Logical Implementation of Arithmetic Operations in the Symmetric Residue Number System

Logical Implementation of Arithmetic Operations in the Symmetric Residue Number System

Eisuke KINOSHITA*, Hideo KOSAKO* and Yoshiaki KOJIMA*

(Received November 15, 1973)

Residue interacting operations can be split into five broad categories: scaling, mixed-radix conversion, base-extension, general division and floating-point arithmetic. These operations are fundamentally composed of the residue independent operations of addition, subtraction and multiplication. For implementation of the three arithmetic operations, the use of logical gate matrices offers conceptual simpliciy. This method has assumed that residues of a number are encoded in a "one of many" representation. This is obviously uneconomical in terms of matrix components. This paper discloses how the three residue operations mentioned above can be mechanized to advantege by using the symmetric residue notation.

1. Introduction

Considerable attempts have been made to use residue number theory for computation. The characteristic of the residue system of interest to computer designers is that in addition, subtraction and multiplication any particular digit of the result is dependent only on the corresponding operand digits. This property makes it possible to add, subtract and multiply without a carry (or borrow) and removes the need to form partial products in multiplication. The operations of addition, subtraction, multiplication and complementation are called residue independent operations. In the symmetric residue system, the complementation, i. e., finding the additive inverse of a residue digit can be accomplished merely by complementing the sign of the digit.

Other operations such as relative magnitude comparison, sign determination, overflow detection, scaling, general division, mixed-radix conversion, base-extension, and floating-point arithmetic are called residue interacting operations. These operations are fundamentally composed of residue addition, subtraction and multiplication.[1)3)4)]

For implementation of the three residue arithmetic operations mentioned above, the use of logical gate matrices offers conceptual simplicity. This method has assumed that residues of a number are encoded in a "one of many" representation. This is obviously uneconomical in terms of matrix components. It has been suggested by the Radio Corporation of America[2)] that a significant reduction in matrix components is realized by using a sign-magnitude notation, and the present investigation makes improvements in this idea.

* Department of Electronics, College of Engineering.

In the subsequent discussion it will be assumed that the reader is familiar with the basic concepts of residue arithmetic.

2. Symmetric Residue Representation

Let there be given a set of odd prime numbers $m_1, m_2, \cdots, m_n$ called moduli. Then, for any integer x in the interval $[-(M-1)/2, (M-1)/2]$, the least remainder in absolute value when x is divided by m_i may be computed, where $M=\prod_{i=1}^{n} m_i$. This quantity is represented by the symbol $/x/_{m_i}$ and is an integer such that $-(m_i-1)/2 \leqq /x/_{m_i} \leqq (m_i-1)/2$. For a given set of moduli an n-tuple $\{/x/_{m_1}, /x/_{m_2}, \cdots, /x/_{m_n}\}$ uniquely represents x. This n-tuple is called the symmetric residue representation of x. The integer $/x/_{m_i}$ is called the ith symmetric residue digit of x.

It is possible to reconstruct the natural number from its residue representation by means of the mixed-radix conversion process.[1] This procedure converts the residue code of a number to its mixed-radix representation.

The method to be described will be applicable only to residue systems with moduli $m_i \geqq 7$. This method may, however, be modified to remove this restriction.

3. Coding the Symmetric Residue Digit

For residue arithmetic, the use of matrix units is attractive because an array of logical components can be wired to give a direct representation of the truth table for the residue arithmetic operation. The relative ease with which the quantities can be obtained is a direct consequence of using 1-out-of-m_i coding, where m_i is the modulus in which the operation is being performed. The number of components required for each matrix is m_i^2.

The symmetric residue representation is a sign-magnitude notation. In this representation a residue digit is expressed in 2-out-of-$(m_i+1)/2$ code. Table 1 shows the 2-out-of-$(m_i+1)/2$ coding for modulus 7. For modulo-m_i matrix, the coding scheme reduces the number of matrix components to $(m_i+1)(m_i+3)/8$, as will be seen later.

Table 1. Binary coding of modulo-7 residue in the 2-out-of-4 code.

Digit	Code			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
−3	1	1	0	0
−2	1	0	1	0
−1	1	0	0	1

4. Arithmetic Unit

Since most digital storage devices consist of two states, the 2-out-of-$(m_i+1)/2$

coding is uneconomical in terms of storage efficiency, too. Hence a recoding process must take place whenever information passes between the arithmetic unit and the storage device.

For the symmetric residue system consisting of moduli $m_1, m_2, \cdots, m_n$ an integer x will be represented by n symmetric residues. The number range for the ith residue digit $/x/_{m_i}$ is

$$-Q_i \leqq /x/_{m_i} \leqq Q_i,$$

where $Q_i = (m_i - 1)/2 \geqq 3$ (recalling that $m_i \geqq 7$).

Now let us assume that $/x/_{m_i}$ is stored in binary code and has t bits, the sign bit and $t-1$ magnitude bits, where $2^{t-2} \leqq Q_i \leqq 2^{t-1} - 1$. For operation modulo m_i, an arithmetic unit employs three registers, as shown in Fig. 1.

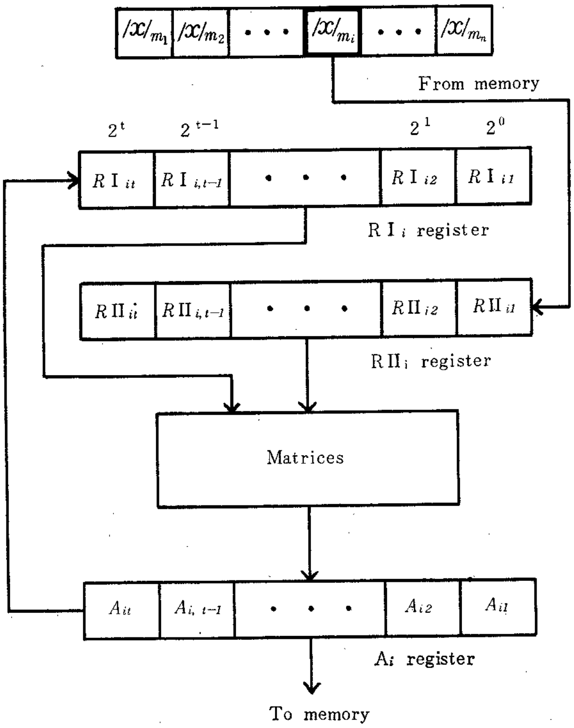

Fig. 1. Three registers for modulo-m_i operations.

These registers are named RI_i, RII_i, and A_i. Capital letters with subscripts $i1$ to it denote the t flipflops in the registers of Fig. 1. RI_{i1}, RII_{i1}, and A_{i1} aer the least significant bits; RI_{it}, RII_{it}, and A_{it}, the sign bits.

Register RI_i stores the first operand transferred from the A_i register, the result of the previous arithmetic operation. Register RII_i merely stores the second operand fetched from memory during arithmetic operations. Register A_i functions as an accumulator. It stores the result of arithmetic operations involving the first and second operands and upon receiving the result transfers it to the RI_i register.

Three separate matrices are provided to perform addition, subtraction and multiplication modulo m_i. Since in the symmetric residue system the only difference between positive and negative numbers is the sign bit, each matrix is implemented assuming that the ith residue digits of the operands are positive. The correct sign is determined by means of additional equipment. The magnitude bits of the RI_i and RII_i registers are decoded into the 1-out-of-$(m_i+1)/2$ coding form. The output lines from each decoder are connected to the three matrices. To simplify the following description, we use Q for Q_i.

4.1. Add Matrix Unit

From the standpoint of logic, the arithmetic tables for the residue operations are essentially truth tables. The modulo m_i addition of positive residues shown in Tabel 2 can be interpreted as truth table of two $(m_i+1)/2$-valued variables.

Table 2. Modulo-m_i addition table.

+	0	1	2	· · ·	Q
0	0	1	2	· · ·	Q
1	1	2	3	· · ·	$-Q$
2	2	3	$/4/m_i$	· · ·	$-(Q-1)$
·	·	·	·		·
·	·	·	·		·
·	·	·	·		·
Q	Q	$-Q$	$-(Q-1)$	· · ·	-1

Now let A denote this table, and let a_{kl} be the (k, l) elements $(k, l=1, 2, \cdots, Q+1)$ of A. Then we have

$$a_{11}=0, \quad a_{Q+1, Q+1}=-1,$$

$$a_{k1}=a_{k-1,2}=\cdots\cdots=a_{1k}=k-1>0 \quad \text{for} \quad k=2, 3, \cdots\cdots, Q+1$$

and

$$a_{Q+1,l}=a_{Q,l+1}=\cdots\cdots=a_{l,Q+1}=-(Q+2-l)<0 \quad \text{for} \quad l=2, 3, \cdots\cdots, Q$$

Proof.

$$a_{11}=/0+0/_{m_i}=0.$$

$$a_{Q+1, Q+1}=/Q+Q/_{m_i}=/m_i-1/_{m_i}=/m_i-1-m_i/_{m_i}=-1.$$

For the subscripts u and v such that $v=k+1-u$ $(u=k, k-1, \cdots\cdots, 1)$,

$$a_{uv}=/u-1+v-1/_{m_i}=/u+v-2/_{m_i}=/k-1/_{m_i}.$$

Since $2\leqq k\leqq Q+1$, $1\leqq k-1\leqq Q$.

Hence,

$$a_{uv}=/k-1/_{m_i}=k-1>0.$$

In similar fashion, we have

$$a_{u-1, v+1}=k-1.$$

For the subscripts u and v such that $u=Q+1+l-v$ $(v=l, l+1, \cdots\cdots, Q+1)$, it can be easily seen that

$$a_{uv}=a_{u-1,\ v+1}=/u+v-2/_{m_i}=/Q+l-1/_{m_i}.$$

Since $\quad 2\leqq l\leqq Q,\ Q+1\leqq Q+l-1\leqq 2Q-1.$

Therefore,

$$-Q\leqq Q+l-1-m_i=-(Q+2-l)\leqq -2.$$

Hence we have

$$/Q+l-1/_{m_i}=-(Q+2-l)<0.$$

From the above examination, it is possible to reduce the number of elements in A. The reduced truth table itself is represented by a folded matrix. Fig. 2 illustrates the wiring for a modulo-7 add matrix. Two of the lines, 0, 1, 2, 3, to the matrix are energized at one time except when the magnitudes of the contents of the $R\text{I}_i$ and $R\text{II}_i$ registers are equal. Line E in Fig. 2 is energized by control logic whenever the magnitudes are equal, and the output S_i is the sign bit.

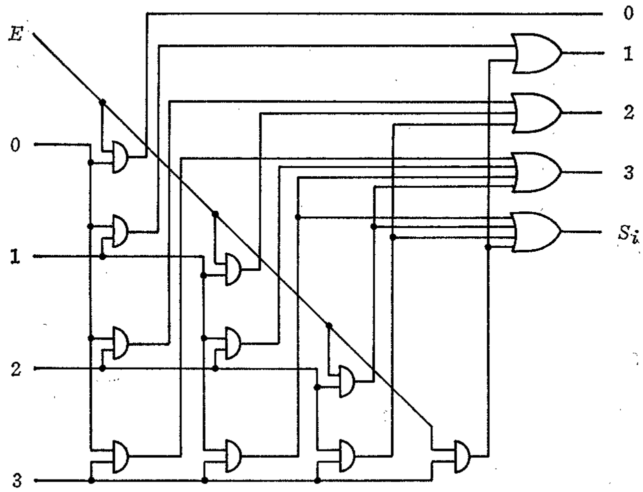

Fig. 2. Add matrix unit for $m_i=7$.

4.2. Subtract Matrix Unit

Table 3 illustrates the modulo m_i subtraction of positive residues in tabular form.

Table 3 Modulo-m_i subtraction table.

	−	Minuend 0	1	2	·	·	·	Q
Subtrahend	0	0	1	2	·	·	·	Q
	1	−1	0	1	·	·	·	$Q-1$
	2	−2	−1	0	·	·	·	$Q-2$
	·	·	·	·				·
	·	·	·	·				·
	·	·	·	·				·
	Q	$-Q$	$-(Q-1)$	$-(Q-2)$	·	·	·	0

Let B denote this table, and let b_{kl} be the (k, l) elements $(k, l=1, 2, \cdots, Q+1)$ of B. Then we have

$$b_{kk}=0,\ b_{1,Q+1}=Q,\ b_{Q+1,1}=-Q,$$

$$b_{1l}=b_{2,l+1}=\cdots\cdots=b_{Q+2-l,Q+1}=l-1>0 \quad \text{for} \quad l=2, 3, \cdots\cdots, Q$$

and $$b_{k1}=b_{k+1,2}=\cdots\cdots=b_{Q+1,Q+2-k}=-(k-1)<0 \ \text{for}\ k=2, 3, \cdots\cdots, Q,$$

which may be easily proven.

The above relationships make possible a reduction in the size of a subtract matrix. The difference winding for a modulo-7 subtract matrix is shown in Fig. 3.

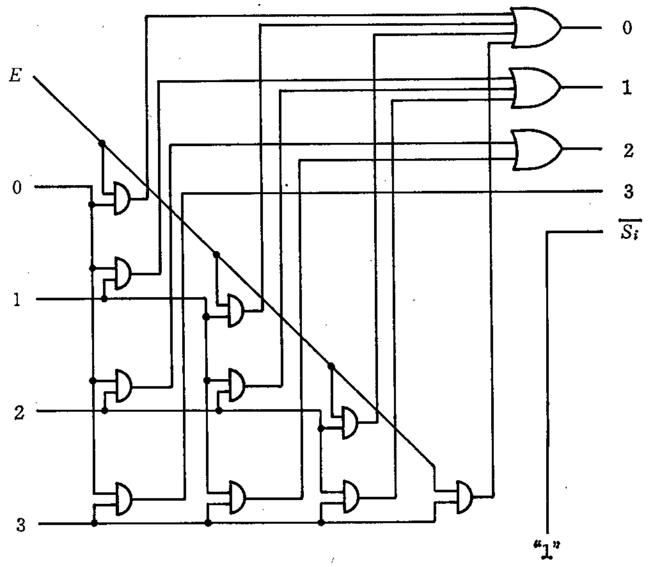

Fig. 3. Subtract matrix unit for $m_i=7$.

Note that the sign bit of the output is positive. This implies that the subtract matrix forms the difference in absolute value.

4.3. Multiplication Using Index Calculus

To have simpler product windings, let us consider index sum multiplication. According to algebraic knowledge, if the modulus m_i is a prime and p denotes a primitive root of m_i, then every nonzero modulo-m_i residue digit can be expressed uniquely as the modulo-m_i residue of p raised to some power. Table 4 illustrates

Table 4. Modulo-7 symmetric residue digits expressed with powers of 3.

Residue digit	Index
$1=/3^0/_7$	0
$2=/3^2/_7$	2
$3=/3^1/_7$	1
$-3=/3^{-2}/_7$	-2
$-2=/3^{-1}/_7$	-1
$-1=/3^3/_7$	3

how the residue digit and the integer power of p correspond each other for $m_i=7$, $p=3$.

From Fermat's Theorem, we obtain

$$|p^{m_i-1}|_{m_i}=1. \tag{1}$$

Then for any integer x we may write

$$x=q(m_i-1)+|x|_{m_i-1}$$

and

$$|p^x|_{m_i}=|p^{q(m_i-1)}\times p^{|x|_{m_i-1}}|_{m_i}, \tag{2}$$

where q is an integer. But, by (1), the first term of (2) on the right-hand side must be equal to 1. Consequently,

$$|p^x|_{m_i}=|p^{|x|_{m_i-1}}|_{m_i}. \tag{3}$$

Now let S_r be the set of the modulo-m_i symmetric nonzero residues, and let S_e be the set of the exponents or indices corresponding to the residues. If the elements r_1 and r_2 of S_r are expressed in the form

$$r_1=|p^{e_1}|_{m_i}$$
$$r_2=|p^{e_2}|_{m_i},$$

it follows from (3) that

$$|r_1\times r_2|_{m_i}=|p^{|e_1+e_2|_{m_i-1}}|_{m_i}.$$

This means that rathar than carrying out modulo-m_i multiplication in S_r, we can carry out modulo m_i-1 addition in S_e and retransform the result into S_r.

Table 4 illustrates this method for performing multiplication. Assume that the mod-7 residue digits, 3 and -2, are to be multiplied. The corresponding indices are 1 and -1. The sum of these indices is 0, and the corresponding residue digit is 1.

Two useful relationships of correspondece are

$$|p^0|_{m_i}=1 \tag{4}$$

and

$$|p^{(m_i-1)/2}|_{m_i}=-1. \tag{5}$$

Proof. The proof of (4) is obvious.

Since

$$|p^{m_i-1}|_{m_i}=|p^{(m_i-1)/2}|_{m_i}\times|p^{(m_i-1)/2}|_{m_i},$$

it follows from (1) that

$$|p^{(m_i-1)/2}|_{m_i}=-1.$$

4.4. Multiply Matrix Unit

For the modulus m_i and its primitive root p, we assume that the indices, 0, 1, 2, $\cdots$, $Q-1$, $-(Q-1)$, $-(Q-2)$, $\cdots$, -1, have one to one correspondence to the residue digits, r_0, r_1, r_2, $\cdots$, r_{Q-1}, $r_{-(Q-1)}$, $r_{-(Q-2)}$, $\cdots$, r_{-1}.

The following simple property of the correspondence will be useful to construction of a multiplication table.

THEOREM For $u \neq v$ and $0 \leqq u,\ v \leqq Q-1$ the following holds:

$$|r_u| \neq |r_v|.$$

Proof. The proof is by contradiction.

Let $|r_u| = |r_v|$ and $u<v$, then

$$|/p^u/_{m_i}| = |/p^v/_{m_i}|.$$

Therefore,

$$/p^u/_{m_i} = \pm /p^v/_{m_i}.$$

This implies

$$/p^u/_{m_i} \pm /p^v/_{m_i} = 0,$$

which in turn implies

$$/p^u/_{m_i} \times /1 \pm p^{v-u}/_{m_i} = /p^u/_{m_i} \times (1 \pm /p^{v-u}/_{m_i}) = 0;$$

but

$$/p^u/_{m_i} \neq 0,$$

therefore the only way $/p^u/_{m_i} \times (1 \pm /p^{v-u}/_{m_i})$ can equal zero is for

$$1 \pm /p^{v-u}/_{m_i} = 0.$$

Consequently, from (4) or (5),

$$v-u=0 \quad \text{or} \quad v-u=Q. \tag{6}$$

It is apparent that $v-u=0$ contradicts the original assumption. Since $u<v$ and $0 \leqq u$, $v \leqq Q-1$,

$$1 \leqq v-u \leqq Q-1.$$

This contradicts (6).

The two contradictions prove the theorem.

It follows from the theorem that the residue digits r_k ($k=0, 1, 2, \cdots, Q-1$) are different from one another in absolute value. Hence a modulo m_i multiplication table as shown in Table 5 is constructed.

Table 5. Modulo-m_i multiplication table.

$\times$	r_0	r_1	r_2	$\cdot\ \cdot\ \cdot$	r_{Q-1}
r_0	$/r_0 \times r_0/_{m_i}$	$/r_0 \times r_1/_{m_i}$	$/r_0 \times r_2/_{m_i}$	$\cdot\ \cdot\ \cdot$	$/r_0 \times r_{Q-1}/_{m_i}$
r_0	$/r_1 \times r_0/_{m_i}$	$/r_1 \times r_1/_{m_i}$	$/r_1 \times r_2/_{m_i}$	$\cdot\ \cdot\ \cdot$	$/r_1 \times r_{Q-1}/_{m_i}$
r_2	$/r_2 \times r_0/_{m_i}$	$/r_2 \times r_1/_{m_i}$	$/r_2 \times r_2/_{m_i}$	$\cdot\ \cdot\ \cdot$	$/r_2 \times r_{Q-1}/_{m_i}$
$\cdot$	$\cdot$	$\cdot$	$\cdot$		$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$		$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$		$\cdot$
r_{Q-1}	$/r_{Q-1} \times r_0/_{m_i}$	$/r_{Q-1} \times r_1/_{m_i}$	$/r_{Q-1} \times r_2/_{m_i}$	$\cdot\ \cdot\ \cdot$	$/r_{Q-1} \times r_{Q-1}/_{m_i}$

Let C denote this table, and let c_{kl} be the (k, l) elements ($k, l=1, 2, \cdots, Q$) of C. Then we have

$$c_{k1} = c_{k-1,2} = \cdots\cdots = c_{1k} = r_{k-1} \quad \text{for} \quad k=1, 2, \cdots\cdots, Q$$

and

$$c_{Ql}=c_{Q-1,l+1}=\cdots\cdots=c_{lQ}=r_{/Q+l-2/_{m_i-1}} \text{ for } l=2,\ 3,\ \cdots,\ Q.$$

Proof. For the subscripts u and v such that $v=k+1-u$ ($u=k,\ k-1,\ \cdots,\ 1$),

$$c_{uv}=/r_{u-1}\times r_{v-1}/_{m_i}=//p^{u-1}/_{m_i}\times /p^{v-1}/_{m_i}/_{m_i}=/p^{/u+v-2/_{m_i-1}}/_{m_i}.$$

Since $u+v-2=k-1$ and $1\leqq k\leqq Q$, it follows that

$$0\leqq u+v-2\leqq Q-1.$$

Therefore,

$$c_{uv}=/p^{/u+v-2/_{m_i-1}}/_{m_i}=/p^{u+v-2}/_{m_i}=r_{u+v-2}=r_{k-1}.$$

In the same manner we have

$$c_{u-1,\ v+1}=r_{k-1}.$$

It can be seen easily that, for the subscripts u and v such that $u=l+Q-v$ ($v=l$, $l+1$, $\cdots$, Q),

$$c_{uv}=c_{u-1,\ v+1}=/p^{/u+v-2/_{m_i-1}}/_{m_i}.$$

Since $u+v-2=Q+l-2$ and $2\leqq l\leqq Q$,

$$Q\leqq u+v-2\leqq 2Q-2.$$

Hence,

$$/p^{/u+v-2/_{m_i-1}}/_{m_i}=r_{/u+v-2/_{m_i-1}}=r_{/Q+l-2/_{m_i-1}}.$$

From the assumption that the symmetric residue digits of the operands are positive, this table needs a little modification. If there are negative residue digits among r_0, r_1, $\cdots$, r_{Q-1}, each of them must be replaced by its absolute value. Consequently, the element c_{kl} must be replaced by $-c_{kl}$ when r_{k-1} and r_{l-1} have the opposite sign.

It is possible, from the above examination, to fold the multiply matrix unit. Fig. 4 illustrates the wiring for a modulo-7 multiplication matrix, assuming $p=3$. It should

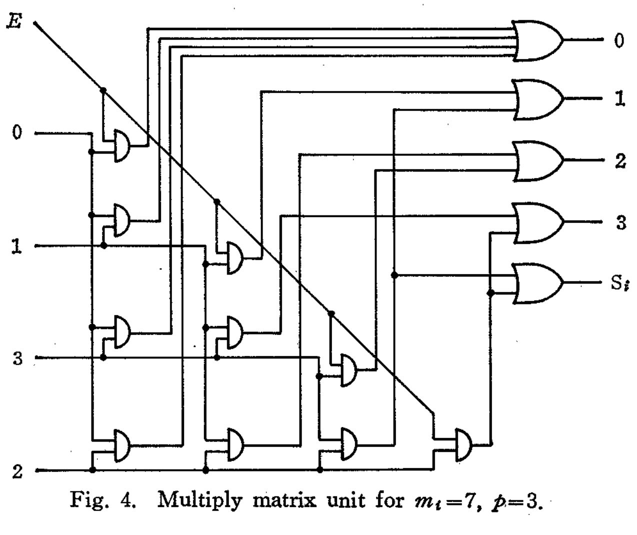

Fig. 4. Multiply matrix unit for $m_i=7$, $p=3$.

be noted that the order of the input variables is, in general, different from that of add or subtract matrix.

5. Selection of Matrix

Let us suppose that the computer control section generates the F_A or F_S function to indicate the add or subtract command, respectively. The addition or subtraction actually performed by the add or subtract matrix is determined by the F_{ADD} function or the F_{SUB} function generated in the add-subtract control block by combination of the functions F_A, F_S, and the two signs in RI_{it} and RII_{it} according to the truth table in Table 6.

Table 6. Truth table for F_{ADD}, F_{SUB}.

RI_{it}	F_A	F_S	RII_{it}	F_{ADD}	F_{SUB}
0	1	0	0	1	0
0	1	0	1	0	1
1	1	0	0	0	1
1	1	0	1	1	0
0	0	1	0	0	1
0	0	1	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1

The logic equations for the functions F_{ADD} and F_{SUB} are given as follows:

$$F_{ADD}=F_A\cdot\bar{G}+F_S\cdot G$$

$$F_{SUB}=F_B\cdot\bar{G}+F_A\cdot G,$$

where $G=RI_{it}\oplus RII_{it}$.

The computer control section generates also the F_M function to indicate the multiply command. The multiply matrix is selected directly by the function F_M.

6. Sign Determination of the Result

Initially, magnitude bits RI_{i1} to $RI_{i,t-1}$ and RII_{i1} to $RII_{i,t-1}$ are compared by the comparator 1 and comparator 2 to generate function F_{i1} and F_{i2}. Assume that

$$F_{i1}=1 \quad \text{when} \quad |(RI_i)|=|(RII_i)|$$

and $$F_{i2}=1 \quad \text{when} \quad |(RI_i)|>|RII_i)|,$$

where the parentheses denote contents of a register.

The arithmetic matrix selected generates the sign signal assuming the residue digits are positive. The sign of the result is determined through the combinatorial logic block which generates the function W_i. If $W_i=1$, the sign S_i from the matrix is to be complemented. If $W_i=0$, S_i is uncomplemented. The truth tables for W_i in three operations are given in Table 7, 8, and 9. From these tables we have

$$W_i=F_{ADD}\cdot RI_{it}+F_{SUB}\cdot\bar{F}_{i1}\cdot(RI_{it}\cdot F_{i2}+\overline{RI}_{it}\cdot\bar{F}_{i2})+F_M\cdot G.$$

The truth table for the correct sign A_{it} of the result is give in Table 10. Hence, we have

$$A_{it}=W_i \oplus S_i$$

Table 7. Truth table for W_i in add operation.

RI_{it}	RII_{it}	W_i
0	0	0
0	1	0
1	0	1
1	1	1

Table 8. Truth table for W_i in subtract operation.

RI_{it}	RII_{it}	F_{i1}	F_{i2}	W_i
0	0	0	0	1
0	1	0	0	1
1	0	0	0	0
1	1	0	0	0
0	0	0	1	0
0	1	0	1	0
1	0	0	1	1
1	1	0	1	1
0	0	1	0	0
0	1	1	0	0
1	0	1	0	0
1	1	1	0	0

Table 9. Truth table for W_i in multiply operation.

RI_{it}	RII_{it}	W_i
0	0	0
0	1	1
1	0	1
1	1	0

Table 10. Truth table for A_{it}.

W_i	S_i	A_{it}
0	0	0
1	0	1
0	1	1
1	1	0

7. Conclusion

Utilizing the concept of matrix unit, we have proposed a method for implementing addition, subtraction and multiplication in the symmetric residue system. A rule of the wiring for modulo-m_i arithmetic matrices was presented in Section 4. The proposed method could be realized with LSI techniques and would give high speed to the three residue operations.

References

1) N. S. Szabó and R. I. Tanaka, Residue Arithmetic and its Applications to Computer Technology, McGraw-Hill, New York (1967).
2) Radio Corporation of America, Aerospace Systems Division, AL TDR 64-86 (1961).
3) E. Kinoshita, H. Kosako and Y. Kojima, IEEE Trans., **C-22**, 134 (1973).
4) E. Kinoshita, H. Kosako and Y. Kojima, IEEE Trans., to be published.