References

[1] J. R. Duley, "DDL—a digital system design language," Ph.D. dissertation, University of Wisconsin, Madison, June 1967.

[2] J. R. Duley and D. L. Dietmeyer, "A digital system design language (DDL)," *IEEE Trans. Computers*, vol. C-17, pp. 850–861, September 1968.

[3] J. Feldman and D. Gries, "Translator writing systems," *Commun. ACM*, vol. 11, pp. 77–113, February 1968.

[4] D. F. Gorman and J. P. Anderson, "A logic design translator," *1962 Fall Joint Computer Conf., AFIPS Proc.*, vol. 22. Washington, D. C.: Spartan, 1962, pp. 251–261.

[5] R. M. Proctor, "A logic design translator experiment demonstrating relationships of language to systems and logic design," *IEEE Trans. Electronic Computers*, vol. E3-13, pp. 422–430, August 1964.

[6] H. Schorr, "Computer-aided digital systems design and analysis using a register transfer language," *IEEE Trans. Electronic Computers*, vol. EC-13, pp. 730–737, December 1964.

[7] J. P. Roth, "Systematic design of automata," *1965 Fall Joint Computer Conf., AFIPS Proc.*, vol. 27, pt. 1. Washington, D. C.: Spartan, 1965, pp. 1093–1100.

[8] R. E. Miller, *Switching Theory*, vol. 2. New York: Wiley, 1965, pp. 99–191.

[9] R. W. Ryniker, II. "Machine translation of a digital design language," M.S. thesis, University of Wisconsin, Madison, Wis., 1967.

[10] D. L. Dietmeyer and P. R. Schneider, "Identification of symmetry, redundancy and equivalence of Boolean functions," *IEEE Trans. Electronic Computers*, vol. EC-16, pp. 804–817, December 1967.

Sign Detection in Residue Number Systems

D. K. BANERJI, STUDENT MEMBER, IEEE, AND J. A. BRZOZOWSKI, MEMBER, IEEE

Abstract—This paper is concerned with the sign detection problem in residue number systems. The proposed solution is applicable only to nonredundant systems. It is shown that under rather general conditions an explicit, closed formula for the sign function can be obtained. In a special case, when one of the moduli is 2, the sign function becomes an EXCLUSIVE-OR function. A sign detection algorithm is proposed and methods of implementing the algorithm are presented.

Index Terms—Algorithm for sign detection, residue number system, sign function.

Introduction

Residue number systems have been of interest to mathematicians for a very long time. However, the use of the system to carry out machine computation has attracted attention quite recently [1], [2], [4], [5], [7]–[9]. The most desirable feature of the residue number system is that in the operations of addition, subtraction, and multiplication, any digit of the result is determined solely by the corresponding digits of the operands. This results in the elimination of carries from one residue position to another. However, one of the drawbacks of the system is the fact that the algebraic sign of any number in an arbitrary residue code is a function of all the residue digits. This makes the sign detection process complicated, slow, and expensive.

It is the purpose of this paper to investigate the sign detection problem, which deserves special attention because it is also closely related to the problems of relative magnitude comparison and overflow detection.

Manuscript received June 17, 1968; revised December 31, 1968. This work was supported in part by the National Research Council of Canada under Grant A-1617. Most of the work was carried out while the authors were with the Department of Electrical Engineering, University of Ottawa, Ottawa, Canada.

The authors are with the Department of Applied Analysis and Computer Science, University of Waterloo, Waterloo, Canada.

Residue Codes

Residue codes and their properties have been widely discussed in literature [1], [5]–[8], and for this reason they will not be dealt with in detail here. Only the essentials will be briefly reviewed.

Definition 1: Let $\mathfrak{M} = \{m_1, m_2, \cdots, m_n\}$ be an ordered set of positive integers, where $m_i \geq 2$ for $i = 1, 2, \cdots, n$. The m_i are called "moduli" or "radices," and the corresponding ordered set $(x_1, x_2, \cdots, x_n)$ of least positive residues of a natural number X, with respect to the moduli, forms the residue representation or code for that number, where the least positive residue of X with respect to m_i is denoted by $|X|_{m_i} = x_i$. For example, if $\mathfrak{M} = \{2, 3, 5\}$ and $X = 14$, then $|14|_2 = 0$, $|14|_3 = 2$, and $|14|_5 = 4$. Thus 14 is represented by (0, 2, 4) in this system.

In order to avoid redundancy (unless redundancy is desirable), the moduli of a residue number system must be pair-wise relatively prime i.e., the greatest common divisor of each pair of moduli must be unity. If this is so, the number of integers that can be coded uniquely in a system with moduli $\{m_1, m_2, \cdots, m_n\}$ equals the product $m_1 m_2 \cdots m_n$. This is a direct consequence of the Chinese Remainder Theorem [6]. In the case $\mathfrak{M} = \{2, 3, 5\}$, therefore, a total of 30 integers can be coded uniquely. These can correspond to the natural numbers 0 through 29.

The most convenient way of representing negative integers is as follows. The residue number range is divided into two parts. One part is assigned to positive integers, and the other to negative integers. The negative integers are then represented in radix complement form, defined in terms of additive inverse. Thus $-X$ is represented by X' ,where X' is the additive inverse of X, and is defined as follows. If $X=(x_1, x_2, \cdots, x_n)$, then $X'=(x_1', x_2', \cdots, x_n')$ where $x_i'=m_i-x_i$ for $i=1, 2, \cdots, n$. Thus for $\mathfrak{M}=\{2, 3, 5\}$, -14 is represented by (0, 1, 1).

Sign Determination

The problem of sign determination is a major problem encountered in using residue arithmetic for computation. Attaching a sign bit to a residue number is of little help because the magnitude of a residue number is not readily available, and therefore, after adding a positive and a negative number, for instance, the sign of the result is not immediately known. We have seen earlier how the whole range of representation in a residue system is divided into two parts to represent positive and negative integers. One obvious way, therefore, is to convert a given residue number to its natural number form which will fall either in the positive or the negative region of the representation. However, this is not an attractive solution for the problem because it is slow, and therefore offsets the advantage of speed in a residue computer.

We consider a sign function S which has a value 0 for positive integers and 1 for negative integers.

In considering the sign detection problem, it could be expected that to get one bit of information (sign) all the residue information is not required. Szabo [8] has proved that such a scheme is impossible and in the general case no reduction of information from any residue digit is possible without loss of sign information.

In his coding theorem, Szabo has proved that all the information from a residue digit must be used in any sign determination process, provided the modulus m_p of the digit is smaller than $\sqrt{M}$, where

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

In the corollary of his coding theorem, Szabo proved that sign detection is impossible if the pth residue digit is coded into less than $\hat{m}_p$ states, where

$$\hat{m}_p = \frac{M}{m_p} \cdot$$

The corollary yields a positive result. It shows that it is possible to reduce the information from a residue digit but only within a certain limit. This limit is fixed by the modulus whose information is to be reduced and by the other moduli of the system. For example, if we consider a system with moduli m_1 and m_2 with $m_1<m_2$, then m_1 is the lower limit on the reduction of information from m_2 since

$$\hat{m}_2 = \frac{M}{m_2} = \frac{m_1 m_2}{m_2} = m_1,$$

and the corollary states that we can not reduce the information from m_2 to less than $\hat{m}_2=m_1$ states. The sign detection approach presented in this paper is within the limitations imposed by the coding theorem.

The Two-Moduli Case

We shall first consider the case of two moduli.

Theorem 1: Given two moduli $2M$ and N, N odd and $N>2M$, let $l=|N|_{2M}$ be the least positive residue of N modulo $2M$. Then the sign of a number X, $0\leq X\leq 2MN-1$, is established by a proposition P such that

$$S = 0 \text{ iff } P = \bigvee_{i=0}^{M-1} (|X|_{2M} = |il + |X|_N|_{2M}) \text{ is true.}$$

The symbol "$\vee$" denotes a logical OR.

Proof: Since the moduli are relatively prime, they can represent $2MN$ integers, 0 to $2MN-1$, uniquely. Without any loss of generality, integers in the range 0 to $MN-1$ will be considered to be positive and those in the range MN to $2MN-1$ to be negative.

Consider an integer X, $0\leq X\leq 2MN-1$. We can write X as

$$X = KN + |X|_N.$$

By definition

$$l = |N|_{2M}.$$

Hence

$$N = 2ML + l \qquad \text{for some positive integer } L.$$

Therefore

$$X = 2MKL + Kl + |X|_N$$

or

$$|X|_{2M} = |Kl + |X|_N|_{2M} \text{ for any } X, 0 \leq X \leq 2MN - 1.$$

It is easily verified that if X is positive then K is bounded between 0 and $M-1$, and if X is negative then K lies between M and $2M-1$. Now we will show that if X is positive and if X satisfies the relation $|X|_{2M} = |K'l+|X|_N|_{2M}$ for some K', then $0\leq K'\leq M-1$. For let us suppose

$$|X|_{2M} = |Kl + |X|_N|_{2M} = |K'l + |X|_N|_{2M},$$
$$0 \leq K \leq M-1, \qquad M \leq K' \leq 2M-1.$$

Choose Y such that $Y=K'N+|X|_N$. Clearly, $Y\geq MN+|X|_N>MN-1$. Also $Y\leq 2MN-N+|X|_N\leq 2MN-1$. Hence Y lies in the negative region and can be expressed as $Y=2MK'L+K'l+|X|_N$. Hence, $|Y|_{2M}=|K'l+|X|_N|_{2M}=|X|_{2M}$. Also $Y=K'N+|X|_N$ implies $|Y|_N=|X|_N$. This means that with respect to moduli $2M$ and N, two different numbers have the same residue code. This is impossible since $2M$ and N form a nonredundant residue system. It is also clear that for any X there is exactly one value of K in the range $0\leq K$

$\leq 2M-1$ satisfying $|X|_{2M} = |Kl + |X|_N|_{2M}$. Therefore

$$S = 0 \text{ iff}$$
$$(|X|_{2M} = ||X|_N|_{2M}) \vee (|X|_{2M} = ||X|_N + l|_{2M}) \cdots \vee$$
$$(|X|_{2M} = ||X|_N + (M-1)l|_{2M})$$

is true or

$$P = \bigvee_{i=0}^{M-1} (|X|_{2M} = |il + |X|_N|_{2M})$$

is true, and

$$S = 1 \text{ iff proposition } P' = \bigvee_{j=M}^{2M-1} (|X|_{2M} = |jl + |X|_N|_{2M})$$

is true or P is false.

Corollary 1: If $M=1$ then X is positive iff $|X|_2 = ||X|_N|_2$ and negative iff $|X|_2 = ||X|_N + 1|_2 = |||X|_N|_2 + 1|_2$.

If binary coding is used for the residues, $||X|_N|_2$ is the least significant bit of $|X|_N$ and $|X|_2$ is either 0 or 1. Therefore, X is positive or negative, depending on whether the EXCLUSIVE-OR sum of $|X|_2$ and $||X|_N|_2$ is 0 or 1, respectively. The sign detection process in this case can be schematically represented as in Fig. 1.

The following example illustrates the use of Theorem 1.

Example 2: Consider a system with $2M=4$ and $N=5$. Then $l = |5|_4 = 1$. This system can represent 20 integers uniquely, as shown in Table I.

In using Theorem 1 here, i varies from 0 to 1. Therefore, a residue number $(|X|_4, |X|_5)$ is positive iff $||X|_4 - |X|_5|_4 = 0$ or 1 and it is negative iff $||X|_4 - |X|_5|_4 = 2$ or 3. This is confirmed from Table I.

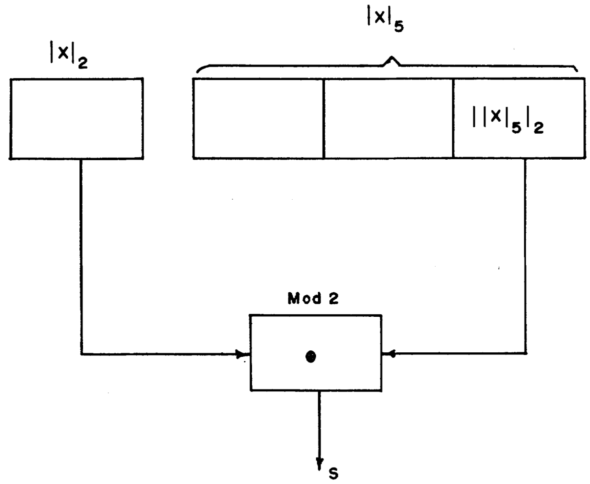

Fig. 1. Sign detection scheme for $M=1$, $N=5$.

TABLE I

TABLE FOR EXAMPLE 2

Integers X	$\lvert X_4\rvert$	$\lvert X\rvert_5$	$\lvert\lvert X\rvert_4 - \lvert X\rvert_5\rvert_4$	S
0	0	0	0	0
1	1	1	0	0
2	2	2	0	0
3	3	3	0	0
4	0	4	0	0
5	1	0	1	0
6	2	1	1	0
7	3	2	1	0
8	0	3	1	0
9	1	4	1	0
−10, 10	2	0	2	1
−9, 11	3	1	2	1
−8, 12	0	2	2	1
−7, 13	1	3	2	1
−6, 14	2	4	2	1
−5, 15	3	0	3	1
−4, 16	0	1	3	1
−3, 17	1	2	3	1
−2, 18	2	3	3	1
−1, 19	3	4	3	1

Sign Detection: A General Approach

The problem of sign detection becomes more complicated when the number of moduli is large. The basic philosophy still remains unchanged and we shall make use of Theorem 1 for determining sign.

Let us consider a system of n mutually prime moduli $m_1, m_2, \cdots, m_n$. There is not much loss of generality if we assume one of the moduli to be even. Let $m_1 = 2m$. It is obvious that all other moduli must be odd. We partition the set of moduli into two subsets $\{m_1, \cdots, m_j\}$ and $\{m_{j+1}, \cdots, m_n\}$ where

$$\prod_{i=1}^{j} m_i = 2m\left(\prod_{i=2}^{j} m_i\right) < \prod_{i=j+1}^{n} m_i$$

and $m_{j+1} < m_{j+2} < \cdots < m_n$. There can be several such partitions available and some consequences of the choice will be discussed later.

Let

$$\prod_{i=1}^{n} m_i = 2K$$

and let all X, $0 \leq X \leq K-1$ represent positive integers and all X', $K \leq X' \leq 2K-1$ represent the corresponding negative integers such that $X + X' = 0 \mod 2K$. Let $(x_1, x_2, \cdots, x_n)$ be the residue representation of an integer X, where $x_i = |X|_{m_i}$, $i = 1, 2, \cdots, n$. Let

$$2m\left(\prod_{i=2}^{j} m_i\right) = 2M \quad \text{and} \quad \prod_{i=j+1}^{n} m_i = N.$$

We shall find $|X|_{2M}$ and $||X|_N|_{2M}$ and represent them in mod $m_1, m_2, \cdots, m_j$ since we do not have any mod $2M$ arithmetic unit in the system. In doing so, we have to use the following lemma.

Lemma 1: If $(x_1, x_2, \cdots, x_j)$ is the residue representation for an integer X in a system of mutually prime moduli $m_1, m_2, \cdots, m_j$ such that $x_i = |X|_{m_i}$, $i = 1, 2, \cdots, j$, then

$$||X|_{m_1 m_2 \cdots m_j}|_{m_i} = |X|_{m_i} = x_i.$$

Proof:

$$x_i = |X|_{m_i}$$

or

$$X = y_i m_i + x_i \qquad \text{for some } y_i \geq 0.$$

Let

$$A = |X|_{m_1 m_2 \ldots m_j}.$$

Then

$$X = Y \cdot m_1 m_2 \cdots m_j + A, \qquad Y \geq 0$$

or

$$y_i m_i + x_i = Y \cdot m_1 m_2 \cdots m_i \cdots m_j + A$$

or

$$|A|_{m_i} = x_i$$

or

$$\big| |X|_{m_1 m_2 \ldots m_j} \big|_{m_i} = x_i = |X|_{m_i},$$

which proves the lemma.

Therefore, $\big| \| X|_N |_{2M} \big|_{m_i} = \| X|_N |_{m_i}$, since $2M = m_1 m_2 \cdots m_j$.

To determine $|X|_N$, we use the mixed-radix conversion process for the set of moduli $m_{j+1}, \cdots, m_n$ [5], [8]:

$$|X|_N = r_n m_{j+1} m_{j+2} \cdots m_{n-1} + \cdots + r_{j+2} m_{j+1} + r_{j+1},$$
$$0 \leq r_k < m_k, \qquad k = j+1, \cdots, n.$$

It is well-known how the mixed-radix digits r_k can be determined from the corresponding residue digits [10].

$$\big| |X|_N \big|_{m_i} = \big| r_n m_{j+1} \cdots m_{n-1} + \cdots + r_{j+2} m_{j+1} + r_{j+1} \big|_{m_i}.$$

Using

$$|A + B|_m = \big| |A|_m + |B|_m \big|_m$$

and

$$|A \cdot B|_m = \big| |A|_m \cdot |B|_m \big|_m$$
$$\big| |X|_N \big|_{m_i} = \Big| \big| |r_n|_{m_i} \cdot |m_{j+1}|_{m_i} \cdot \cdots \cdot |m_{n-1}|_{m_i} \big|_{m_i} + \cdots + |r_{j+1}|_{m_i} \Big|_{m_i}.$$

Once the choice of the moduli has been made, the fixed quantities in the above relation are known and can be stored. Then all we have to do is to compute the quantities $|r_k|_{m_i}, i = 1, \cdots, j; k = j+1, \cdots, n$. Once $\| X|_N |_{m_i}$ has been computed, it is compared with $|X|_{m_i}$. If each of $\| X|_N |_{m_i}$ and $|X|_{m_i}$ compare, then the number $(x_1, x_2, \cdots, x_n)$ is positive. If they do not compare, we add $l = |N|_{2M}$ to $\| X|_N |_{m_i}$ and again compare with $|X|_{m_i}$. In the worst case $(M-1)$ additions of l will be required before we can say the number is not positive. This will be the case for numbers near the boundary separating the positive and the negative regions. A schematic flow chart for this method is shown in Fig. 2.

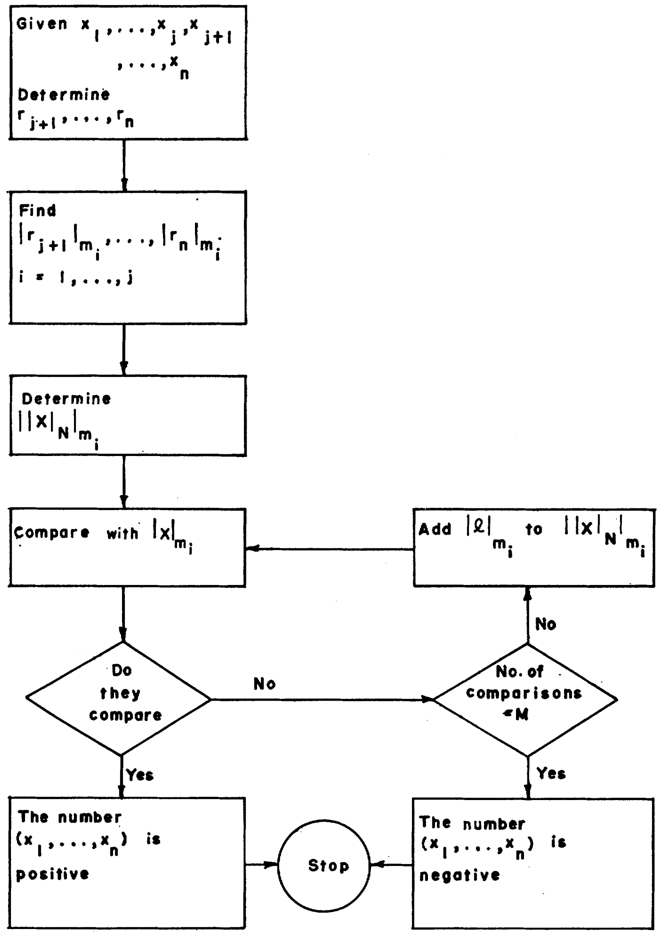

Fig. 2. General flow chart for sign detection.

Example 3: Consider the set of moduli 2, 3, 5, 7, and 11. Let us partition the set so that $2M = 2 \cdot 3 = 6$ and $N = 5 \cdot 7 \cdot 11 = 385$. Let us consider a residue number $X = (0, 1, 0, 1, 4)$. i varies from 0 to $M-1$ or from 0 to 2.

By Lemma 1

$$\big| |X|_6 \big|_2 = |X|_2 = 0$$
$$\big| |X|_6 \big|_3 = |X|_3 = 1.$$

Now, we shall find $\| X|_{385} |_2$ and $\| X|_{385} |_3$, which are nothing but $\| X|_{385} |_6$ represented in mod 2 and mod 3. To find $|X|_{385}$ we have to use the mixed-radix conversion process [10] for moduli 5, 7, and 11.

	Mixed-Radix Conversion			Mixed-Radix Digits
Moduli	5	7	11	
Residues	0	1	4	$r_3 = 0$
Subtract $\lvert r_3 \rvert_{m_i}$	0	0	0	
	0	1	4	
Multiply by $\lvert \frac{1}{5} \rvert_{m_i}$		3	9	
		3	3	$r_4 = 3$
Subtract $\lvert r_4 \rvert_{m_i}$		3	3	
		0	0	
Multiply by $\lvert \frac{1}{7} \rvert_{m_i}$			8	
			0	$r_5 = 0$

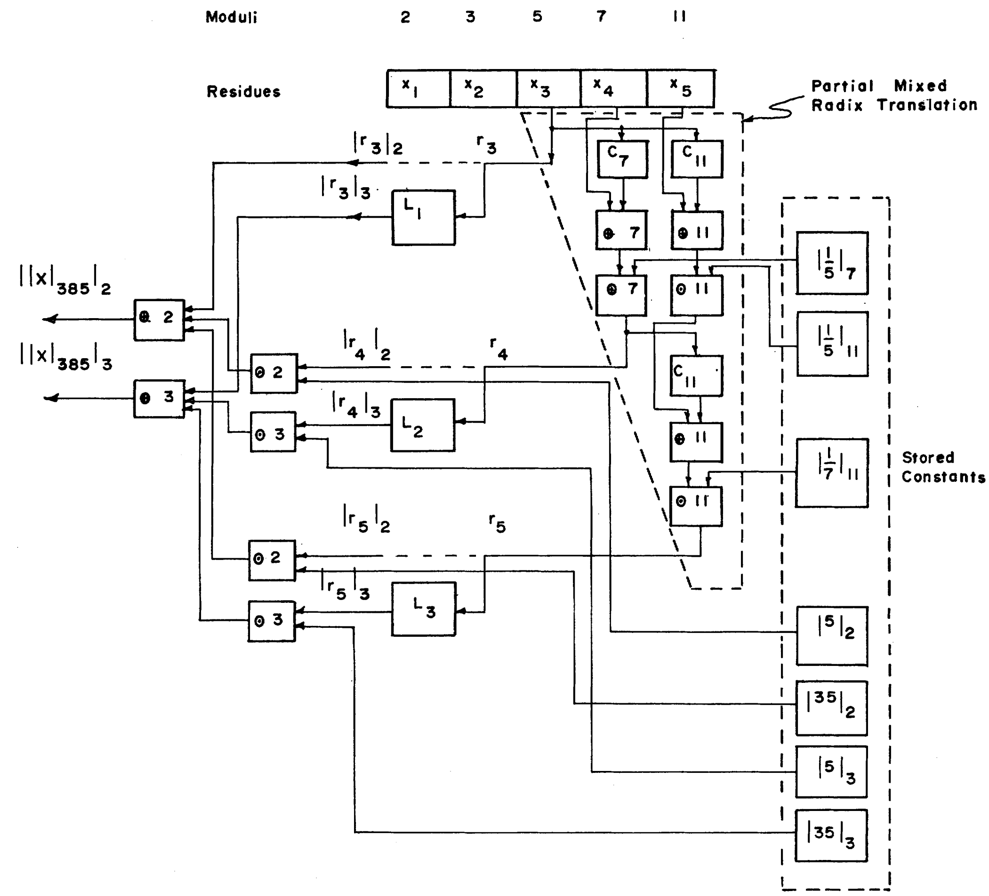

Fig. 3. Initial stage for sign detection for Example 3.

$$|X|_{385} = r_5 \cdot m_3 \cdot m_4 + r_4 \cdot m_3 + r_3$$
$$= r_5 \cdot 35 + r_4 \cdot 5 + r_3.$$

Therefore

$$\big||X|_{385}\big|_2 = \big|\,\big||r_5|_2 \cdot |35|_2\big|_2 + \big||r_4|_2 \cdot |5|_2\big|_2 + |r_3|_2\big|_2$$
$$= \big||0 \cdot 1|_2 + |1 \cdot 1|_2 + 0\big|_2$$
$$= 1$$
$$\big||X|_{385}\big|_3 = \big|\,\big||r_5|_3 \cdot |35|_3\big|_3 + \big||r_4|_3 \cdot |5|_3\big|_3 + |r_3|_3\big|_3$$
$$= \big||0 \cdot 2|_3 + |0 \cdot 2|_3 + 0\big|_3$$
$$= 0.$$

In mod 2 and mod 3

$$|X|_6 = (0, 1)$$
$$|X|_{385} = (1, 0).$$

They do not compare. So we add l to $|X|_{385}$.

$$l = |385|_6 = 1 = (1, 1)$$

in mod 2, mod 3 representation.

$$\begin{array}{r} |X|_{385} + l = (1,\ 0) \\ + \ (1,\ 1) \\ \hline (0,\ 1). \end{array}$$

This compares with $|X|_6$. Hence the number (0, 1, 0, 1, 4) is positive. Consider another residue number (0, 0, 0, 3, 1). We can check that

$$\big||X|_{385}\big|_2 = 1$$
$$\big||X|_{385}\big|_3 = 0.$$

In mod 2 and mod 3 representation

$$|X|_6 = (0, 0)$$
$$|X|_{385} = (1, 0).$$

They do not compare. Hence we add l to $|X|_{385}$; $|X|_{385} + l = (0, 1)$. It still does not compare with $|X|_6$. Adding l again

$$|X|_{385} + 2l = (1, 2).$$

Again, this does not compare with $|X|_6$. We have made $M = 3$ comparisons. Hence the number (0, 0, 0, 3, 1) is negative. Fig. 3 shows schematically how to find $||X|_{385}|_2$ and $||X|_{385}|_3$. Circled "+" and "·" denote modular operations. Boxes labeled C_7 and C_{11} are complement units mod 7 and 11, respectively. L_1, L_2, and L_3 represent simple logic required to convert r_3, r_4, and r_5 to $|r_3|_3$, $|r_4|_3$, and $|r_5|_3$, respectively.

The logic for comparing $||X|_{385}|_2$ and $||X|_{385}|_3$ with $|X|_2$ and $|X|_3$, respectively, is shown in Fig. 4 and 5. The output in each case is 1 only for identical inputs. When both comparators have outputs of 1, the sign of the residue number $(x_1, x_2, x_3, x_4, x_5)$ is positive. The

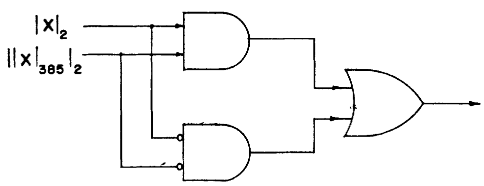

Fig. 4. Mod 2 comparatoı.

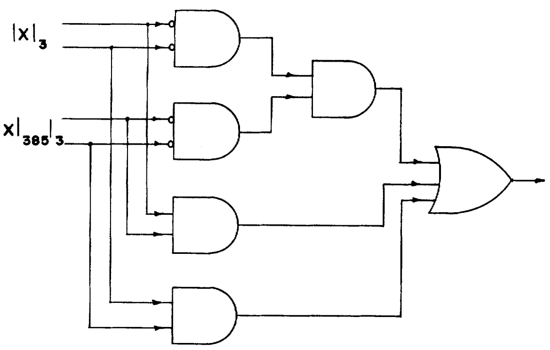

Fig. 5. Mod 3 comparator.

TABLE II
TABLE FOR EXAMPLE 3

$i=0$		$i=1$		$i=2$	
y_1	y_2	y_1^*	y_2^*	y_1^{**}	y_2^{**}
0	0	1	1	0	2
1	1	0	2	1	0
0	2	1	0	0	1
1	0	0	1	1	2
0	1	1	2	0	0
1	2	0	0	1	1

following discussion shows how the additions of il to $||X|_{385}|_2$ and $||X|_{385}|_3$ are achieved.

Let $y_1=||X|_{385}|_2$, $y_2=||X|_{385}|_3$, $y_1^*=|y_1+|l|_2|_2$, $y_2^*=|y_2+|l|_3|_3$, $y_1^{**}=|y_1^*+|l|_2|_2$, and $y_2^{**}=|y_2^*+|l|_3|_3$. Table II shows the combinations of y_1 and y_2 and the resulting combinations of y_1^*, y_2^*, y_1^{**}, and y_2^{**}.

It is clear that $y_1^*=\bar{y}_1$, $y_2^*=(y_2+1)$ mod 3, $y_1^{**}=y_1$, and $y_2^{**}=(y_2+2)$ mod 3. Fig. 6 shows the remaining part of the sign detection process after y_1 and y_2 have been determined. The stored constants 1 and 2 are added to y_2 in mod 3 adders to find y_2^* and y_2^{**}, respectively.

The mixed-radix translation can be reduced by partitioning the set of moduli into 30(2, 3, 5) and 77(7 and 11) or into 42(2, 3 and 7) and 55(5 and 11). This would require one complementation, one addition, and one multiplication, all modulo 11, in order to determine the two mixed-radix digits. Thus a reduction in hardware for mixed-radix translation is possible. But, on the other hand, more hardware is required in the remaining sign detection process after the two mixed-radix digits r_4 and r_5 are determined. Partitioning the set of moduli into 2 and 1155(3, 5, 7 and 11) would require more hardware for mixed-radix translation. The remaining

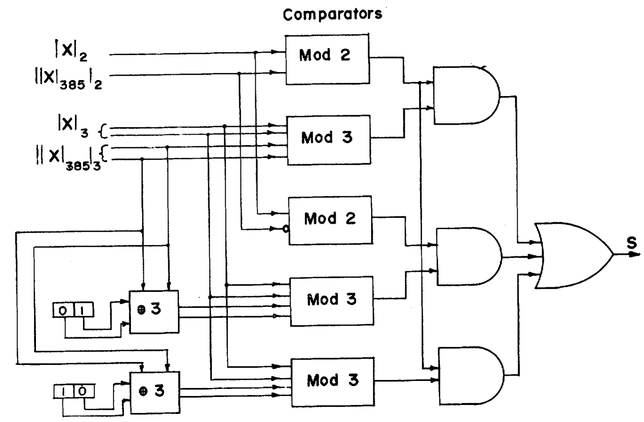

Fig. 6. Final stage for sign detection for Example 3.

process of sign detection requires only an EXCLUSIVE-OR gate, to which $|X|_2$ and the least significant bits of mixed-radix digits are fed (see Corollary 2). A compromise between this partial mixed-radix translation and the remaining sign detection process has to be finally made by the designer.

Thus far we have presented a completely hardware approach to sign detection. A combination of software and hardware can be used resulting in a saving on hardware. The process allows the designer a lot of flexibility in this respect.

It is interesting to study the case when in a system of n mutually prime moduli, the even modulus is 2. Then the following corollary can be used to establish the sign of a residue number.

Corollary 2: In a residue system consisting of n mutually prime moduli $m_1, m_2, \cdots, m_n$ where one of the moduli, say $m_1=2$, the sign of a number X with residue representation $(x_1, x_2, \cdots, x_n)$ is given by the EXCLUSIVE-OR sum of $|X|_2=x_1$ and the least significant bits of the mixed radix digits corresponding to the residue representation $(x_2, \cdots, x_n)$.

Proof: The proof follows directly from Corollary 1. Let

$$N = \prod_{i=2}^{n} m_i.$$

If we just had two moduli, $m_1=2$ and the composite modulus N, then we could use Corollary 1 for sign detection. We can still use it because N consists of $(n-1)$ "component moduli." $|X|_N$ is given by

$$|X|_N = r_n m_2 \cdot m_3 \cdot \cdots \cdot m_{n-1} + \cdots + r_3 m_2 + r_2,$$

r_i being the mixed-radix digits corresponding to the residue representation $(x_2, \cdots, x_n)$.

Hence

$$||X|_N|_2 = |||r_n|_2 \cdot |m_2|_2 \cdot \cdots \cdot |m_{n-1}|_2|_2 + \cdots + ||r_3|_2 \cdot |m_2|_2|_2 + |r_2|_2|_2.$$

Since moduli $m_2, \cdots, m_n$ are all odd (because m_1 is even) by assumption

$$|m_i|_2 = 1, \qquad \text{for all } i = 2, \cdots, n.$$

Hence

$$||X|_N|_2 = ||r_n|_2 + \cdots + |r_3|_2 + |r_2|_2|_2.$$

Also

$$|X|_2 = x_1.$$

From Corollary 1, the sign function is given by the mod 2 or the EXCLUSIVE-OR sum of $|X|_2$ and $||X|_N|_2$. Hence

$$S = |x_1 + |r_n|_2 + \cdots + |r_3|_2 + |r_2|_2|_2$$

which is the EXCLUSIVE-OR sum of x_1 and the least significant bits of r_i. This completes the proof.

If binary coded residues are used for carrying on residue arithmetic, the mixed-radix digits $r_2, \cdots, r_n$ will be in binary form and their least significant bits will be readily available. Therefore, wires from least significant positions of x_1 and all the $|r_i|_2$ can be fed as inputs to an EXCLUSIVE-OR gate whose output is the sign function S.

Binary to Residue Translation

We shall now consider the problem of converting the input data into modular form. The input will be in a fixed-radix form, with radix m. Any number X can be represented by a polynomial

$$X = C_n m^n + C_{n-1}m^{n-1} + \cdots + C_1 m + C_0,$$
$$0 \leq C_j < m, \qquad j = 0, 1, \cdots, n.$$

Then

$$|X|_{m_i} = ||C_n m^n|_{m_i} + |C_{n-1}m^{n-1}|_{m_i} + \cdots + |C_0|_{m_i}|_{m_i}.$$

If the quantities $|C_n m^j|_{m_i}$ can be evaluated without the actual multiplication mod m_i being performed, then $|X|_{m_i}$ can be determined using a mod m_i adder.

Let us consider binary inputs for a system with moduli 2, 3, 5, and 7. This system can uniquely represent 210 integers. For binary coding of 210 integers, the number of bits required is equal to $\langle \log_2 210 \rangle = 8$, where $\langle I \rangle$ denotes the least integer greater than or equal to I. Any X, $0 \leq X < 210$, with an 8-bit binary code can be expressed as a polynomial

$$X = C_7 2^7 + C_6 2^6 + C_5 2^5 + C_4 2^4 + C_3 2^3 + C_2 2^2 + C_1 2^1 + C_0, \qquad 0 \leq C_j < 2.$$

Obviously any C_j can now take only two values—0 or 1. Then

$$|X|_2 = |C_0|_2 = C_0$$
$$|X|_3 = |2C_7 + C_6 + 2C_5 + C_4 + 2C_3 + C_2 + 2C_1 + C_0|_3$$
$$|X|_5 = |3C_7 + 4C_6 + 2C_5 + C_4 + 3C_3 + 4C_2 + 2C_1 + C_0|_5$$
$$|X|_7 = |2C_7 + C_6 + 4C_5 + 2C_4 + C_3 + 4C_2 + 2C_1 + C_0|_7.$$

TABLE III

A. CONVERSION OF C_i FOR PROPER INPUT TO MOD 3 ADDER

Coefficient of C_i	Operation
1	add 0 to left of C_i
2	add 0 to right of C_i

B. CONVERSION OF C_i FOR PROPER INPUT TO MOD 5 ADDER

Coefficient of C_i	Operation
1	add 00 to left of C_i
2	add 0 to left and right of C_i
3	add $0C_i$ to left
4	add 00 to right

C. CONVERSION OF C_i FOR PROPER INPUT TO MOD 7 ADDER

Coefficient of C_i	Operation
1	add 00 to left of C_i
2	add 0 to left and right of C_i
4	add 00 to right of C_i

All these multiplications and additions in different moduli occur simultaneously. The maximum time taken for input translation is then the time required in the worst of the above cases. The translation process can be speeded up, with no extra hardware, by not using the modular multipliers. Then the process will need at most 7 additions. To discuss this in a little more detail, let us focus our attention on the translation equations. Let us assume binary coding for all the operands and the use of 2-state switching devices in the mechanization of the machine. The inputs to the mod 3 adder must be 2-bit numbers. Therefore, C_0, C_2, C_4, and C_6 (all one-bit numbers) must be converted into 2-bit numbers by adding a 0 at the left. $2C_1$, $2C_3$, $2C_5$, and $2C_7$ can only take the values 0 and 2, which in binary code are 00 and 10, respectively. Therefore, we can place a 0 to the right of these quantities before feeding into the mod 3 adder. In other words, depending on whether the coefficient of any C_j is 1 or 2, we have to add a 0 to the left or right of C_j, as the case may be, before feeding it to the mod 3 adder. Similar additions of 0's to the left or right of C_j would be necessary before feeding them to the mod 5 and mod 7 adders. Table III shows the effect of the coefficients of C_j on the addition of 0's and 1's to the left or right of C_j.

CONCLUSIONS

In this paper, within the limitations imposed by Szabo's Coding Theorem, we have found an explicit formula for the sign function S. In the special case when one of the two moduli is 2, S is the EXCLUSIVE-OR sum of two bits. The theorem for two moduli has been extended to the general case of n moduli and it has been shown that in the special case when one of the n moduli is 2, S is the EXCLUSIVE-OR sum of n bits.

It should be pointed out that the use of the mixed-radix translation process has been previously suggested for sign detection, where it is used as an intermediate step in translating from the residue code to binary or decimal form. However, we have shown that sign detection is possible by using partial mixed-radix conversion and additional combinational circuitry or software, as desired, without translating to decimal or binary form. Our approach provides a degree of flexibility in implementation. It has been shown that varying amounts of hardware and software can be used, depending on the system requirements and design.

A very simple solution for the binary to residue translation problem has been found. It has been shown that this requires modular adders only. There is no need for modular multiplication in this process and this certainly reduces the complexity and time required for binary to residue translation.

References

[1] H. Aiken and W. Semon, "Advanced digital computer logic," Computing Labs., Harvard University, Cambridge, Mass., Rept. WADS TR-59-472, July 1959.

[2] P. W. Cheney, "A digital correlator based on the residue number system," *IRE Trans. Electronic Computers*, vol. EC-10, pp. 63–70, March 1961.

[3] W. E. Deskins, *Abstract Algebra*. New York: Collier-Macmillan, 1964.

[4] I. Flores, *The Logic of Computer Arithmetic*. Englewood Cliffs, N. J.: Prentice-Hall, 1963.

[5] H. L. Garner, "The residue number system," *IRE Trans. Electronic Computers*, vol. EC-8, pp. 140–147, June 1959.

[6] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*. New York: Oxford, 1954.

[7] A. Svoboda, "The numerical system of residual classes (SRC)," *Computer Prog. Czechoslovakia*.

[8] N. Szabo, "Sign detection in nonredundant residue systems," *IRE Trans. Electronic Computers*, vol. EC-11, pp. 494–500, August 1962.

[9] "Modular arithmetic computing techniques," Wright-Patterson Air Force Base, Ohio, Rept. ASD-TDR-63-280, May 1963.

[10] N. S. Szabo and R. I. Tanaka, *Residue Arithmetic and its Applications to Computer Technology*. New York: McGraw-Hill, 1967.

A Statistical Approach to the Computation of Delays in Logic Circuits

H. DANIEL SCHNURMANN, MEMBER, IEEE, AND KLIM MALING

***Abstract*—This paper describes a method whereby multiple regression techniques are used to predict the delay between input and output signals through a combinatorial logic chain.**

A delay equation is developed whose variables are chosen to represent the variations found in an actual system environment.

First, a technique is described whereby a statistical delay model for a single circuit is developed from measurements on many circuits. The total delay through logic chains of various lengths is predicted by successive application of this model to the elements of the chain. The accuracy of the prediction was checked experimentally, and agreement was found to be very good.

***Index Terms*—Combinational logic, computer environmental variations, delay equation, logic chain delay, multivariate regression analysis, prediction technique, statistical modeling.**

I. Introduction

THE problem of predicting delays in any logic configuration is of great importance for the design and proper operation of high-performance digital computers. The logic designer must know with a high degree of confidence that a pulse will arrive at a given place in the system at a given time, since its failure to do so will cause marginal operation or will generate a system malfunction. On the other hand, if an overly conservative design is made to achieve the desired level of confidence, one must accept the penalty of lower system performance.

The method of using delay equations to predict delay is one of the most widely used approaches. This equation is a mathematical expression which describes qualitatively as well as quantitatively the effects of environment and of loading on the circuit performance. It takes into account factors such as variations of power supply voltages, shape characteristics of input pulses, temperature, etc., and also takes into consideration the geometry associated with logic configuration, such as distribution of loads along the main line, length of stubs, etc.

This approach has an advantage over other methods such as simulation techniques, equivalent circuit methods, etc., in that it is simple to develop and easy to use. There are, however, several disadvantages which cannot be ignored.

1) The delay equation is empirically determined. Consequently, in order to generate a delay model which gives meaningful results, a large number of experiments

Manuscript received May 31, 1968; revised December 31, 1968.

H. D. Schnurmann is with IBM Corporation, Thomas J. Watson Research Center, Yorktown Heights, N. Y. 10598.

K. Maling is with IBM Corporation, Systems Development Division, Poughkeepsie, N. Y.