FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. 15 (2000), 123–132

DETERMINING THE NUMBER OF PROCESSING ELEMENTS IN SYSTOLIC ARRAYS

I. Ž. Milovanović, T. I. Tokić, E. I. Milovanović, M. K. Stojčev

Dedicated to Professor R. Ž. Djordjević in the occasion of his 65th birthday

Abstract. In this paper we determine the minimal number of processing in the 2D systolic implementation for one class of nested loop algorithms. The number of processing elements is derived depending on the projection direction and size of loops. Obtained results are illustrated on matrix multiplication algorithm.

1. Introduction

VLSI technology has made possible the integration of circuits with hundreds of thousands of components into a single silicon chip. This high level of integration opens the way for massive parallel computations. Systolic processing constitutes a feasible solution for massive parallel computations. Its principles are compatible with VLSI technology characteristics [8]. Since systolic arrays are highly regular, only algorithms with repetitive computations perform well on them. Algorithms with nested loops fall into this category.

An important problem associated with designing systolic arrays is the mapping algorithm into systolic array architecture. Several techniques have been proposed for this purpose. Particularly useful here is the approach based on space–time representation of computation structure [1–10]. This method may be also used to examine the performances of possible systolic array implementation. Various criteria can be used to compare the performances of systolic arrays. The array size, which is defined as the number of processors in the array, obviously determines the basic hardware cost.

Received July 2, 1999.
1991 *Mathematics Subject Classification*. Primary 68M07; Secondary 68Q35.

Therefore a systolic array (SA) which has a minimum number of processing elements (PE) gives the optimal solution with respect to this cost function.

The objective of this paper is to determine the minimal number of processing elements (PE) in the 2D systolic implementations for one class of nested loop algorithms, according to projection direction and size of loops. For given projection direction, μ, we first introduce a linear transformation H, which maps index space C_D into a new index space $\bar{C}_D$. This transformation accommodates C_D to the projection direction μ. When a space–time transformation, which maps index space into systolic array, is applied on $\bar{C}_D$ the result is a 2D systolic array with minimal number of PEs.

The rest of the paper is organized as follows. Section 2 contains background and problem definition. In section 3 we define a class of adaptable algorithms. In section 4 we define a one–to–one mappings of index space for adaptable algorithms, and determine minimal number of PEs in the 2D systolic array implementation. Then we compare obtained results with those found in the literature.

2. Background

Each regular 3–nested loop algorithm can be characterized by a pair (D, C_D) (see for example [7–9]), where D is data dependency matrix and $C_D = \{(i,j,k) \mid 1 \le i \le N_1, 1 \le j \le N_2, 1 \le k \le N_3\}$ is the index space where the data are used or computed. The systolic array implementation can be obtained by a linear transformation

$$T = \begin{bmatrix} \Pi \\ -- \\ S \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ -- & -- & -- \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}, \tag{2.1}$$

where Π determines time scheduling, and S is the space mapping function determining PE locations and the communication channels between them. If matrix T is nonsingular, i.e. $\det T \ne 0$, and all elements of $\Delta_t = \Pi D$ are positive (or negative depending on convention), it is said that T is valid space–time transformation of (D, C_D) denoted as

$$T(D, C_D) = (\Delta, C_\Delta), \tag{2.2}$$

where

$$\Delta = (\Delta_t, \Delta_s), \quad \Delta_t = \Pi D, \quad \Delta_s = SD, \quad C_\Delta = [\,t \quad x \quad y\,]^T$$

and

$$t = \Pi\,[i \quad j \quad k\,]^T, \quad [x \quad y\,]^T = S\,[i \quad j \quad k\,]^T \quad \text{for all} \quad [i \quad j \quad k\,]^T \in C_D.$$

Vector $[x\,y]^T$ determines the $x - y$ coordinates of the PE in the projection plane. Several designing tools have been proposed for finding valid transformations T [1, 7–9].

Each transformation matrix T defined by (2.1) is associated with unique direction projection $\mu = [\mu_1\mu_2\mu_3]^T$, for which the following is valid

$$S \cdot \mu = 0. \tag{2.3}$$

It is assumed that rows of matrix S are linearly independent, i.e. that rank$S = 2$.

As we have already mentioned we are looking for transformations that give the smallest number of PEs in the 2D systolic arrays. For the sake of comparison, we will first present the corresponding results obtained in [11] (see also [12] for 3-nested loop algorithm. The expression for number of processing elements, Ω_p, depends only on space–time transformation, T, and the size of loops (N_1, N_2, N_3).

Theorem 1 ([11]). *Let $\omega = (N_1 - a_1)(N_2 - a_2)(N_3 - a_3)$. Then*

$$\Omega_p = \begin{cases} N_1N_2N_3, & \text{if } a_i > N_i \text{ for some } 1 \le i \le 3, \\ N_1N_2N_3 - \omega, & \text{otherwise,} \end{cases} \tag{2.4}$$

where

$$a_i = \left| \frac{T_{1i}}{\gcd(T_{11}, T_{12}, T_{13})} \right| .$$

In the previous expression, $T_{1i}, i = 1, 2, 3$ is $(1, i)$ – cofactor of matrix T, while $\gcd(T_{11}, T_{12}, T_{13})$ denotes the greatest common divisor of the nonzero integers, T_{11}, T_{12} and T_{13}. It is obvious that the expression (2.4) for Ω_p depends only on transformation T and the size of loops N_1, N_2 and N_3. But, if we take into account some properties of the algorithm, the result can be optimized. Namely, there is a broad class of algorithms with the property that index space C_D can be accommodated to the projection direction μ. This accommodation is performed by one–to–one mapping $H : \quad C_D \to \bar{C}_D$, where H depends on μ. If we then apply transformation T on $\bar{C}_D$ we obtain the set C_Δ with fewer number of processing elements.

3. Definitions

In this section, we give some preliminary definitions as a basis for the description that follows.

Let $\mathcal{A}$ be a regular 3–nested loop algorithm with index space $C_D = \{(i,j,k)|1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3\}$. We introduce the following subclasses of $\mathcal{A}$.

Definition 1. *If the ordering of computations in algorithm $\mathcal{A}$, for some fixed j (i), may be performed over arbitrary permutations of index variables i and k (j and k), we say that $\mathcal{A}$ is $\mathcal{A}(i,k)$ ($\mathcal{A}(j,k)$) adaptable.*

Remark 1. If a given algorithm $\mathcal{A}$ is both $\mathcal{A}(i,k)$ and $\mathcal{A}(j,k)$ adaptable, we say that it is adaptable.

In the sequel we define one–to–one mappings for adaptable algorithms which in composition with T enable to obtain 2D systolic arrays with minimal number of processing elements.

4. One–to–one Mappings for Adaptable Algorithms

Let $\mathcal{A}$ be an algorithm characterized by a pair (D, C_D) and valid transformation T. Let $\mu = [\mu_1\, \mu_2\, \mu_3]^T$ be a projection which corresponds to T. The accommodation of index space C_D to the direction μ is performed by "1–1" mapping $H = (F, G)$, $\quad H: \quad C_D \to \bar{C}_D$, where F is 3×3 matrix whose elements depend on μ, and G is 3×1 matrix with constant coefficients. Matrix G elements are determined from the condition that H performs mapping from the first into the first octant of Euclidian space. The definition of H for adaptable algorithms is as follows:

Definition 2. Suppose that a given algorithm is of type $\mathcal{A}(i,k)$. If $\mu = [\mu_1\, \mu_2\, \mu_3]^T$ is allowable projection direction with $\mu_2 = 1$, then mapping $H = (F, G)$ is defined by

$$F = \begin{bmatrix} 1 & \mu_1 & 0 \\ 0 & 1 & 0 \\ 0 & \mu_3 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} g_1 \\ 0 \\ g_3 \end{bmatrix}, \tag{4.1}$$

where g_1 and g_3 are smallest integers determined such that for each $[i\, j\, k]^T \in C_D$ the following is valid

$$u = i + \mu_1 j + g_1 > 0, \quad w = k + \mu_3 j + g_3 > 0. \tag{4.2}$$

The elements u and w are obtained according to

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = F \begin{bmatrix} i \\ j \\ k \end{bmatrix} + G = \begin{bmatrix} 1 & \mu_1 & 0 \\ 0 & 1 & 0 \\ 0 & \mu_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \\ k \end{bmatrix} + \begin{bmatrix} g_1 \\ 0 \\ g_3 \end{bmatrix}. \tag{4.3}$$

Remark 2. If $\mu_2 = -1$, then μ_1 and μ_3 in (4.1), (4.2) and (4.3) should be substituted by $(-\mu_1)$ and $(-\mu_3)$, respectively.

Definition 3. Suppose that a given algorithm is of type $\mathcal{A}(j,k)$. If $\mu = [\mu_1\, \mu_2\, \mu_3]^T$ is allowable projection direction with $\mu_1 = 1$, then mapping $H = (F, G)$ is defined by

$$F = \begin{bmatrix} 1 & 0 & 0 \\ \mu_2 & 1 & 0 \\ \mu_3 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ g_2 \\ g_3 \end{bmatrix}, \tag{4.4}$$

where g_2 and g_3 are smallest integers determined such that for each $[i\ j\ k]^T \in C_D$ the following is valid

$$v = \mu_2 i + j + g_2 > 0, \quad w = \mu_3 i + k + g_3 > 0. \tag{4.5}$$

The elements v and w are obtained according to

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = F \begin{bmatrix} i \\ j \\ k \end{bmatrix} + G = \begin{bmatrix} 1 & 0 & 0 \\ \mu_2 & 1 & 0 \\ \mu_3 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \\ k \end{bmatrix} + \begin{bmatrix} 0 \\ g_2 \\ g_3 \end{bmatrix}. \tag{4.6}$$

Remark 3. If $\mu_1 = -1$, then μ_2 and μ_3 in (4.4), (4.5) and (4.6) should be substituted by $(-\mu_2)$ and $(-\mu_3)$, respectively.

Before we determine the number of PEs in the systolic array, let us point out to some properties of mapping $H = (F, G)$ defined by (4.4) or (4.1):

- The mapping $H = (F, G)$ is "1–1";
- Suppose that $H = (F, G)$, $H : \ C_D \to \bar{C}_D$ is mapping defined by (4.4) (or (4.1)). Then each line parallel with direction $\mu = [1\, \mu_2\, \mu_3]^T$ (or $\mu = [\mu_1\, 1\, \mu_3]^T$) which passes through one point of $\bar{C}_D$ contains N_1 (i.e., N_2) points from $\bar{C}_D$. There are $N_2 N_3$ (i.e. $N_1 N_3$) such lines.
- The composition of T and F, i.e., $M = T \circ F$, is a regular mapping.

In the remainder of this section we will determine the minimal number of PEs in 2D systolic implementation for adaptable algorithms.

Theorem 2. *Suppose that a given algorithm $\mathcal{A}$ is $\mathcal{A}(i,k)$ adaptable. The number of PEs in the* 2D *array obtained by the projection direction* $\mu = [\mu_1 \;\; 1 \;\; \mu_3]^T$ *is*

$$\Omega_p = N_1 N_3. \tag{4.7}$$

Proof. Suppose that algorithm $\mathcal{A}$ is characterized by a pair (D, C_D), $C_D = \{(i,j,k) \mid 1 \le i \le N_1, 1 \le j \le N_2, 1 \le k \le N_3\}$, and valid transformation T, defined by (2.1). Since the algorithm is of $\mathcal{A}(i,k)$ type we can apply mapping $H = (F, G)$ defined by (4.1). The systolic array implementation is obtained according to composite mapping $T \circ H$, i.e., according to

$$(C_D) \overset{H}{\longmapsto} (\bar{C}_D) \quad \text{and} \quad (D, \bar{C}_D) \overset{T}{\longmapsto} (\Delta, C_\Delta).$$

Since G is matrix with constant coefficients, the number of PEs, Ω_p, in the array depends only on matrix $M = T \circ F$, that is

$$\begin{aligned} M = T \circ F &= \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & \mu_1 & 0 \\ 0 & 1 & 0 \\ 0 & \mu_3 & 1 \end{bmatrix} \\ &= \begin{bmatrix} t_{11} & \mu_1 t_{11} + \mu_2 t_{12} + \mu_3 t_{13} & t_{13} \\ t_{21} & 0 & t_{23} \\ t_{31} & 0 & t_{33} \end{bmatrix}. \end{aligned} \tag{4.8}$$

Note that in (4.8) we have used the condition (2.3), $S \cdot \mu = 0$, i.e., $\mu_1 t_{21} + \mu_2 t_{22} + \mu_3 t_{23} = 0$ and $\mu_1 t_{31} + \mu_2 t_{32} + \mu_3 t_{33} = 0$. Since M is valid transformation and $M_{11} = M_{13} = 0$, then according to Theorem 1 we have that $a_1 = 0, \quad a_2 = 1, \quad a_3 = 0$, i.e.,

$$\Omega_p = N_1 N_2 N_3 - N_1(N_2 - 1) N_3 = N_1 N_3. \quad \square$$

Theorem 3. *Suppose that a given algorithm $\mathcal{A}$ is $\mathcal{A}(j,k)$ adaptable. The number of PEs in the* 2D *array obtained by the projection direction* $\mu = [1 \;\; \mu_2 \;\; \mu_3]^T$ *is*

$$\Omega_p = N_2 N_3. \tag{4.9}$$

The proof is similar to that of Theorem 2.

Corollary 1. *Suppose that a given algorithm $\mathcal{A}$ is adaptable. The number of PEs in the* 2D *array obtained by the projection directions* $\mu = [1\ \ 1\ \ \mu_3]^T$, *or* $\mu = [1\ \ -1\ \ \mu_3]^T$, *is*

$$\Omega_p = N_3 \cdot \min\{N_1, N_2\}. \tag{4.10}$$

Remark 4. We assume that directions μ and $-\mu$ are equal.

Remark 5. If for a given algorithm $\mathcal{A}$ allowable direction is $\mu = [0\ \ 0\ \ 1]^T$, we have a trivial case $F = I$ and $G = 0$. In that case

$$\Omega_p = N_1 N_2. \tag{4.11}$$

Let us point out that for adaptable algorithms results obtained according to (2.4) are inferior compared to those obtained according to (4.7), (4.9) or (4.10). We will illustrate this fact on the example of matrix multiplication algorithm.

Algorithm (matrix multiplication $C = A \times B$)
 for $k := 1$ **to** N_3 **do**
  for $j := 1$ **to** N_2 **do**
   for $i := 1$ **to** N_1 **do**
   $a(i,j,k) := a(i,j-1,k);$
   $b(i,j,k) := b(i-1,j,k);$
   $c(i,j,k) := c(i,j,k-1) + a(i,j,k) * b(i,j,k);$

where $a(i,0,k) \equiv a_{ik}$, $b(0,j,k) \equiv b_{kj}$, $c(i,j,0) \equiv 0$ for all i,j and k. It is not difficult to conclude that a given algorithm is both of type $\mathcal{A}(i,k)$ and $\mathcal{A}(j,k)$. For the purpose of comparison we will take the same valid transformations T_1 and T_2 as in [11], that is

$$T_1 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad T_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

The projection direction which corresponds to T_1 is $\mu = [1\ \ 1\ \ 0]^T$. Accordingly, we can use Corollary 1 to determine the number of PEs in the corresponding systolic array. According to (4.10) we have

$$\Omega_p(T_1) = N_3 \cdot \min\{N_1, N_2\} \tag{4.12}$$

compared to

$$\Omega_p(T_1) = N_3 \cdot (N_1 + N_2 - 1)$$

obtained according to Theorem 1 (see [11]).

Similarly, for T_2 the corresponding direction is $\mu = [1 \;\; 1 \;\; -1]^T$. Thus, according to Corollary 1

$$\Omega_p(T_2) = \Omega_p(T_1) = N_3 \cdot \min\{N_1, N_2\}$$

compared to

$$\Omega_p(T_2) = N_1N_2N_3 - (N_1 - 1)(N_2 - 1)(N_3 - 1)$$

obtained by the Theorem 1 (se [11]).

Similar results are obtained for all other allowable projection directions ($\mu = [1 \;\; 0 \;\; 0]^T$, $\mu = [0 \;\; 1 \;\; 0]^T$, $\mu = [0 \;\; 0 \;\; 1]^T$, $\mu = [1 \;\; 0 \;\; 1]^T$, $\mu = [0 \;\; 1 \;\; 1]^T$, $\mu = [1 \;\; 1 \;\; 1]^T$, $\mu = [1 \;\; -1 \;\; 1]^T$, $\mu = [1 \;\; -1 \;\; -1]^T$) for a matrix multiplication algorithm.

Figures 1 and 2 show systolic array implementations obtained by T_2 for the case $N_1 = N_2 = N_3 = 2$ obtained according to Theorem 1 and Corollary 1, respectively.

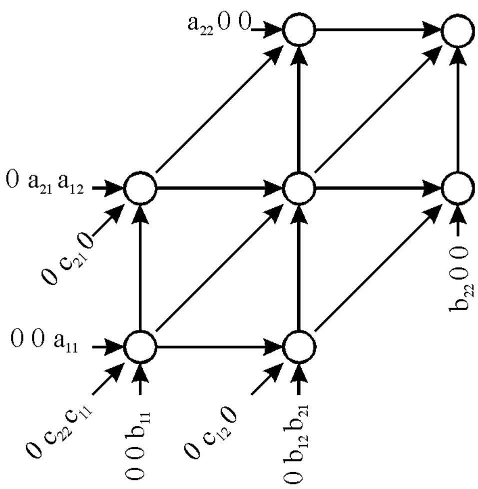

FIG. 1. Systolic array obtained according to Theorem 1

5. Conclusion

In this paper we have determined the minimal number of PEs in the 2D systolic implementations for one class of 3–nested loop algorithms. We

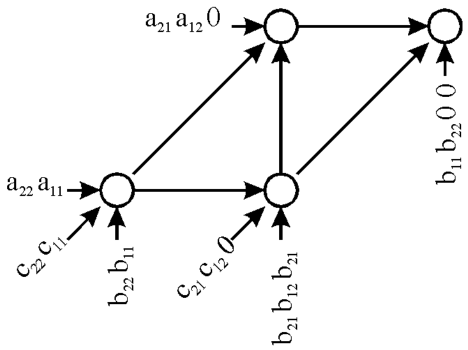

FIG. 2. Systolic array obtained according to Corollary 1

have defined a class of adaptable algorithms for which we introduce some linear transformations that accommodate the index space of algorithm to the projection direction. This accommodation enables us to obtain 2D SAs with the smallest number of PEs. The number of PEs depends on the size of loops and projection direction. We have illustrated the obtained results on the matrix multiplication algorithm.

REFERENCES

1. M. O. ESONU, J. AL–KHALILI, S. HARIRI, D. AL–KHALILI: *Systolic arrays: How to chose them.*, IEE Proc. **139**, No. 3 (1992), 179–188.
2. S. Y. KUNG: *VLSI Array Processors.* Prentice Hall, New Jersey, 1988.
3. P.-Z. LEE, Z.-M. KEDEM: *Mapping nested loop algorithms into multidimensional systolic arrays.* IEEE Trans. Parallel Distr. Syst. **1**, No. 1 (1990), 64–76.
4. G. J. LI, B. W. WAH: *The design of optimal systolic arrays.* IEEE Trans. Comput. **C–34**, No. 1 (1985), 66–77.
5. C.-M. LIU, C.-W. JEN: *Design of algorithm–based fault–tolerant VLSI array processor.* IEE Proc. **136**, Pt. E, 6 (1989), 539–547.
6. I.Z. MILENTIJEVIĆ, I.Ž. MILOVANOVIĆ, E.I. MILOVANOVIĆ, M.K. STOJČEV: *The design of optimal planar systolic arrays for matrix multiplication.* Comput. Math. Appl. **33**, No. 6 (1997), 17–35.
7. W. L. MIRANKER, A. WINKLER: *Space time representations of computational structures.* Computing **32** (1984), 93–114.
8. D. I. MOLDOVAN: *Parallel Processing: From Applications to systems.* Morgan Kaufman, San Mateo, CA, 1993.

9. D. I. MOLDOVAN, J. A. B. FORTES: *Partitioning of algorithms for fixed size VLSI architectures.* IEEE Trans. Comput. **C–35**, No. 1 (1986), 1–12.

10. S. G.SEDUKHIN: *The designing and analysis of systolic algorithms and structures.*, Programming **2** (1991), 20–40 (Russian).

11. C. N.ZHANG, J. H.WESTON, Y.-F. YAN: *Determining object functions in systolic array designs.* IEEE Trans. Very Large Scale Integration (VLSI) Systems **2**, No. 3 (1994), 357–360.

12. C. N. ZHANG, T. M. BAHTIAR, W. K. CHOU: *Optimal fault–tolerant design approach for VLSI array.* IEEE Proc. Comput. Dig. Techn. **144**, No. 1 (1997), 15–23.

Faculty of Electronic Engineering
P.O. Box 73
18000 Niš, Serbia