetic. The ultimate usefulness of the residue code for general-purpose computation appears very much dependent on the development of simple techniques for the determina-

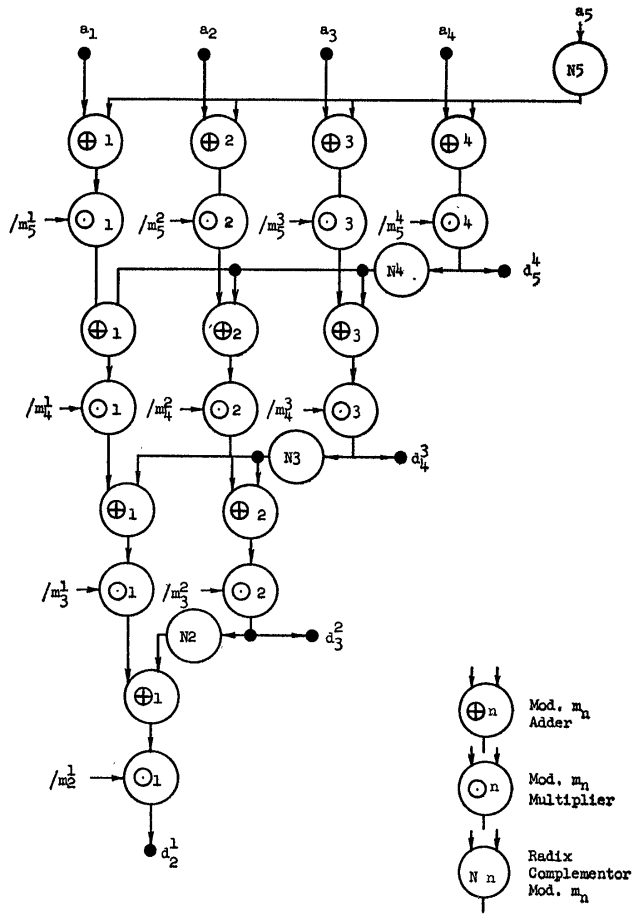

Fig. 1—Logic for the determination of the greater or less than relationship.

tion of the relative magnitude of two residue code digits.

Division

The division process for residue codes is complicated by two factors. The first is the absence of a multiplicative inverse for the zero element. The second difficulty is the fact that residue division and the normal division process are in one to one correspondence only when the resulting quotient is an integer value. We shall consider first the problem of residue division of the elements of a single field and shall consider later the elements of several fields considered as a residue code. The division process represented in equation form as

$$\frac{a}{b} = q$$

implies

$$a = bq.$$

The difference between normal arithmetic and residue arithmetic is that in residue arithmetic the product bq need not necessarily be equal to a; only the congruence of a and bq is required.

$$bq \equiv a \bmod m_n$$

Multiplication by the multiplicative inverse of b designated $/b$ obtains

$$q \equiv a/b \bmod m_n.$$

The correct interpretation of q in the above equation is that the number a is obtained by forming the modulo sum consisting of b, q representations. The sum is carried out in a closed and finite modulo number system of base m_n. Thus, q corresponds to the quotient only when the quotient has an integer value. Examples may be obtained from the consideration of a modulo 5 number system:

$$\frac{4}{2} = q$$

$$2q \equiv 4 \bmod 5$$

$$q \equiv 2 \bmod 5$$

$$\frac{4}{3} = q$$

$$3q \equiv 4 \bmod 5$$

$$q \equiv 3 \bmod 5$$

$$\text{note } 3 \times 3 \equiv 4 \bmod 5$$

$$\frac{3}{4} = q$$

$$4q \equiv 3 \bmod 5.$$

$$q \equiv 2 \bmod 5.$$

In the above examples, q corresponds to the quotient only in the first example.

The residue code representation of a number consists of many digits, $A = (a_1, a_2, \cdots, a_n)$. Each digit of the representation is associated with a different prime base. The number system is a modulo m system where

$$m = \prod_{i=1}^{n} m_i.$$

The division of two numbers in residue code may be expressed by a system of congruences. The solution $Q = (q_1, q_2, \cdots, q_n)$ must satisfy all the congruence relationships of the system. A zero digit in the divisor $B = (b_1, b_2, \cdots, b_n)$ means that B and m are not relatively prime hence the multiplicative inverse of B does not exist.

$$QB \not\equiv A \bmod m.$$

For the special case in which $b_i = 0$ and $a_1 = 0$, a valid congruence relationship of the form

$$\frac{QB}{m_i} \equiv \frac{A}{m_i} \bmod \frac{m}{m_i}$$

is obtainable.

The process of residue division has certain interesting properties and quite possibly has applications in respect to special problems. Unfortunately, the residue division process is not a substitute for normal division. It appears that the only way in which division can be effected in the residue code is by the utilization of techniques similar to those employed for division in a consistently weighted number system. The division process then requires trial and error subtraction or addition and the greater than or less than relationship. The division algorithm could also include trial multiplication, since in the residue system addition and multiplication require the same period of time.

Conclusions

The material of this paper forms a preliminary investigation of the applicability of residue number systems to the arithmetic operations of digital computers. The residue system has been found attractive in terms of the operations of multiplication and addition. It is possible to realize practical logical circuitry to yield the product in the same operation time as for the sum, since the product is not obtained by the usual procedure of repetitive addition. The main disadvantages of the residue number system are associated with the necessity of determining absolute magnitude. Thus, the division process, the detection of an overflow, and the determination of the correct sign of a subtraction operation are processes which at this stage of the investigation seem to involve considerable complexity. Nevertheless, many special-purpose applications are certainly well-suited to the residue code. In particular, there exists a class of control problems characterized by the absence of the need for division and the existence of a well-defined range for the variables, and also by the fact that the sign of the variables is known. For the problems of this class, the use of the residue code should result in a reduction of the over-all computation period and should yield a computer with a higher bandwidth than obtainable with the conventional number system.

The ultimate usefulness of the residue code will probably be determined largely by the success of the circuit designer in perfecting circuitry ideally suited for residue code operations.

The material of this paper is essentially Chapter 5 of the author's doctoral dissertation.[5] At the time of the completion of the dissertation, the author was unaware of the work of M. Valach[6] and A. Svoboda[7,8] in Czechoslovakia. Additional literature[6–8] was obtained from recent visitors to the Soviet Union. The author wishes to take this opportunity to acknowledge the work of Valach and Svoboda.

[5] H. L. Garner, "Error Checking and the Structure of Binary Addition," Ph.D. dissertation, University of Michigan, Ann Arbor, pp. 105–140; 1958.

[6] M. Valach, "Vznik kodu a ciselne soustavy zbytkovych trid," *Stroje Na Zpracovani Informaci*, Sbornik III; 1955.

[7] A. Svoboda and M. Valach, "Operatorove obvody," *Stroje Na Zpracovani Informaci*, Sbornik III; 1955.

[8] A. Svoboda, "Rational numerical system of residual classes," *Stroje Na Zpracovani Informaci*, Sbornik V; 1957.

System Evaluation and Instrumentation for Military Special-Purpose Digital Computer Systems

A. J. STRASSMAN† and L. H. KURKJIAN†

Introduction

TESTING and instrumentation are essential prerequisites for the completion and operation of any new system. A system can be defined as a number of components that are amalgamated or integrated together to perform a desired operation. Throughout this paper a "component" is considered to be a complete functional part of a data processing system such as an arithmetic unit or a buffer. To ascertain if a component in the system is going to perform correctly its specific function, it is sometimes necessary for the implementation of tests to be more complex than the component undergoing the testing. This becomes apparent when the component is a part of a large system and has many inputs and outputs.

To prove the system feasibility or operation of the components it is necessary to do either of two things: 1) duplicate and maintain an entire system and use it as one master test fixture to evaluate each functional component; or 2) provide individual test facilities for the evaluation of each of the functional components. The second approach requires the design of simulation equipment to provide the necessary inputs (control signals and data) to check out completely the operation of each

† Hughes Aircraft Co., Fullerton, Calif.