A SYSTEMATIC APPROACH FOR DESIGNING SYSTOLIC ARRAYS

C. N. Zhang A. G. Law and A. Rezazadeh

Department of Computer Science
University of Regina
Regina, Sask. Canada S4S 0A2

Abstract

In this paper we have show that the problems of determining the existence of a valid transformation and finding an optimal valid transformation (if it exists) for a given nested loop algorithm can be computed easily. Our strategy exploits restricted row operation and normal form of integer matrix as well as the concept of the generalized inverse of a non-square matrix. In particular, three procedures, corresponding to three different rank values of the given dependency matrix, are proposed.

1. Introduction

New technologies such as VLSI and Wafer Scale Integration (WSI) have made possible the manufacturing of systolic arrays [1-4]. Systolic arrays have been used in diverse applications such as signal processing, matrix inversion and multiplication [1-2] and the trend to develop larger systems is expected to continue, because of the increasing demand for computation power. The first and foremost problem in designing such arrays is the mapping of algorithms into systolic architectures. A frequent structure, nested do-loops, can be characterized by a dependency matrix and a computation space which represents the uniform data flow employed within the abstract computation space. Space-time mapping is an established technique for transforming such a computation into a systolic implementation [5-6]. In order for a computation structure to be implementable in a systolic array, the conceptual computation space $C_D = \{(i,j,k)\}$ must be mapped into $C_S = \{(t,x,y)\}$, where t denotes time and (x,y) represents the 2-D space normally available for VLSI systolic array embedding. If a computation structure is characterized by a constant dependency matrix consisting of a set of constant dependency vectors, the structure can be mapped into a systolic implementation with a new dependency by means of a matrix multiplication. As we know, in order to map an continuous integer space $\{(i,j,k)\}$ into an integer space $\{(t,x,y)\}$ the transformation must be an integer matrix. Theoretically, the problem of finding the integer solutions of the valid transformations for a given algorithm represented by matrix D and C_D can be formulated as an integer linear programming problem. Based on this observation, several mapping approaches have been proposed [7-16]. However, for practical applications, there are two fundamental problems still left to be tackled: (i) how to determine the existence of valid transformation matrices for a

0-8186-2610-0/92 $3.00 © 1992 IEEE

given computation, and (ii) how to derive such valid transformations efficiently. In order to solve these questions, the concepts of the restricted row normal form and generalized inverses of a non-square matrix and related results are introduced. The restricted row normal form is unique and can be used to eliminate algorithms that cannot be made systolic as well as make the search space reasonably small. We show that the problem of finding a valid transformation for a given algorithm represented by a dependency matrix D can be formalized into a matrix equation which can by solved by means of generalized inverse machinery of a non-square matrix. To specify an optimal solution for a systolic array implementation, several important parameters of the systolic implementations are precisely defined or redefined such as the computation time or the computational space in which systolic array computations are taking place [7,16]. We show that all these parameters depend on the transformation and the mapping algorithm (dependency matrix, sizes of the loops) and can be calculated easily. Based on the theoretical results, three procedures, corresponding to three different rank values of the matrix D, are proposed. A software package, based on the three procedures, which accepts the matrix D and loop sizes as inputs is being implemented in the C language.

2. Mapping Nested Loops into Systolic Arrays

In presenting our results, we confine our attention to three-dimensional nested loop algorithms so that we obtain two-dimensional systolic arrays in the realization. However, the results are equally applicable to n-dimensional computations and the transformed dependency matrices would correspond to (n-1)-cube systolic arrays. A three dimensional nested loop algorithm (suppose there are n data dependences) can be represented a tuple pair, (D, C_D), in which C_D is the index space {(i,j,k)} where the data are computed or used, and $D = \begin{bmatrix} d_{11} & d_{12} & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ d_{31} & d_{32} & d_{3n} \end{bmatrix}$ is a matrix which characterizes the data dependency relationship among the variables [5,6]. In order for a given nested loop algorithm to be implemented in a 2-D systolic array, an integer matrix T in $C^{3\times3}$ should be selected such that the product T D = Δ becomes a systolic matrix, as defined below. Such a matrix T is called a valid transformation of (D, C_D).

Definition 1. A matrix T in $C^{3\times3}$ is a valid transformation of (D, C_D) if, and only if,
i. TD = Δ is a systolic matrix and
ii. $|T| \neq 0$ (T is a one to one mapping).

Definition 2. A matrix Δ in $C^{3\times n}$ is a systolic matrix if, and only if, $\delta_{1j} < 0$, $\delta_{2j} \in \{-1, 0, 1\}$ and
$\delta_{3j} \in \{-1, 0, 1\}$, but there is at least j such that $\delta_{2j} \neq 0$ and $\delta_{3j} \neq 0$, where $\Delta = \begin{bmatrix} \delta_{11} & \delta_{12} & \delta_{1n} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2n} \\ \delta_{31} & \delta_{32} & \delta_{3n} \end{bmatrix}$.

For each vector $\begin{bmatrix} \delta_{1j} \\ \delta_{2j} \\ \delta_{3j} \end{bmatrix}$ in the matrix Δ, which corresponds a vector $\begin{bmatrix} d_{1j} \\ d_{2j} \\ d_{3j} \end{bmatrix}$ in the matrix D, represents a new data dependency on the t-x-y space where t is time and x-y is a plan in which the systolic array is embedded. The negative value of δ_{1j} is required to

enforce the proper time schedule on the systolic array (before starting the current computation, all its inputs must be prepared or precomputed). $\delta_{1j} = -1$ represents that there is a direct link connected a PE with one of its neighbors whose location is indicated by the δ_{2j} (on the x-coordinate direction) and δ_{3j} (on the y-coordinate direction). If $\delta_{1j} = -a$ ($a \neq 1$), then there are (a-1) time delay units (buffers) required to be built on the link. In practice, the maximum member of such time delay units is limited. In this paper, we assume that all $|\delta_{1}j| < B$ where B is a constant. $\delta_{2j} \in \{-1, 0, 1\}$ and $\delta_{3j} \in \{-1, 0, 1\}$ are required to ensure that only nearest neighbor connections are allowed in systolic arrays.

For an example, a nested loop algorithm given in [7] is:

```
for i := 1 to n do
for j := 1 to n do
for k := 1 to n do
    begin
      a(i,j,k) := a(i-1,j+1,k) b(i-1,j,k+1) ;
      b(i,j,k) := b(i-1,j-1,k+2) + b(i,j-3,k+2) ;
    end.
```

There are two variables (a and b) and four data dependencies a(i-1,j+1,k), b(i-1,j,k+1), b(i-1,j-1,k+2) and b(i,j-3,k+2) represented by $D = \begin{bmatrix} -1 & -1 & -1 & 0 \\ 1 & 0 & -1 & -3 \\ 0 & 1 & 2 & 2 \end{bmatrix}$. The computation space of the algorithm is defined by $C_D = \{ (1,1,1), (1,1,2),..., (n,n,n) \}$. If we choose $T = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$, then we have $\Delta = \begin{bmatrix} -1 & -2 & -3 & -2 \\ 0 & 0 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$. Since $|T| \neq 0$ and $\Delta = T D$ is a systolic matrix, T is a valid transformation. In [7], a software tool called ADVIS to derive such a valid transformation T was proposed. The ideas underlying ADVIS can be described as follows:

(i) Enumerate all possible integer matrices $T = (t_{ij})$ in $C^{3\times3}$ and examine whether each such matrix T makes T D a systolic matrix Δ. If it does, then add this matrix into set { T }. To limited the search space, this approach restricts that the absolute values of all elements of the matrix T are less than some constant.

(ii) If the set { T } is empty then no valid transformation exist for the given algorithm represented by D. Otherwise find a best one among the elements of the set.

For example, under the constraint $|t_{ij}| \leq 1$, ADVIS found one valid transformation and if let $|t_{ij}| \leq 3$, then there total thirteen solutions. However, from both theoretical studies and actual applications there are still some important tacks remaining.

1. First, how to determine whether there is a valid transformation for a given algorithm.
It is possible, for a certain algorithm, that ADVIS finds no valid transformations for the algorithm under the constraint $|t_{ij}| \leq c$, but in fact, there may be one or more valid transformations if a large value of c is allowed.
2. Second, how to derive the valid transformation efficiently.
The exhaustive search used in ADVIS is acceptable if the constraint constant c ($|t_{ij}| \leq c$) is small. Since this is not always the case, an alternative approach to find an optimal solution (if it exists) in a reasonable time should be sought.

Before addressing these two questions, we introduce some more definitions, notation and

results. Suppose T is a valid transformation mapping an algorithm represented by (D, C_D) into a systolic array implementation represented by (Δ, C_S), where

$$\begin{bmatrix} t \\ x \\ y \end{bmatrix} = T \begin{bmatrix} i \\ j \\ k \end{bmatrix}, \quad (1)$$

C_D = { (i,j,k) } and C_D = { (t,x,y) }. Let T = $\begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$. We have

$$t = t_{11}i + t_{12}j + t_{13}k, \quad (2)$$

$$x = t_{21}i + t_{22}j + t_{23}k \text{ and} \quad (3)$$

$$y = t_{31}i + t_{32}j + t_{33}k \quad (4)$$

Definition 3. The computation time required by the systolic array implementation (Δ, C_S) obtained by T is defined as t_c = max { t } - min { t } + 1 for all possible values t defined in (2).

Definition 4. The systolic array space required by the systolic array implementation (Δ, C_S) obtained by the transformation T is defined as s_c = (max { x } - min { x } + 1) × (max { y } - min { y } + 1) for all possible values x and y defined in (3) and (4).

According to the above definitions and the equation (1), we can get the following formulas for computing t_c and s_c respectively, where l_1, l_2 and l_3 are the loop length of indices i, j and k respectively.

$$t_c = |t_{11}|(l_1 - 1) + |t_{12}|(l_2 - 1) + |t_{13}|(l_3 - 1) + 1$$

$$s_c = [|t_{21}|(l_1-1) + |t_{22}|(l_2-1) + |t_{23}|(l_3-1) + 1] \; [|t_{31}|(l_1-1) + |t_{32}|(l_2-1) + |t_{33}|(l_3-1) + 1]$$

Definition 5. Three restricted row operations applied to a matrix A in $C^{n\times m}$ are defined as follows.

i. interchange two rows.

ii. multiply one row by -1.

iii. replace row_i by $row_i = row_i + k row_j$ where $i \neq j$ and k is an integer.

These restricted row operations can be implemented by pre-multiplying A by a restricted elementary matrix J_r in $C^{n\times n}$. For example:

$$row_1 \longleftrightarrow row_2\text{: } J_r = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$row_1 := -row_1\text{: } J_r = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$row_1 := row_1 + k row_3\text{: } J_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ k & 0 & 1 \end{bmatrix}.$$

Suppose J is a product of such restricted elementary matrices We call J a restricted row operation matrix.

Lemma 1. Suppose $J = J_n \cdots J_2 J_1$ is a restricted row operation matrix where the J_i's are restricted elementary matrices. Then

(i) $|J| = \pm 1$.

(ii) J^{-1} exists and is an integer matrix.

proof:

(i) Since $|J_i| = \pm 1$ i = 1, 2, ... , n. $|J| = |J_n| \cdots |J_1| = \pm 1$.

(ii) Since J_i^{-1} is an integer matrix for i = 1, 2, ... , n. $J^{-1} = J_1^{-1} \cdots J_n^{-1}$ exists and is an integer matrix.

Definition 6. Suppose the matrix A in $C^{m\times n}$ has rank 2. The restricted row normal form of A, denoted by A_N, is obtained by applying a sequence of restricted row operations and has the following form:

$$A_N = \begin{bmatrix} a_1 & a_2 & \dots & a_m \\ 0 & b_2 & \dots & b_m \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

where $a_1 > 0$, and $b_2 > a_2 \geq 0$

The restricted row normal form of a matrix has several interesting properties.

Lemma 2. If $B = J A$, then $A_N = B_N$ where J is a restricted row matrix, and A_N and B_N are restricted normal forms of A and B respectively.

Proof: Refer to [19] for details.

Lemma 3. If $A_N = J A$ where J is a restricted row operation matrix, then there exists a non-singular matrix Q in $C^{n\times n}$ such that

$$JAQ = \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix}$$ where I_2 denotes the second order identity.

Definition 7. Let A in $C^{m\times n}$ be an $m \times n$ matrix. A matrix A^G in $C^{n\times m}$ is called a generalized inverse of A [17] if, and only if,

$$A\ A^G\ A = A. \tag{5}$$

Lemma 4. If A^G is a generalized inverse of A, then

(i) a linear system $A\ x = b$ has a solution if, and only if, $A\ A^G\ b = b$.

(ii)

$$A^G = Q \begin{bmatrix} I_r & X \\ Y & Z \end{bmatrix} J \tag{6}$$

where X, Y and Z are any matrices in $C^{2\times(n-2)}$, $C^{(m-2)\times 2}$ and $C^{(m-2)\times(n-2)}$ respectively.

proof: See [17].

3. A Systematic Approach

Suppose T is a valid transformation of D, that is $|T| \neq 0$ and $\Delta = TD$ is a systolic matrix. let

$$T\ D = \Delta \tag{7}$$

where $T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$

$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ d_{31} & d_{32} & \dots & d_{3n} \end{bmatrix}$ and

$\Delta = \begin{bmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2n} \\ \delta_{31} & \delta_{32} & \dots & \delta_{3n} \end{bmatrix}$.

Transpose both sides of (7): $D^t\ T^t = \Delta^t$. Consider the following three cases, according to the value of the rank of D.

Case 1. rank of D is 3.

Since rank of D is 3, we can always choose a sub-matrix of D^t, D_3^t in $C^{3\times 3}$, such that $|D_3^t| \neq 0$. Suppose that D_3^t consists of first three rows of D^t (otherwise one can arrange such by exchanging columns in D). Let that matrix Δ_3^t be a sub-matrix of Δ^t consisting of the first three rows of Δ^t. We have $D_3^t\ T^t = \Delta_3^t$ or

$$(D_3^t)^{-1}\Delta_3^t = T^t. \tag{8}$$

Assume that $|\ \delta_{1j}\ | < B$ (which means that the maximum time delay units between two processing elements is less than B). All possible valid transformation matrices T's can be found by the following procedure.

Procedure 1.

step 1. Construct a set { T } by trying all possible Δ_3^t (total $O(B^3)$): if there is an integer solution T^t in equation (8) and $|\ T\ | \neq 0$, then add T to set { T }.

step 2. If the set { T } is empty, then there is no solution for D.

step 3. Otherwise, find optimal solution(s) among the set { T } which has (have) the minimum value of the product of $t_s \times s_s$.

Case 2. rank of D^t is 2.

Let $J D^T = D_N^t$ and $J D^t Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ . & . & . \\ . & . & . \\ . & . & . \\ 0 & 0 & 0 \end{bmatrix}$,

where D_N^t is the restricted row normal form of D^t, $J = [j_{ik}]$ in $C^{n\times n}$ and $Q = [q_{ij}]$ in $C^{3\times 3}$, $|P| \neq 0$, $|Q| \neq 0$. let $(D^t)^G$ be a generalized inverse of D^t. According to Lemma 4, $(D^t)^G = Q \begin{bmatrix} I_2 & X \\ Y & Z \end{bmatrix} J$ where $X \in C^{2\times(n-2)}$, $Y \in C^{1\times 2}$ and $Z \in C^{2\times(n-2)}$.

In light of Lemma 4, equation (1) ($D^t T^t = \Delta^t$) has solution

$<=> D^t (D^t)^G \Delta^t = \Delta^t$

$<=> J D^t (D^t)^G \Delta^t = J \Delta^t$

$<=> J D^t Q \begin{bmatrix} I_2 & X \\ Y & Z \end{bmatrix} J \Delta^t = J\Delta^t$

$<=> \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ . & . & . \\ . & . & . \\ . & . & . \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_2 & X \\ Y & Z \end{bmatrix} J \Delta^t = J \Delta^t$

$<=> \begin{bmatrix} I_2 & X \\ 0 & 0 \end{bmatrix} J \Delta^t = J \Delta^t$

(Let $J \Delta^t = \begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix}$ where

H_1 in $C^{2\times 2}$, H_2 in $C^{2\times(n-2)}$, H_3 in $C^{(n-2)\times 2}$, and H_4 in $C^{(n-2)\times(n-2)}$.)

$<=> H_1 = H_1 + XH_3$ and

$H_2 = H_2 + XH_4$ and

$H_3 = 0$ and

$H_4 = 0$.

<=>

$$\sum_{j=1}^{n} j_{kj}\delta_{ij} = 0 \quad k \geq 3 \text{ and } 1 \leq i \leq 3. \quad (9)$$

Therefore, the search space of all possible systolic matrices Δ can be reduced by the following constraints:

(i) rank of $\Delta = 2$ (rank of $D = 2$ and $|T| \neq 0$)

(ii) $\sum_{j=1}^{n} j_{kj}\delta_{ij} = 0$ for all $k \geq 3$ and $i = 1, 2, 3$.

In fact, if D has n columns and the rank of D is 2, there are only two possible forms of Δ:

(i) All elements of the first row of Δ are the same.

$\begin{bmatrix} -a & -a & \cdots & -a \\ b & b & \cdots & b \\ \delta_{31} & \delta_{32} & \cdots & \delta_{3n} \end{bmatrix}$ where a = 1, 2, ... , B and b in { 1,0,-1 }.

(ii) All elements of the first row of Δ are not the same.

$\begin{bmatrix} \delta_{11} & \delta_{12} & \cdots & \delta_{1n} \\ b & b & \cdots & b \\ b & b & \cdots & b \end{bmatrix}$ where δ_{1j} is in { 1, 2, ... , B }, and there are i_1 and i_2 such that $\delta_{1i_1} \neq \delta_{1i_2}$, j = 1, 2, ... , n and b $\in$ { 1, -1, 0 }.

Lemma 5. T is a valid transformation of D if and only if there is a valid transformation T^* of D_N where $D_N = J D$ is the restricted row normal form of D.

proof:

(i) If T is a valid transformation of D, we have $T D = \Delta$. Substituting D by $D = J^{-1} D_N$, yields $T J^{-1} D = \Delta$. Let $T^* = T J^{-1}$. We have $T^* D_N = \Delta$.

(ii) If T^* is a valid transformation of D_N,

substituting D_N by $D_N = J\,D$ in $T^* D_N = \Delta$ gives $T^* J D = \Delta$ Let $T = T^* J$. We have $T D = \Delta$.

Now, we consider the equation $T^* D_N = \Delta$. Let Δ_N be the restricted row normal form of Δ obtained by a restricted row operation matrix P: $\Delta_N = P \Delta$ where P is in $C^{3\times3}$. We have $T' D_N = \Delta_N$ where $T' = P T^*$.

Let $T' = \begin{bmatrix} t'_{11} & t'_{12} & t'_{13} \\ t'_{21} & t'_{22} & t'_{23} \\ t'_{31} & t'_{32} & t'_{33} \end{bmatrix}$,

$D_N = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ 0 & b_2 & \cdots & b_n \\ 0 & 0 & \cdots & 0 \end{bmatrix}$ and

$\Delta_N = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \\ 0 & d_2 & \cdots & d_n \\ 0 & 0 & \cdots & 0 \end{bmatrix}$. According to equation $T' D_N = \Delta_N$,

$$t'_{11} = \frac{c_1}{a_1},\ t'_{22} = \frac{d_2}{b_2},$$

$$t'_{12} = \frac{a_1 c_2 - c_1 a_2}{a_1 b_2} \qquad (10)$$

and $t'_{21} = 0$, $t'_{31} = 0$, and $t'_{32} = 0$.

Summarizing above results produces the following procedure to find an optimal valid transformation for a given algorithm (if it exists).

Procedure 2.

step 1. Construct $P \in C^{n\times n}$ such that $P D^t Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ \cdots & & \\ \cdots & & \\ \cdots & & \\ 0 & 0 & 0 \end{bmatrix}$, where P is the restricted row operation matrix and $P D^t = D_N^t =$

$\begin{bmatrix} a & b & d \\ 0 & c & e \\ 0 & 0 & 0 \\ \cdots & & \\ \cdots & & \\ \cdots & & \\ 0 & 0 & 0 \end{bmatrix}$ $a > 0$, $c > b \geq 0$ and $ac - b \neq 0$.

step 2. Construct a set $\{\Delta\}$, Δ in $C^{3\times n}$, by trying all possible systolic matrices Δ with rank value 2 that satisfy equation (9) (total $O(B^3 n^3)$).

step 3. If the set $\{\Delta\}$ is empty then there is no solution for the matrix D.

step 4. Examine each element Δ in the set $\{\Delta\}$ and form a set { T } by the following substeps until $\{\Delta\}$ is empty:

step 4.0 extract an element Δ from the set $\{\Delta\}$ and delete it from $\{\Delta\}$.

step 4.1. find restricted row normal forms of the matrices D and Δ, D_D and Δ_N, respectively ($P \Delta = \Delta_N$ and $J D = D_N$).

step 4.2. compute t'_{11}, t'_{22} t'_{12} according to (10). Let $t'_{13} = x$, $t'_{23} = y$, and $t'_{33} = z$.

step 4.3 if T' is not an integer matrix, then go to step 5; else compute $T^* = PT'$.

step 4.4 compute $T = T^* J$.

step 4.5 choose integers x, y and z in the matrix T such that the product of t_s and s_s has the minimum value.

step 4.6 add T into the set { T }.

step 5. If the set { T } is empty then there is no solution for the algorithm D else find optimal solution(s) among the set { T } which has (have) the minimum value of the product of t_s and s_s.

Case 3. rank of D = 1.

By using the same approach as in case 2 we can get a Procedure 3 which is similar to Procedure 2.

5. Conclusion

In this paper we have addressed the following two issues: (1) for a given nested loop algorithm , does there exist a valid transformation into some systolic array implementation ? (2) how to one can derive a valid transformation efficiently. We introduce the use of restricted row operations and normal form of a matrix and the concept and related properties of the generalized inverse to answer the above question for constant-delay systolic arrays: arrays which have a limited number of time delays between communication processing elements. The restricted row normal form is proven to be unique and can reduce the search space significantly. The generalized inverse is a power tool to solve the linear systems where the number of the variables do not equal to the number of simultaneously equations and the coefficient matrix may not necessary to be non-singular. A software package for automaticly selecting an optimal valid transformation (if it exists) in terms of the corresponding systolic array implementation requiring minimum computation time and minimum and VLSI space for a given nested loop algorithm is being implemented. The package uses the dependency matrix of a given algorithm and loop lengthes as its inputs. The outputs of the package are a set of optima solutions as well as their 2-D systolic array implementations including several important design parameters, interconnections among the processing elements and I/O scheduling.

6. References

[1] Kung, H. T., Why Systolic Architectures, CMU-SS-81-148, Carnegie-Mellon University, 1981.

[2] Kung, S. Y., Arun, K. S., Gal-Ezer, R. J. and Rao, D. V. B., Wavefront Array Processors: Language, Architecture and Applications, IEEE Trans. Comput., C-31, pp 1054-1066, 1982.

[3] Kung, H. T. and Leiserson, C. E., Systolic Arrays for VLSI, SIAM Sparse Matrix Symp., pp 256-282, 1980.

[4] Guibas, L. J., Kung, H. T. and Thompson, C. D., Direct VLSI Implementation of Combinatorial Algorithms, Proc. Caltech Conference on VLSI, pp 509-525, 1979.

[5] Moldovan, D. I., On the Design of Algorithms for VLSI Systolic Arrays, Proc. IEEE, Vol. 71, No. 1, pp 113-120, 1983.

[6] Miranker, W. L. and Winkler, A., Space-time Representations of Computational Structures, Journal of Computing, Vol. 32, pp 93-114, 1984.

[7] Moldovan, D. I., ADVIS: A Software Package for the Design of Systolic Arrays, IEEE Trans. Comput.-Aided Design, Vol. CAD-6, pp 33-40, 1987.

[8] Cappello, P. R. and Steiglitz, K., Unifying VLSI Array Designs with Geometric Transformations, in Proc. 1983 Conf. on Parallel Processing, pp 448-457, 1983.

[9] Culik and Fris, Topological Transformations as a Tool in the Design of Systolic Network, TCS, pp 183-216, 1985.

[10] Quinton, P., Automatic Synthesis of Systolic Arrays from Uniform Recurrent Relations, in Proc. 11th Int. Symp. Computer Architecture, pp 208-214.

[11] Gachet, P., Joinnault, B. and Quinton, P., Synthesizing Systolic Arrays Using DLASTOL. Systolic Arrays, Will Morre et al eds., Adam Hilger, pp 25-36, 1987.

[12] V. Van Dongen, V. V. and Quinton, P., Uniformization of Linear Recurrence Equations: A Step Towards the Automate Synthesis of Systolic Arrays. Int'l. Conf. Systolic Arrays, pp 478-482, 1988.

[15] Bertolazzi, C., Gnerra, C. and Salza, S., A Systematic Approach to the Designs of Modular Systolic Arrays. Int'l. Conf. Systolic Arrays, pp 453-462, 1988.

[16] Zhang, C., Law, A. G., and Rezazadeh, A., Designing VLSI Systolic Arrays with Complex Processing Elements. First Great lakes Symp. on VLSI, pp 207-213, 1991.

[17] Jodar, L., Law, A. G., Rezazadeh, A. Weston, J. H. and Wu, G., Computation for the Moore Penrose and Other Generalized Inverses, to appear in Congressus Numerantium, 1991.

[18] Razazadeh, A, Law, A. G. and Zhang, C., Generalized Inverses in Designing Systolic Arrays, to be presented in 21st Manitoba Conf. on Numerical math & Comb., Oct. 1991.

[19] Zhang, C. and Li, H. F., Mapping Nested DO-Loops into Universal Systolic Arrays, submitted to Journal of Parallel Computing, 1991.