MINIMIZATION METHODS FOR MACROCELLULAR ARITHMETIC NETWORKS

Renato DE MORI (*), Michele ELIA (°), Angelo SERRA (*)

(*) Istituto Elettrotecnico Nazionale "G. Ferraris" - CENS
Istituto di Elettrotecnica Generale del Politecnico

(°) Istituto Matematico del Politecnico -
Corso Duca degli Abruzzi, 24
10129 - TORINO - ITALY

SUMMARY

This paper presents a new method to study arithmetic combinatorial circuits. Using polynomial associated to the input-output sequences and to the system, it is possible to solve the problem of minimization of the number of the component blocks. Particularly, the important case of the multiple outputs elementary units can be treated.

Applications of the introduced procedures to multiplier and to fast networks for performing convolution are presented.

I. INTRODUCTION

The design of arithmetic arrays using macrocells seems to be very promising for many reasons. High speed of arithmetic operations is an important requirement for general purpose computers and for many special processors as well. The possibility of using macrocells greater than the conventional full-adders, in large combinatorial or pipelined arithmetic arrays, allows the designer of digital systems to obtain fast and complex networks with a relatively small number of equal modules. Furthermore, the designer of integrated circuits can proficously decide the complexity of arithmetic operators, considering the technological convenience of their implementation.

A recent paper proposed a formal characterization of the arithmetic combinatorial circuits and presented procedures to minimize the number of elementary "triangular" units /1/ in a parallel multiplier of positive numbers. Another paper [2] showed how very fast multipliers of signed numbers can be designed, using macrocellular iterative structures, whose cells are pseudo-adders having several inputs of some weights and one or more outputs for each weight. For these modules the techniques presented in [1] are not sufficient to solve the above-mentioned minimization problem.

The purpose of this paper is to present original methods for minimizing cellular arrays containing cells that can also be not triangular. The cells must be, hovewer, elementary "Operative Full Systems", according to the definition introduced in part II, that is realizable with only full-adders /2/. This class is the most important and excludes only the blocks that can be realized, either with full adders or half adders. The generalization of the precedures introduced here to all classes can be found in another paper [3]. Finally, example of applications of the minimization procedures to the multipliers described in [2] and to pipelined arrays for performing convolutions will be presented in parts IV and V.

II. ANALYSIS OF ARITHMETIC COMBINATORIAL CIRCUITS

In the following, many definitions and properties introduced by Meo [1] are extended to the general case and the polynomial transforms of the system sequences are introduced. They are useful either in analysis or in synthesis and offer simple and practical interpretations that are omitted here for the sake of brevity.

Definition 1: A Binary Arithmetic Combinatorial Circuit (BACC) is a combinatorial network having p+1 input sets I_i and q+1 output sets U_j:

$$I_i = \left\{x_{i1}, x_{i2}, \dots, x_{im_i}\right\} \qquad 0 \le i \le p$$

$$U_j = \left\{y_{j1}, y_{j2}, \dots, y_{jm_j}\right\} \qquad 0 \le j \le q$$

satisfying the following requirements:

i) if $I_n \neq 0$ then $U_n \neq \emptyset$;

ii) in order to have a consistent weight reference $I_o \neq \emptyset$ and $U_o \neq \emptyset$

$$\text{iii)} \quad \sum_{i=o}^{p} \sum_{j=1}^{m_i} x_{ij}\, 2^i = \sum_{j=o}^{p} \sum_{i=1}^{m_j} y_{ji}\, 2^j \qquad (1)$$

where the symbols x_{ij} and y_{ij} each indicate an output lead as well as the binary signal coming out from it.

Definition 2: The number i is the relative weight of an input $x_{ij} \in I_i$ or of an output $y_{ih} \in U_i$. Let $m_i = |I_i|$ the number of inputs of the set I_i, $n_j = |U_j|$ the number of outputs of the set U_j, $e_i \le m_i$ the number of inputs of the set I_i that are really excited by a binary signal, then we introduce:

/1/ An arithmetic unit is "triangular" if it has only one output for each weight.

/2/ The cells realizable with full-adders can be, of course, implemented in other ways to minimize the delay.

Definition 3: The Lead Set Characteristic Sequence (LSCS) of a BACC is the sequence:

$$m_p, m_{p-1}, \dots m_1, m_o$$

and

$$M(z) = \sum_{i=o}^{p} m_i z^i ,$$

the associated z-transform, is the Input Characteristic Polynomial (ICP).

The Output Characteristic Sequence (OCS) of a BACC is:

$$n_q, n_{q-1}, \dots, n_1, n_o$$

and

$$M(z) = \sum_{i=o}^{q} n_i z^i$$

is the output characteristic polynomial (OCP).

The Excitation Characteristic Sequence (ECS) is:

$$e_p, e_{p-1}, \dots, e_1, e_o$$

and

$$E(z) = \sum_{i=o}^{p} e_i z^i$$

is the Excitation Characteristic Polynomial (ECP).

Definition 4: An arithmetic network built up with BACC is regular, if the following conditions are verified:

3.1) each output can be connected to at most one input;

3.2) each input can be fed by at most one output;

3.3) if an output and an input are connected, they must have the same weight;

3.4) every input of the arithmetic network must affect those outputs of the network.

Definition 5: The Network Characteristic Sequence (NCS) is the sequence:

$$m_p - n_p, \dots, m_1 - n_1, n_o - p_o$$

and the associated z-polynomial:

$$\phi(z) = \sum_{i=o}^{p} (m_i - n_i) z^i$$

is named Network Characteristic Polynomial (N.C.P.).

From the above definitions, one gets:

$$\phi(z) = M(z) - N(z) \qquad (2)$$

The following propositions justify the introduction of $\phi(z)$.

Proposition 1:

i) if two arithmetic sub-networks, whose NCPs are respectively $\phi^{(1)}(z)$ and $\phi^{(2)}(z)$, are connected according to Definition 4, the resulting network has an NCP $\phi(z)$, as follows:

$$\phi(z) = \overset{(1)}{\phi}(z) + \overset{(2)}{\phi}(z) \qquad (3)$$

ii) multiplying the NCP $\phi(z)$ of a network by a positive integer p, a new z-polynomial is obtained, that is again the NCP of another network:

$$\phi(z) = p\,\phi(z) \qquad (4)$$

iii) Multiplying the NCP of a network by z^k, a new NCP is obtained that corresponds to a shift of the weight-reference by k positions.

Definition 6: Characteristic Sequence of a Regular System (CSRS) is the sequence of positive integers:

$$a_m, a_{m-1}, \dots, a_1, a_o$$

where a_j is the number of equal BACCs whose least significant weight is shifted by j positions with respect to the least significant weight of the whole network; the associated z-polynomial:

$$S(z) = \sum_{i=o}^{m} a_i z^i$$

is the System Characteristic Polynomial (SCP).

The BACC that compose the system are called cells or macrocells. Defining as $\varphi(z)$ the NCP of the cell, if $\phi(z)$ is the NCP of a regular arithmetic network, then from proposition 1 and definition 6 one gets:

$$\phi(z) = \varphi(z)\; S(z) \qquad (5)$$

Definition 7: A system is separable if its NCP $\phi(z)$ can be written as:

$$\phi(z) = z^1 Q(z) + P(z)$$

with the polynomials $\phi(z)$ and P(z) satisfying the following conditions:

$$P(z) \nleq 0$$

$$Q(z) \neq 0$$

and the degree of Q(z) is $\leq 1-1$.

Intuitively, a separable system is composed of two or more distinct sub-systems that can be studied independently.

Proposition 2: Let:

$$\phi(z) = \sum_{i=o}^{p} b_i z^i$$

and

$$\phi_h(z) = \sum_{i=o}^{h} b_i z^i \qquad 0 \leq h \leq p-1$$

the system is not separable if, and only if,

$$\phi_h(2) > 0$$

for every

$$1 \leq h \leq p-1$$

The proof comes directly from definition 7.

Definition 8: An Operatively Full System (OFS) is one that when all the inputs are excited all the outputs can vary.

For the OFSs, the (1) becomes:

$$\sum_{i=1}^{p} m_i 2^i = \sum_{i=o}^{q} n_i 2^i \qquad (6)$$

The eqn. (6) can be rewritten, on the basis of definition 2, as:

$$M(2) = N(2)$$

which, because of (2), implies that:

$$\not{o}(2) = 0 ;$$

therefore the NCP can be divided by (2-z), giving:

$$\not{o}(z) = T(z) (-z +2) \qquad (7)$$

An important property of $T(z)$, for the non-separable systems, is given by the following theorem:

Theorem 1: An arithmetic network is a non-separable OFS if and only if $\not{o}(z)$ can be written as in (7) and the coefficients of $T(z)$ are all positive numbers.

Proof:

Only If. The OFS can be written as in (7); it remains to prove that if OFS is not-separable, then $T(z)$ has every coefficient positive. Let:

$$T(z) = \sum_{i=o}^{p-1} t_i z^i$$

$$M_h = (\sum_{i=o}^{h} t_i z^i) (-z +2) =$$

$$= - t_h z^{h+1} + 2 t_h z^h + (-z +2) \sum_{i=o}^{h-1} t_i z^i ;$$

$\not{o}_h(z)$ is obtained from $M_h(z)$, neglecting the term with z^{h+1} :

$$\not{o}_h(z) = 2 t_h z^h + (-z +2) \sum_{i=o}^{h-1} t_i z^i$$

The proposition 2 imposes $\not{o}_h(2) > 0$ for every $h \leq p-1$; thus:

$$\not{o}_h(2) = 2 t_h 2^h > 0$$

and $t_h > 0$.

If.

If $\not{o}(z)$ can be written as in (7) the system is a OFS, if $t_h > 0$ for every $h \leq p-1$, then $\not{o}_h(2) > 0$, thus the OFS is not separable for proposition 2.

Corollary 1: Every regular (cellular), non-separable OFS can be implemented with only full-adders, that is with sub-systems whose NCP is:

$$\varphi(z) = - z + 2 \qquad (8)$$

Every cellular non separable OFS is composed of cells whose NCP $G(z)$ /3/ can be expressed by Corollary 1, as follows:

$$G(z) = s(z) (-z +2) \qquad (9)$$

The NCP of the whole network, because of (5), can be expressed as:

$$\not{o}(z) = S(z) G(z) = S(z) s(z) (-z +2) \qquad (10)$$

In the cellular arrays, for which (5) holds, setting $z=1$ in the polynomial $S(z)$ gives the number of cells in the arithmetic network.

III. SYNTHESIS PROCEDURES

The general problem of designing the cellular BACC with the minimum number of elementary blocks, each realizable with only full-adders, can be now set as follows.

Given $E(z)$, $N(z)$ compatible with it, that is

$$E(2) \leq N(2) ,$$

and $G(z)$ satisfying eqn. (9), find $S(z)$ such that the OFS, having $\not{o}(z)$ satisfying eqn. (10), will accept either $E(z)$ and $N(z)$. Furthermore, $S(1)$ must be the minimum.

In general, there is more than one solution to this problem and the resulting system can have an NCP that does not match exactly $E(z) - N(z)$, that is $m_i \geq e_i$ for some value of i. To obtain the minimum unknown sequence $S(z)$, the following procedure can be used:

Step 1: $G(z)$ is factorized as in eqn. (9)

Step 2: The NCP $\not{o}^*(z)$ that accepts exactly $E(z)$ and $N(z)$ is computed:

$$\not{o}^*(z) = E(z) - N(z) \qquad (11)$$

Step 3: If $\not{o}^*(z)$ can be factorized as in eqn. (7), the polynomial $R(z)$ is easily determined:

$$\not{o}^*(z) = R(z) (-z +2) \qquad (12)$$

otherwise another NCP, $\not{o}_1^*(z)$, factorizable, by $-z +2$, having all coefficients greater than or equal to $\not{o}^*(z)$, must be found:

$$\not{o}_1^*(z) = \not{o}^*(z) + K(z) = R(z) (-z +2) \qquad (12\ bis)$$

The polynomial $K(z)$ can be obtained in the following way /4/ :

3.1 $\not{o}^*(z)$ is evaluated with $z = 2$, giving:

$$\not{o}^*(2) = E(2) - N(2) < 0$$

/3/ It can be noted that $s(z)$ is the transform of the sequence of full adders that form the macrocell.

/4/ The proof that $K(z)$ computed as in Step 3 gives the polynomial $R(z)$ of the minimal solution is reported in Appendix.

3.2 The coefficients C_i of the positive binary number $-\emptyset^*(2)$ are found:

$$\sum_{i=0}^{p} C_i \, 2^i = -\emptyset^*(2) \qquad (13)$$

3.2 Finally K(z) is given by:

$$K(z) = \sum_{i=0}^{p} C_i \, z^i \qquad (14)$$

An important result at this step has been obtained; in fact the polynomial R(z), that in any case has been determined, indicates for each weight, the minimum number of full-adders necessary for implementing the BACC when the chosen cells are the same full-adders.

Step 4: In general $\emptyset(z)$ must include $\emptyset_1^*(z)$ or $\emptyset^*(z)$, that is, remembering eqns. (10), (12) and (12 bis), the polynomial S(z), solution of the problem, must satisfy the following relation:

$$S(z) \; s(z) = R(z) + H(z) \qquad (16)$$

where the unknown H(z) is a polynomial, having non-negative integer coefficients.

This equation shows that the minimization problem can be formulated as a positive integer linear programming problem:

Find S(z) such that:

i) the coefficients of the product S(z) s(z) are greater than, or equal to, the corresponding coefficients in the polynomial R(z);

ii) S(1) must be minimum.

H(z), that gives a measure, with the full-adder as unit, of the greater cost to implement the BACC with the macrocell than with the full-adder, in many interesting cases, it can be easily determined using the particular properties of M(z), N(z) and G(z).

S(z) is thus obtained by means of faster procedures of division between polynomial with integer coefficients. Furthermore, from eqn. (16), a lower bound to the minimal number N of cells can be obtained:

$$N = S(1) \geq \frac{R(1)}{s(1)}$$

It must be noted that also Meo [1] solved the problem using integer linear programming methods, but his formulation fails when cells having more than one output at the most significant weight are used. Eqn. (16), allows to obtain the right solution in any case; an example of cells that cannot be considered with the procedure of Meo is presented in the following paragraph.

IV. MACROCELLULAR MULTIPLIERS

In this paragraph the minimization methods are applied to implement full multipliers using a new family of macrocells introduced in a previous paper [2]. For what is interesting here, where the generation of the partial product bits are not considered, the single cell can be represented as in fig. 1.

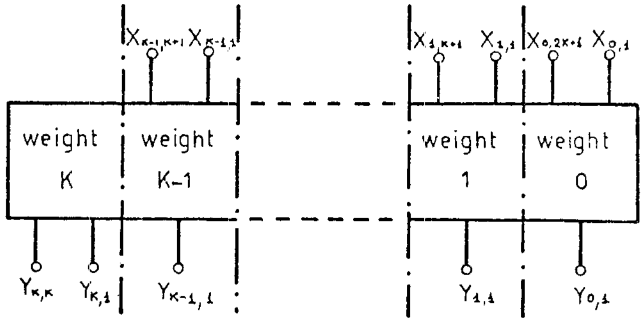

Fig. 1 - Non Triangular Operative Full cell with NCS (-K, k, ..., K, 2K).

It has no input at the most significant weight, 2k+1 inputs at the least significant weight and k+1 at all the others. The cell has, moreover, k outputs at the most significant weight and one output at all the others. As a result it is a OFS. Its NCP G(z) can be written as:

$$G(z) = -kz^k + \sum_{j=1}^{k-1} kz^j + 2^k$$

and can be, furthermore, factorized for corollary 1, giving:

$$G(z) = k \left(\sum_{j=0}^{k-1} z^j\right) (-z+2) \qquad (17)$$

For the sake of simplicity an M by M bits full multiplier will be considered. The full multiplier is an arithmetic arrays having two M bits multiplicative inputs W and Z, two M bits additive inputs H and K, and 2 N bits output Q, related as in the following formula:

$$Q = W \, . \, Z + H + K \qquad (18)$$

The full multiplier can be used also to implement the product of numbers represented in 2's complement form, using the algorithm and the correcting networks proposed in another paper [4]. The minimum number of excitation leads in a full-multiplier is the one corresponding to the following ECP:

$$E(z) = \sum_{j=0}^{M-2} (j+1) \, z^{2M-j-2} + M \, z^{M-1} + \sum_{j=0}^{M-2} (j+1) \, z^j + 2\sum_{j=0}^{M-1} z^j$$

Furthermore, there has to be only one output for each weight; so the OCP of the network will be given by:

$$N(z) = \sum_{j=0}^{2M-1} z^j \qquad (19)$$

For the NCP, one gets:

$$\emptyset(z) = E(z) - N(z) = -z^{2M-1} + \sum_{j=0}^{M-2} j \, z^{2M-j-2} + (M+1) \, z^{M-1} + \sum_{j=0}^{M-2} (j+2) \, z^j$$

because $\emptyset(2) = 0$, for theorem 1, the following factorisation is obtained:

$$\phi(z) = (-z + 2) \sum_{j=o}^{M-2} (j+1)\ z^{2M-j-2} + M\ z^{M-1} + \sum_{j=o}^{M-2} (j+1)\ z^{j} \tag{20}$$

comparison with eqns. (12) gives:

$$R(z) = \sum_{j=o}^{M-2} (j+1)\ z^{2M-j-2} + M\ z^{M-1} + \sum_{j=o}^{M-2} (j+1)\ z^{j} \tag{21}$$

Now the z-polynomial $H(z)$ has to be found, such that:

$$R(z) + H(z) = S(z)\ R\left[\sum_{j=o}^{k-1} z^{j}\right] \tag{22}$$

with the condition that $S(1)$ is minimum.

In order to solve the problem, the following points have to be considered.

1) If $S(1)$ is minimum, also $H(1)$ is minimum, because:

$$R(1) + H(1) = S(1)\ k^2,$$

hence:

$$H(1) = S(1)\ k^2 - R(1)\ .$$

2) All the coefficients of $R(z) + H(z)$ must be multiples of k, therefore, $H(z)$ can be rewritten as:

$$H(z) = H_1(z) + k\ H_2(z)\ ,$$

where $H_1(z)$ is such that $H_1(z)$ is minimum and $R(z) + H_1(z)$ has all the coefficients multiples of k.

3) Let $M = m\,k + n$ $(0 \leq n \leq k - 1)$; in order to determine $H_1(z)$, three cases are of interest.

3a) $\underline{n = 0}$.

$$R(z) = \sum_{j=o}^{k-1} (j+1)\ z^{2M-j-2} + \sum_{j=k}^{2k-1} (j+1)\ z^{2M-j-2} + \ldots + \sum_{j=(m-1)k}^{m\,k-1} (j+1)\ z^{2M-j-2} + \sum_{j=o}^{k-1} (j+1)\ z^{j} + \ldots + \sum_{j=(m-1)k}^{m\,k-2} (j+1)\ z^{j}\ .$$

In this case $H_1(z)$ can be easily determined; it corresponds to the minimal "filling" of $R(z)$, that makes $R(z) + H_1(z)$ divisible by k, leading to:

$$R(z) + H_1(z) = \sum_{j=o}^{k-1} k\ z^{2M-j-2} + \sum_{j=k}^{2k-1} 2\,k\ z^{2M-j-2} + \ldots + \sum_{j=(m-1)k}^{k\,m-1} m\,k\,z^{2M-j-2} + \sum_{j=o}^{k-1} k\ z^{j} + \ldots + \sum_{j=(m-1)k}^{mk-2} k\ m\ z^{j} =$$

$$= k\left\{\left(\sum_{j=o}^{k-1} z^{j}\right)\left[z^{2M-k-1} + 2\ z^{2m-2k-1} + \ldots + m\ z^{2M-m\,k-1} + 1 + 2\,z^{k} + \ldots + (m-1)\ z^{(m-2)\,k}\right] + m \sum_{j=(m-1)k}^{m\,k-2} z^{j}\right\}$$

3b) $\underline{n = 1}$.

$$R(z) = \sum_{j=o}^{k-1} (j+1)\ z^{2M-j-2} + \ldots + \sum_{j=(m-1)k}^{m\,k-1} (j+1)\ z^{2M-j-2} + (m\,k+1)\ z^{2M-m\,k-2} + \sum_{j=o}^{k-1} (j+1)\ z^{j} + \ldots + \sum_{j=(m-1)k}^{m\,k-1} (j+1)\ z^{j}\ .$$

As for case 3a), one gets:

$$R(z) + H_1(z) = k \sum_{j=o}^{k-1} z^{2M-j-2} + \ldots + m\,k \sum_{j=(m-1)k}^{m\,k-1} z^{2M-j-2} + (m+1)\,k\,z^{2M-m\,k-2} + k \sum_{j=o}^{k-1} z^{j} + \ldots + m\,k \sum_{j=(m-1)k}^{m\,k-1} z^{j} =$$

$$= k\left\{\left(\sum_{j=o}^{k-1} z^{j}\right)\left[z^{2M-2k-1} + \ldots + m\ z^{2M-km-1} + 1 + 2\ z^{k} + \ldots + m\ z^{(m-1)\,k}\right] + (m+1)\ z^{2M-mk-2}\right\}.$$

3c) $\underline{n \geq 2}$.

Analogously with the previous cases, one gets:

$$R(z) + H_1(z) = k\left\{\left(\sum_{j=o}^{k-1} z^{j}\right)\left[z^{2M-k-1} + 2\ z^{2M-2k-1} + \ldots + m\ z^{2M-km-1} + m\ z^{(m-1)k} + \ldots + 2\ z^{k} + 1\right] + \sum_{j=mk}^{mk+n-1} (m+1)\ z^{2M-j-2} + \sum_{j=mk}^{mk+n-2} (m+1)\ z^{j}\right\}.$$

4) Using the results obtained above, the z-polynomial $H_2(z)$ can be determined in order to make $R(z) + H(z)$ divisible by the NCP of the cells given in (17); the following cases are of interest.

4a) $\underline{n = 0}$.

It must hold:

$$\left(\sum_{j=o}^{k-1} z^{j}\right)\left[z^{2M-k-1} + 2\ z^{2M.2k-1} + \ldots + m\ z^{2M-mk-1} + (m-1)\ z^{(m-2)k} + \ldots + 1\right] + m \sum_{j=(m-1)k}^{m\,k-2} z^{j} + H_2(z) = S(z)\ \left(\sum_{j=o}^{k-1} z^{j}\right)\ ;$$

it turns out that there are two equivalent solutions for $H_2(z)$, namely

$$H_2^1(z) = m\ z^{mk-1}$$

$$H_2^2(z) = m\ z^{(m-1)\ k-1}\ ;$$

selecting the former, one gets :

$$S(z) = z^{2M-k-1} + 2\ z^{2M-2k-1} + \ldots + m\ z^{mk-1} + m\ z^{(m-1)k} + (m-1)\ z^{(m-2)k} + \ldots + 1\ ,$$

$$S(1) = 1 + 2 + \ldots + m + m + \ldots + 2 + 1 = m\ (m+1)\ .$$

4b) $n \neq 0$

Assuming that a summation has zero value if the upper limit is less then the lower, $H_2(z)$ is determined from the condition:

$$\left(\sum_{j=o}^{k-1} z^j\right)\left[z^{2M-k-1} + 2\, z^{2M-2k-1} + \ldots + m\, z^{2M-km-1} + m\, z^{(m-1)k} + \ldots + 2\, z^k + 1\right] + \sum_{j=mk}^{mk+n-1} (m+1)\, z^{2M-j-2} + \sum_{j=mk}^{mk+n-2} (m+1)\, z^j + H_2(z) = S(z)\left(\sum_{j=o}^{k-1} z^j\right);$$

$H_2(z)$ must be such that:

$$(m+1)\sum_{j=mk+2n-2}^{mk+n-1} z^j + (m+1)\sum_{j=mk}^{mk+n-2} z^j + H_2(z) = q(z)\left(\sum_{j=o}^{k-1} z^j\right);$$

$$(m+1)\sum_{j=mk}^{mk+2n-2} z^j + H_2(z) = q(z)\left(\sum_{j=o}^{k-1} z^j\right);$$

$q(z)$ is an unknown polynomial.

Two sub-cases are of interest:

i) $2n - 2 + 1 \leq k$
ii) $2n - 2 + 1 > k$.

For sub-case i), $n \leq \frac{k+1}{2}$, the summation $\sum_{j=m\,k}^{mk+2n-2} z^j$ has less than k terms, then $H_2(z)$ must extend the summation to k terms, leading to:

$$q(z) = (m+1)\, z^{m\,k},$$

$$S(z) = z^{2M-k-1} + \ldots + m\, z^{2M-km-1} + (m+1)\, z^{m\,k} + m\, z^{(m-1)\,k} + \ldots + 1$$

$$S(1) = 1 + \ldots + m + (m+1) + m + \ldots + 1 = m\,(m+1) + m + 1 = (m+1)^2 .$$

For sub-case ii), $n > \frac{k+1}{2}$, $\sum_{j=m\,k}^{mk+2n-2} z^j$ has more than k terms and it has to be completed by $H_2(z)$ in order to have $2k$ terms; thus:

$$(m+1)\, z^{m\,k}\sum_{j=o}^{2n-2} z^j + H_2(z) = (m+1)\, z^{m\,k}\sum_{j=o}^{2k-1} z^j = (m+1)\, z^{m\,k}\,(z^k+1)\left(\sum_{j=o}^{k-1} z^j\right),$$

$$q(z) = (m+1)\, z^{m\,k}(z^k+1),$$

$$S(z) = z^{2M-k-1} + \ldots + m\, z^{2M-km-1} + (m+1)\, z^{(m+1)k} + (m+1)\, z^{m\,k} + m\, z^{(m-1)\,k} + \ldots + 1,$$

$$S(1) = 1 + 2 + \ldots + m + (m+1) + (m+1) + m + \ldots + 1 = (m+1)\,(m+2) .$$

Structure corresponding to the minimum number of components found above can be arranged as iterative arrays of macrocells. The details of such structures, as well as the extension to signed numbers, a discussion on cell implementation, taking into account the partial products generation, and the delay analysis are reported in another paper [2].

V. FAST NETWORKS FOR PERFORMING CONVOLUTIONS

The proposed minimization procedure can be also applied to any BACC, such that its output can be obtained as a sum of positive partial products. As an example, it is possible to implement a fast network to compute the following sum of products:

$$S = \sum_{i=1}^{N} X_i\, Y_i \qquad (23)$$

To increase the operation speed two techniques can be used: pipelining and paralleling of operations. The latter is considered first: three products are performed at the same time. Let:

$$Z_1 = X_{31}\, Y_{31} + X_{31-1}\, Y_{31-1} + X_{31-2}\, Y_{31-2} \qquad 1 = 1,\ldots N/3$$

eqn. (23) can be written as:

$$S = \sum_{1=1}^{N/3} Z_1 \qquad (24);$$

where N is supposed to be a multiple of 3 for simplicity.

Furthermore, a recursive procedure can be alternatively used:

$$S_1 = S_{1-1} + Z_1 \qquad 1 = 1, \ldots, N/3$$
$$S = S_{N/3} \qquad (25)$$

with the initial condition:

$$S_o = 0$$

Multiplication Algorithm

The multiplication algorithm must be chosen according to the representation of the factors.

In the following, the eqns. (22), (23) and (24) are used for computing the output of a linear filter (recursive or not). Its coefficients X_i are fixed, thus they can be represented in sign and magnitude, while Y_i, the samples that can be obtained from other computation, are 2's-complement binary fractions. The result of the multiplication must be also in 2's-complement form, to compute Z_1. The algorithms that consider negative factors, like Booth's method and Robertson's second method, must be rejected, because the partial products matrix becomes larger than in the positive case. Furthermore, also the algorithms for 2's complement parallel multipliers [4,5] give, in this case, solutions more expensive than the special procedure developed here.

							$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$	$\tilde{x}_4$	$\tilde{x}_5$	$\tilde{x}_6$
							z_1	z_2	z_3	z_4	z_5	z_6
							$\tilde{x}_1 z_6$	$\tilde{x}_2 z_6$	$\tilde{x}_3 z_6$	$\tilde{x}_4 z_6$	$\tilde{x}_5 z_6$	$\tilde{x}_6 z_6$
						$\tilde{x}_1 z_5$	$\tilde{x}_2 z_5$	$\tilde{x}_3 z_5$	$\tilde{x}_4 z_5$	$\tilde{x}_5 z_5$	$\tilde{x}_6 z_5$	
					$\tilde{x}_1 z_4$	$\tilde{x}_2 z_4$	$\tilde{x}_3 z_4$	$\tilde{x}_4 z_4$	$\tilde{x}_5 z_4$	$\tilde{x}_6 z_4$		
				$\tilde{x}_1 z_3$	$\tilde{x}_2 z_3$	$\tilde{x}_3 z_3$	$\tilde{x}_4 z_3$	$\tilde{x}_5 z_3$	$\tilde{x}_6 z_3$			
			$\tilde{x}_1 z_2$	$\tilde{x}_2 z_2$	$\tilde{x}_3 z_2$	$\tilde{x}_4 z_2$	$\tilde{x}_5 z_2$	$\tilde{x}_6 z_2$				
		$\tilde{x}_1 z_1$	$\tilde{x}_2 z_1$	$\tilde{x}_3 z_1$	$\tilde{x}_4 z_1$	$\tilde{x}_5 z_1$	$\tilde{x}_6 z_1$					
	$\bar{\tilde{x}}_1 u_0$	$\bar{\tilde{x}}_2 u_0$	$\bar{\tilde{x}}_3 u_0$	$\bar{\tilde{x}}_4 u_0$	$\bar{\tilde{x}}_5 u_0$	$\bar{\tilde{x}}_6 u_0$	$\tilde{x}_1 x_0$	$\tilde{x}_2 x_0$	$\tilde{x}_3 x_0$	$\tilde{x}_4 x_0$	$\tilde{x}_5 x_0$	$\tilde{x}_6 x_0$
						u_0						
p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}

Fig. 2 – Positive partial products matrix for $M = 6$.

Let $(x_0, x_1, \ldots, x_M)$ and $(u_0, y_1, \ldots, y_M)$ be the representation, in 2' complement form of the generic X and Y, whose value is given by:

$$X = -x_0 + \sum_{j=1}^{M} x_j 2^{-j} = -x_0 + X^* \qquad (26)$$

$$Y = -y_0 + \sum_{j=1}^{M} y_j 2^{-j} = -y_0 + Y^*$$

where x_0 and y_0 are the sign bits (0 if positive, 1 if negative), and X^*, Y^* the fractional parts.

The representation of X in sign magnitude form is $(x_0, \tilde{X}^*)$, where $\tilde{X}^*$ indicates the magnitude.

Between $\tilde{X}^*$ and X^* there is the relation:

$$X^* = \tilde{X}^* \quad \text{if } x_0 = 0$$

$$X^* = 1 - \tilde{X}^* \quad \text{if } x_0 = 1 \qquad (27)$$

The true value of the product P can be given by:

$$P = XY = X^*Y^* - x_0 Y^* - y_0 X^* + x_0 y_0 \qquad (28)$$

The following representation of P, as binary 2'complement fraction, must be found:

$$P = -p_0 + \sum_{j=1}^{2M} p_j 2^{-j}$$

Two different cases are considered.

Case 1: X positive: Using X^* and Y^* as factors, the M by M bit partial products matrix is obtained; by means of (27), its value Q is:

$$Q = \tilde{X}^* Y^* = X^* Y^*$$

A corrective term C must be considered:

$$C = P - Q = -y_0 X^* = -y_0 \tilde{X}^* \qquad (29)$$

Case 2: X negative: $\tilde{X}^*$ and $\bar{Y}^*$, the 1's complement of Y^*, are multiplied. Remembering that:

$$\bar{Y}^* = 1 - Y^* - 2^{-M} \qquad (30)$$

A can be obtained:

$$Q = \tilde{X}^* \bar{Y}^* = 1 - X^* - Y^* + X^* Y^* - (1 - X^*) 2^{-M} =$$

$$= X^* Y^* - Y^* - 2^{-M} \tilde{X}^* + \tilde{X}^* \qquad (31)$$

In this case using eqns. (27), (28) and (31) and the relation:

$$y_0 - 1 = -\bar{y}_0$$

C becomes :

$$C = y_0 (1 - X^*) - \tilde{X}^* + 2^{-M} \tilde{X}^* =$$

$$= 2^{-M} \tilde{X}^* - y_0 \tilde{X}^* \qquad (32)$$

Since:

$$-\tilde{X}^* = \bar{\tilde{X}}^* + 2^{-M}$$

The eqns. (29) and (32) can be summarized in one formula only:

$$C = 2^{-M} x_0 \tilde{X}^* + (x_0 \oplus y_0)(\bar{\tilde{X}}^* + 2^{-M}) \qquad (33)$$

thus the true value of the product P is:

$$P = Q + C = \tilde{X}^* Z^* + 2^{-M} x_0 \tilde{X}^* + u_0 \bar{\tilde{X}}^* + 2^{-M} u_0 \qquad (34)$$

with:

$$u_0 = x_0 \oplus y_0$$

$$z_i = \overline{x_i \oplus y_i} \qquad i = 1, \ldots, M$$

Fig. 2 shows an example with $M = 6$ of the all positive partial products matrix, for the multiplication algorithm proposed. As can be seen, the corrective terms are one bit of weight 2^{-6} and one 12 bit binary fraction.

In the following, we will consider $M = 9$. The procedure of eqn. (25) is used to show how it can be applied to the minimization method in a more difficult case than eqn. (24). When, otherwise, the speed of operation is the most important parameter eqn. (24) allows one to use standard pipelining techniques, because it is not recursive. The sum of eqn. (24) is executed "modulo 8", to take into account a possible ra ge of the result. Under the previous assumption the ECS is:

(1,1,1,4,7,10,13,16,19,22,25,31,31,28,25,22,19,16,13,,10,7) ,

while the OCS is:

(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1) .

Referring to step 2 of the procedure of minimization, the sequence of coefficients of $\emptyset^*(z)$ is:

(-1,0,0,0,3,6,9,12,15,18,21,24,30,30,27,24,21,18,15,,12,9,6) .

The correcting polynomial K(z) is obtained as follows:

$$N(2) - E(2) = 2^{19} + 2^{10} + 2^{9}$$

and so:

$$K(z) = z^{19} + z^{10} + z^{9}$$

The polynomial $\emptyset^*_1(z)$ has the sequence:

(-1,0,1,0,3,6,9,12,15,18,21,25,31,30,27,24,21,18,15,,12,9,6) ;

and that associated with R(z) becomes:

(1,2,3,6,9,12,15,18,21,24,27,29,24,21,18,15,12,9,6,3) .

Realizing the network with the full-adder as cell, the minimum number requested is R(1) = 282, less than using 3 multipliers for 2' complement numbers and 2 adders.

The chosen cell is one of the family presented in part IV, because k equals 3, its NCP G(z) can be written as:

$$G(z) = -3z^3 + 3z^2 + 3z + 6 = 3\ (z^2 + z + 1)\ (-z + 2).$$

Two possible realizations can be obtained with binary trees of carry lookahead adders.

Referring to step 4, the following relation must be satisfied:

$$3\ S(z)\ (z^2 + z + 1) = R(z) + H(z)$$

thus H(z) can be written as follows:

$$H(z) = 3\ H_1(z) + H_2(z)$$

where $H_2(z)$ is the polynomial that must be added to R(z) to give a polynomial with all coefficients multiples of three.

Because the polynomial R(z) + H(z) is divisible by $z^2 + z + 1$, $H_1(z) \equiv 0$ and so the NCS of the network performing eqn. (24)[4] is:

$$S_1 = (1,0,0,2,1,1,3,2,2,4,3,3,3,2,2,2,1,1,1)$$

and the minimum number of cells is (34).

Remembering that G(1), equal to 9, gives for every cell the equivalent number of full-adders, the number of full-adders, wasted in S_1, is 24. A look at S(z) suggests that these structures cannot be very fast, because there are not zeros in the sequence of coefficients. So another network, faster than the former, is found, forcing to zero two successive coefficients. The resulting NCS is:

$$S_2 = (1,0,0,4,0,0,7,0,0,10,0,0,9,0,0,6,0,0,3)\ .$$

The number of cells in this case is 40. If the cell can also generate the sequence (3,3,3) of partial product bits, as in ref. [2], an iterative structures can be obtained from the sequence of cells S_2. In this case can be shown that the maximum delay of the structure is approximately $\frac{2M}{3}\tau$, if the carry delay of the lookahead adders is equal to the sum delay τ and $M \gg 3$. The time necessary to compute S of eqn. (24) is thus:

$$T_1 = \frac{N}{3}\ \frac{2M}{3}\ \tau$$

latching all furthermore the outputs of the macrocell as shown in fig. 3 at every clock six factors can be

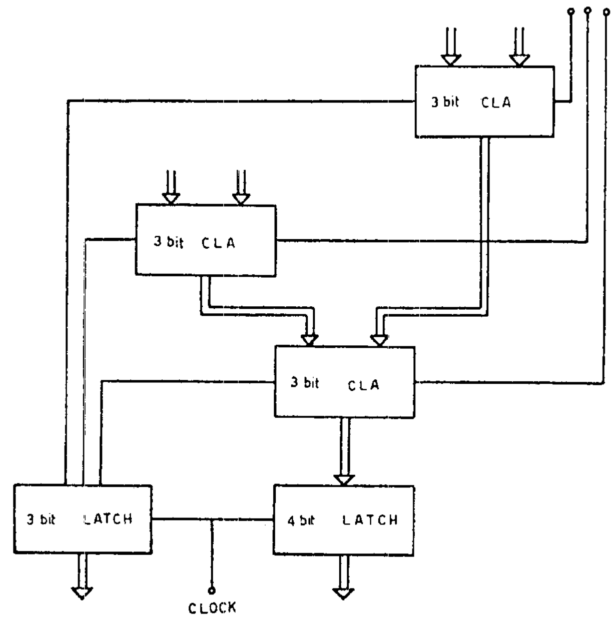

Fig. 3 - Cell implementation.

presented at the input of the structures. The rate of the clock can be approximatively $\frac{1}{2\tau}$ because of the cell implementation. After $\frac{N}{3}$ clocks the final sum of the partial results stored into the latch is obtained, by-passing the registers. The time for computing S in this case is

$$T_2 = \frac{N}{3}\ 2\tau + \tau_a$$

where τ_a is the delay time of the array.

CONCLUSIONS

A new methods to study arithmetic combinatorial networks has been presented. It gives many interesting results in analysis and synthesis, allowing to obtaining the minimum number of modules in cellular arrays using in general linear programming techniques or, in many interesting cases, faster procedures. The cell must be "operative full", that is they must be implemented with only full adders. No other restriction are imposed. An extension to the most general class of networks is reported in ref. [3]. Application of the minimization methods to macrocellular multipliers and to fast networks for performing convolutions are also presented.

ACKNOWLEDGMENTS

This work was carried out at the Centro di Elaborazione Numerale dei Segnali and was supported by the Consiglio Nazionale delle Ricerche of Italy.

The authors are grateful to Profrs. R. Sartori and A. R. Meo for the useful discussions and suggestions.

REFERENCES

[1] A. R. Meo, "Arithmetic Networks and their Minimization Using a New Line of Elementary Units", IEEE Trans. Comput., vol. C-24, pp. 258-280, March 1975.

[2] R. De Mori, M. Elia and A. Serra, "Macrocellular Structures for Parallel Multipliers", to be published on IEEE Trans. Comput.

[3] M. Elia and A. Serra, "Analysis and Synthesis of Arithmetic Combinatorial Circuits Using Generating Functions", CENS Tech. Rep., June 1975, Turin, Italy.

[4] R. De Mori and A. Serra, "A Parallel Structure for Signed-Numbers Multiplication and Addition", IEEE Trans. Comput., vol. C-21, pp. 1453-1455, December 1972.

[5] C. R. Baugh and B. A. Wooley, "A Two's Complement Parallel Array Multiplication Algorithm", IEEE Trans. Comput, vol. C-22, pp. 1045-1047, December 1973.

APPENDIX

The following theorem must be proved, when eqn. (12) is not satisfied:

The polynomial $K(z)$, evaluated by means of eqns. (13) and (14), gives a polynomial $R(z)$, satisfying eqn. (12 bis), and such that $R(1)$ represents the minimal number of full-adders necessary to implement the wanted BACC, when the full-adder is used as elementary unit.

Proof:

Dividing $\phi^*(z)$ of eqn. (11) by $(-z+2)$, a negative remainder $-B$ is obtained, so the following relation can be written:

$$\phi^*(z) = P(z)\,(-z+2) - B \qquad (A1)$$

with

$$B = -\phi^*(2) = N(2) - E(2) > 0$$

Let us consider two polymonials $\phi^*_{11}(z)$ and $\phi^*_{12}(z)$, factorizable by $-z+2$, and obtained by $\phi^*(z)$ adding two polymonials $K_1(z)$ and $K_2(z)$:

$$\phi^*_{11}(z) = \phi^*(z) + K_1(z) = R_1(z)\,(-z+2) \qquad (A2)$$

$$\phi^*_{12}(z) = \phi^*(z) + K_2(z) = R_2(z)\,(-z+2) \qquad (A3)$$

where:

$$K_1(z) = \sum_{i=1}^{m_1} K_{1i}\, z^i$$

$$K_2(z) = \sum_{i=1}^{m_2} K_{2i}\, z^i$$

$$K_1(1) = K_2(1) = B > 0$$

because of eqns. (A1), (A2) and (A3).

Subtracting eqn. (A2) from eqn. (A3), one can obtain :

$$\phi^*_{11}(z) - \phi^*_{12}(z) = K_1(z) - K_2(z) = \left[R_1(z) - R_2(z)\right](-z+2) \qquad (A4)$$

The evaluation of eqn. (A4) with z equal to 3, gives:

$$R_1(1) - R_2(1) = K_1(1) - K_2(1) \qquad (A5)$$

This relationship shows that the determination of the minimum $R(1)$ is equivalent to the determination of the minimum $K(1)$.

Remembering that:

$$K_j(1) = \sum_{i=1}^{m_j} K_i$$

$$K_j(2) = \sum_{i=1}^{m_j} k_i\, 2^i$$

between all the possible representations $K_j(2)$ of B the one that gives the minimum, $K_m(2)$ $K_j(1)$ is the binary representation of B

$$K_m(2) = B = -\phi^*(2) = \sum c_i\, 2^i \qquad (A6)$$

thus $K_m(z)$ satifies eqn. (14).