FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 29, No 3 (2014), 243–259

SYSTOLIC ALGORITHMS FOR MATRIX MULTIPLICATION ON SPACE OPTIMAL 1D SYSTOLIC ARRAYS

B. M. Randjelović, E. I. Milovanović, I. Ž. Milovanović

Abstract. In this paper we define and discuss various systolic algorithms for synthesis of one-dimensional systolic arrays (1DSA) with two-dimensional links, suitable for multiplication of rectangular matrices. It is shown that by choosing appropriate algorithm it is always possible to design optimal systolic array with respect to dimension of matrices.

Keywords: Systolic algorithms; Systolic arrays; Matrix multiplication; Graph.

1. Introduction

Matrix multiplication is one of the essential operations in various fields of science, engineering and technology, such as signal and image processing, system theory, statistical and numerical analysis, biomedical researches, etc. This operation is characterized by intensive computational complexity and regularity, and it is often required under real time constraints. Today's high performance computing systems exploit one or more forms of parallelism to achieve high speed computations. To fulfill the desired throughput rates for time-critical and computationally intensive problems, special-purpose, high-speed computing systems optimized for processing specific tasks have been designed. A systolic array is a type of special-purpose system that can be used for implementing such tasks.

Let $A = (a_{ik})$ and $B = (b_{kj})$ be two rectangular matrices of order $N_1 \times N_3$ and $N_3 \times N_2$, respectively. In this paper we derive all systolic algorithms, which can be used for synthesis of 1DSA, suitable for computing matrix product $C = A \cdot B$. We are interested in algorithms for three different type of systolic arrays: static (type I), bidirectional (type II) and unidirectional (type III). All arrays should be space optimal with respect to dimensions of matrices N_1, N_2 and N_3, i.e. have a minimal number of processing elements (PE), for the given problem size (see, for example [3],[4],[5] and [12]). The procedure of SA synthesis will be given briefly, since it is discussed in the literature thoroughly (see [1],[4] and [5]). Explicit formulas for the PEs locations and initial data schedule in the (x, y)-plane will be given.

Received July 24, 2014.; Accepted August 01, 2014.
2010 *Mathematics Subject Classification.* Primary 65Y05; Secondary 65Y10, 65F30

Proposed algorithms are compared with the ones proposed in [6, 7, 8, 3, 11] for matrix-vector multiplication, and with those proposed in [4, 5, 12], for matrix multiplication, with respect to space optimality.

2. Mathematical background

There are many different methods for computing the product of two matrices $C = A \cdot B$, where $A = (a_{ik})$ and $B = (b_{kj})$ are rectangular matrices of order $N_1 \times N_3$ and $N_3 \times N_2$, respectively. Apart from straightforward methods, which are given at the beginning, several more methods are discussed: middle product algorithm, dual middle product algorithm, and method of inner products (see [13]).

The standard method for computing $C = (c_{ij})$ is given by the following recurrence relations

$$\begin{aligned} c_{ij}^{(0)} &:= 0, \\ c_{ij}^{(k)} &:= c_{ij}^{(k-1)} + a_{ik}b_{kj}, \; k = 1, 2, \ldots, N_3 \\ c_{ij}^{(N_3)} &= c_{ij} \end{aligned}$$

for $i = 1, 2, \ldots, N_1$, $j = 1, 2, \ldots, N_2$.

Middle product algorithm, dual middle product algorithm, and inner product algorithm, are defined as follows:

$$C = A \cdot B = \left[\, A\vec{B}_{\bullet 1} \; A\vec{B}_{\bullet 2} \; \cdots \; A\vec{B}_{\bullet N_2} \,\right], \tag{2.1}$$

$$C = A \cdot B = \left[\, \vec{A}_{1\bullet}B \; \vec{A}_{2\bullet}B \; \cdots \; \vec{A}_{N_1\bullet}B \,\right], \tag{2.2}$$

$$C = A \cdot B = \sum_{k=1}^{N_3} \vec{A}_{\bullet k}\vec{B}_{k\bullet}, \tag{2.3}$$

where $\vec{A}_{i\bullet}$ and $\vec{B}_{k\bullet}$ denote i-th and k-th row-vectors, and $\vec{A}_{\bullet k}$ and $\vec{B}_{\bullet j}$, k-th and j-th column-vectors, of matrices A and B, respectively.

As we have already mentioned, we consider 1DSAs with two-dimensional links. Without loss of generality, we design algorithms for computing a part of matrix C, from (2.1), (2.2) and (2.3). Namely, we derive systolic algorithms for computing column-vector $\vec{C}_{\bullet 1} = A\vec{B}_{\bullet 1}$, row-vector $\vec{C}_{1\bullet} = \vec{A}_{1\bullet}B$ and only first iteration of the resulting matrix $C^{(1)} = \vec{A}_{\bullet 1}\vec{B}_{1\bullet}$. Matrix C can be obtained by repeating the corresponding computations n times.

All algorithms in this paper are considered in the three-dimensional Cartesian space, generated by unity vectors

$$\vec{e}_1 = [1\;0\;0]^T,\; \vec{e}_2 = [0\;1\;0]^T \text{ and } \vec{e}_3 = [0\;0\;1]^T. \tag{2.4}$$

To compute column-vector $\vec{C}_{\bullet 1} = A\vec{B}_{\bullet 1}$, row-vector $\vec{C}_{1\bullet} = \vec{A}_{1\bullet}B$ and $C^{(1)} = \vec{A}_{\bullet 1}\vec{B}_{1\bullet}$, the following basic algorithms can be used:

Algorithm I
for $k := 1$ **to** N_3 **do**
for $i := 1$ **to** N_1 **do**
$a(i,1,k) := a(i,0,k)$
$b(i,1,k) := b(0,1,k)$
$c(i,1,k) := c(i,1,k-1) + a(i,1,k)b(i,1,k)$

where $a(i,0,k) \equiv a_{ik}$, $b(0,1,k) \equiv b_{k1}$, $c(i,1,k) \equiv c_{i1}^{(k)}$, $c(i,1,0) \equiv c_{i1}^{(0)} = 0$, for each $i = 1,2,\ldots,N_1$, $k = 1,2,\ldots,N_3$.

Algorithm II
for $k := 1$ **to** N_3 **do**
for $j := 1$ **to** N_2 **do**
$a(1,j,k) := a(1,0,k)$
$b(1,j,k) := b(0,j,k)$
$c(1,j,k) := c(1,j,k-1) + a(1,j,k)b(1,j,k)$
where $a(1,0,k) \equiv a_{1k}$, $b(0,j,k) \equiv b_{kj}$, $c(1,j,k) \equiv c_{1j}^{(k)}$, $c(1,j,0) \equiv c_{1j}^{(0)} = 0$, for each $j = 1,2,\ldots,N_2$, $k = 1,2,\ldots,N_3$.

Algorithm III
for $j := 1$ **to** N_2 **do**
for $i := 1$ **to** N_1 **do**
$a(i,j,1) := a(i,0,1)$
$b(i,j,1) := b(0,j,1)$
$c(i,j,1) := c(i,j,0) + a(i,j,1)b(i,j,1)$
where $a(i,0,1) \equiv a_{i1}$, $b(0,j,1) \equiv b_{1j}$, $c(i,j,1) \equiv c_{ij}^{(1)}$, $c(i,j,0) \equiv c_{ij}^{(0)} = 0$, for each $i = 1,2,\ldots,N_1$, $j = 1,2,\ldots,N_2$.

3. Systolic algorithms and systolic arrays

Algorithms I, II and III have global data dependencies and, consequently, they are not convenient for systolic processing. To make them systolic, it is necessary to remove global data dependencies.

Denote with $\vec{r}_b$, $\vec{r}_a$ and $\vec{r}_c$ data dependency vectors of matrices B, A and C, respectively. For **Algorithm I**, we have $\vec{r}_b = [i\ 0\ 0\ ]^T$, for **Algorithm II** we have $\vec{r}_a = [0\ j\ 0\ ]^T$. For **Algorithm III** we have $\vec{r}_a = [0\ j\ 0\ ]^T$ and $\vec{r}_b = [i\ 0\ 0\ ]^T$. Since these vectors are not constant, but depend of index-variables, above algorithms have global dependencies, so they are not systolic (see [2], [10], [14]).

Systolization of Algorithms I, II and III is performed by removing global dependencies and by localization ([1], [14]). This is achieved by the substitution of dependency vectors $\vec{r}_b$, $\vec{r}_a$ and $\vec{r}_c$, in all three algorithms, with vectors $\vec{e}_b^{\ 3} = [1\ 0\ 0]^T$, $\vec{e}_a^{\ 3} = [0\ 1\ 0]^T$ and $\vec{e}_c^{\ 3} = [0\ 0\ 1]^T$. The obtained systolic algorithms have the following form:

Algorithm 1

for $k := 1$ **to** N_3 **do**

for $i := 1$ **to** N_1 **do**

$a(i, 1, k) := a(i, 0, k)$

$b(i, 1, k) := b(i-1, 1, k)$

$c(i, 1, k) := c(i, 1, k-1) + a(i, 1, k)b(i, 1, k)$

Algorithm 2

for $k := 1$ **to** N_3 **do**

for $j := 1$ **to** N_2 **do**

$a(1, j, k) := a(1, j-1, k)$

$b(1, j, k) := b(0, j, k)$

$c(1, j, k) := c(1, j, k-1) + a(1, j, k)b(1, j, k)$

Algorithm 3

for $j := 1$ **to** N_2 **do**

for $i := 1$ **to** N_1 **do**

$a(i, j, 1) := a(i, j-1, 1)$

$b(i, j, 1) := b(i-1, j, 1)$

$c(i, j, 1) := c(i, j, 0) + a(i, j, 1)b(i, j, 1)$

Each algorithm can be joined to a unique oriented graph $G = (P, D)$, where $P = \left\{\vec{r} = [i\ j\ k]^T\right\}$ and $D = [\vec{r}_b\ \vec{r}_a\ \vec{r}_c]^T$. The set of nodes $P = \left\{\vec{r} = [i\ j\ k]^T\right\}$ corresponds to the points where the computations are performed in the corresponding algorithm. Directed edges between the nodes are defined by the columns of data dependency matrix $D = [\vec{r}_b\ \vec{r}_a\ \vec{r}_c]^T$.

Denote by $\vec{\mu} = [\mu_1\ \mu_2\ \mu_3]^T$, $\mu_1^2 + \mu_2^2 + \mu_3^2 \neq 0$, projection direction vector. The projection of graph $G = (P, D)$ along direction $\vec{\mu}$ on the plane orthogonal to $\vec{\mu}$, is either a directed graph with loops, or a directed graph, or a directed multigraph, or some other similar structure. We refer to all of them as a *graph* and denote by $\Gamma = (Q, \Delta)$. Graph Γ is placed in a two-dimensional Cartesian space, i.e. in (x, y)-plane. This graph corresponds to the systolic array (SA) that implements the systolic algorithm. Set of vertices, Q, corresponds to a set of processing elements

(PEs) in the SA, while set of directed edges, Δ, corresponds to a communication channels in the array.

Each projection vector $\vec{\mu} = [\mu_1\ \mu_2\ \mu_3]^T$ corresponds to the matrix

$$S = \begin{bmatrix} s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}$$

which is called a valid transformation matrix.

Note that the transformation matrix is not unique for a given projection vector. For details about the selection of the appropriate projection vectors and corresponding transformation matrices for a given algorithm, see [1].

The Vertices of graph Γ, i.e. locations of PEs, in the (x, y)-plane and direction of edges (communication channels) are determined from the following equations

$$PE \rightarrow [\ x\ y\ ]^T = S \cdot \vec{p},\ \ \vec{p} \in P \ \text{ and } \ \Delta = \mathrm{S} \cdot \mathrm{D} = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^{\mathrm{T}}. \tag{3.1}$$

Equations (3.1) uniquely define the systolic array that implements the corresponding algorithm.

Algorithm 1 can be joined with directed coordinate graph $G_1 = (P_1, D_1)$, defined by

$$P_1 = \left\{ \vec{p} = [\ i\ \ 1\ \ k\ ]^T \mid 1 \leqslant i \leqslant N_1,\ 1 \leqslant k \leqslant N_3 \right\}$$

$$D_1 = \begin{bmatrix} \vec{e}_b^{\,3} & \vec{e}_a^{\,3} & \vec{e}_c^{\,3} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Possible projection vectors (see [9]) are $\vec{\mu} = [1\,0\,0]^T$, $\vec{\mu} = [0\,0\,1]^T$, $\vec{\mu} = [1\,0\,1]^T$ and $\vec{\mu} = [1\,0\ -1]^T$.

One of the valid transformation matrices that corresponds to the projection vector $\vec{\mu} = [1\,0\,0]^T$ have the following form

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Using transformation matrix S, directed graph $G_1 = (P_1, D_1)$ is mapped into directed graph with loops $\Gamma_1 = (Q_1, \Delta_1)$. Having in mind (3.1), we obtain

$$PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ k \end{bmatrix},\ \ \Delta_1 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.2}$$

for each $k = 1, 2, \ldots, N_3$. According to (3.2), we conclude that systolic array, SA1, consists of $\Omega = N_3$ processing elements (PEs) (see [3]). Since $\vec{e}_b^{\,2} = [0\,0]^T$, the obtained array is static (type I), i.e. one of the matrices is resident in the array. During the realization of **Algorithm 1**,the elements of column-vector $\vec{B}_{\bullet 1}$ are resident, while the elements of column-vector $\vec{C}_{1\bullet}$ are pipelined through the array.

For projection vector $\vec{\mu} = [0\ 0\ 1]^T$, one of the valid transformations is

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

According to (3.1), this transformation maps directed graph $G_1 = (P_1, D_1)$ into directed graph with loops $\Gamma_2 = (Q_2, \Delta_2)$ (i.e. the array SA2) defined by

$$\text{(3.3)} \qquad PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} i \\ 1 \end{bmatrix}, \ \Delta_2 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

for each $i = 1, 2, \ldots, N_1$. Based on (3.3), we conclude that systolic array, SA2, consists of $\Omega = N_1$ PEs (see [3]). Since $\vec{e}_c^{\,2} = [0\ 0]^T$, the array is also static (type I), with the elements of the resulting matrix C resident in the array and the elements of column-vectors $\vec{B}_{\bullet 1}$ pipelined through the array.

One of the valid transformation matrices for projection vector $\vec{\mu} = [1\ 0\ 1]^T$ is of the form

$$S = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

This transformation maps graph $G_1 = (P_1, D_1)$ into directed graph with $\Omega = N_1 + N_3 - 1$ nodes. The corresponding systolic array does not have an optimal number of PEs. To obtain the array with an optimal number of PEs we have to adjust graph $G_1 = (P_1, D_1)$ to the projection vector $\vec{\mu} = [1\ 0\ 1]^T$. This is accomplished by mapping set $P_1 = \left\{ \vec{p} = [\ i \ \ 1 \ \ k\ ]^T \ | \ 1 \leqslant i \leqslant N_1, \ 1 \leqslant k \leqslant N_3 \right\}$ into $P_2 = \left\{ \vec{p} = [\ u \ \ 1 \ \ w\ ]^T \right\}$, where

$$\begin{bmatrix} u \\ 1 \\ w \end{bmatrix} = [\vec{\mu}\ \ \vec{e}_2\ \ \vec{e}_3] \cdot \begin{bmatrix} i \\ 1 \\ k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} i \\ 1 \\ i+k-1 \end{bmatrix},$$

for each $i = 1, 2, \ldots, N_1$ and $k = 1, 2, \ldots, N_3$. The Obtained directed graph $G_2 = (P_2, D_1)$, that is also coordinate, uniquely corresponds to the following systolic algorithm, equivalent to **Algorithm 1**:

Algorithm 4
for $k := 1$ **to** N_3 **do**
for $i := 1$ **to** N_1 **do**
$a(i, 1, i+k-1) := a(i, 0, i+k-1)$
$b(i, 1, i+k-1) := b(i-1, 1, i+k-1)$
$c(i, 1, i+k-1) := c(i, 1, i+k-2) + a(i, 1, i+k-1)b(i, 1, i+k-1)$

where $a(i, 0, t+N_3) \equiv a(i, 0, t)$, $b(0, 1, t+N_3) \equiv b(0, 1, t)$, for each $i = 1, 2, \ldots, N_1$, $t = 1, 2, \ldots, N_3$. Using (3.1), directed graph $G_2 = (P_2, D_1)$ can be mapped into directed graph $\Gamma_3 = (Q_3, \Delta_3)$, defined by

$$PE \to \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1-k \\ 1 \end{bmatrix}, \ \Delta_3 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{3.4}$$

for each $k = 1, 2, \ldots, N_3$. According to (3.4), we conclude that corresponding systolic array, SA3, consists of $\Omega = N_3$ PEs (see [3]). Since $\vec{e}_b^{\,2} = -\vec{e}_c^{\,2}$, the array is bidirectional (type II). the elements of column-vectors $\vec{B}_{\bullet 1}$ and $\vec{C}_{1\bullet}$, are pipelined through the array in opposite directions.

For the projection vector $\vec{\mu} = [1\,0\,-1]^T$, one of the possible transformation matrices is

$$S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

According to (3.1), directed graph $G_1 = (P_1, D_1)$, is mapped into a directed multigraph defined by

$$PE \to \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} i+k \\ 1 \end{bmatrix}, \ \Delta = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The corresponding SA consists of $\Omega = N_1 + N_3 - 1$ PEs (see [3]), which is not an optimal number for the given problem size. On the other hand, since $\vec{e}_b^{\,2} = \vec{e}_c^{\,2}$, **Algorithm 1** cannot be implemented on this array correctly. However, this problem could be solved in two steps.

First, we will interpolate the set of nodes P_1 of graph G_1, with the set of nodes

$$\bar{P}_1 = \left\{ \vec{p} = \begin{bmatrix} i - \frac{1}{2} & 1 & k \end{bmatrix}^T \mid 1 \leqslant i \leqslant N_1,\ 1 \leqslant k \leqslant N_3 \right\}.$$

After that, we adjust set $P_1 \cup \bar{P}_1$ to projection vector $\vec{\mu} = [1\,0\,-1]^T$. The adjustment is implemented as mapping of $P_1 \cup \bar{P}_1$ into $P_3 \cup \bar{P}_3$, where $P_3 = \left\{ \vec{p} = [\ u\ \ 1\ \ w\ ]^T \right\}$ and $\bar{P}_3 = \left\{ \vec{p} = \begin{bmatrix} u - \frac{1}{2} & 1 & w \end{bmatrix}^T \right\}$, defined by

$$\begin{bmatrix} u \\ 1 \\ w \end{bmatrix} = [\vec{\mu}\ \ \vec{e}_2\ \ \vec{e}_3] \cdot \begin{bmatrix} i \\ 1 \\ k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ N_1 \end{bmatrix} = \begin{bmatrix} i \\ 1 \\ -i + k + N_1 \end{bmatrix}$$

for each $i = 1, 2, \ldots, N_1$ and $k = 1, 2, \ldots, N_3$. Directed edges in the obtained graph $G_3 = (P_3 \cup \bar{P}_3, D_3)$ are determined by column-vectors of the matrix

$$D_3 = \begin{bmatrix} \vec{e}_b^{\,3} & \vec{e}_a^{\,3} & \vec{e}_c^{\,3} \end{bmatrix}^T = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This graph corresponds to the following systolic algorithm, equivalent to **Algorithm 1**:

Algorithm 5
for $k := 1$ **to** N_3 **do**
for $i := 1$ **to** N_1 **do**
$b(i-\frac{1}{2},1,-i+k+N_1) := b(i-1,1,-i+k+N_1)$
$b(i,1,-i+k+N_1) := b(i-\frac{1}{2},1,-i+k+N_1)$
$a(i,1,-i+k+N_1) := a(i,0,-i+k+N_1)$
$c(i,1,-i+k+N_1) := c(i,1,-i+k+N_1-1) + a(i,1,-i+k+N_1)b(i,1,-i+k+N_1)$

where $a(i,0,t+N_3) \equiv a(i,0,t)$, $b(0,1,t+N_3) \equiv b(0,1,t)$, for each $i = 1,2,\ldots,N_1$, $t = 1,2,\ldots,N_3$.

Using valid transformation

$$S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

and (3.1), graph $G_3 = (P_3 \cup \bar{P}_3, D_3)$ is mapped into directed multi-graph $\Gamma_4 = (Q_4 \cup \underline{Q}_4, \Delta_4)$, defined by

$$PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k+N_1 \\ 1 \end{bmatrix}, \quad d \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k+N_1-\frac{1}{2} \\ 1 \end{bmatrix},$$

(3.5)

$$\Delta_4 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} \frac{1}{2} & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

for each $k = 1,2,\ldots,N_3$. The corresponding array, SA4, consists of $\Omega = N_3$ PEs (see [3]). Since $\vec{e}_b^{\,2} = \frac{1}{2}\vec{e}_c^{\,2}$, we conclude that the array is unidirectional (type III). During implementation of **Algorithm 5** on SA4 the elements of column-vectors $\vec{B}_{\cdot 1}$ and $\vec{C}_{1\bullet}$ are pipelined through the array in the same direction.

Systolic arrays SA1, SA2, SA3 and SA4, for matrix-vector multiplication, are well-studied in literature (see [6], [2], [4]).

In the case of **Algorithm 2**, possible projection vectors are $\vec{\mu} = [0\;1\;0]^T$, $\vec{\mu} = [0\;0\;1]^T$, $\vec{\mu} = [0\;1\;1]^T$ and $\vec{\mu} = [0\;1\;-1]^T$. Corresponding graph $G_5 = (P_5, D_5)$ is defined by

$$P_5 = \left\{\vec{p} = [\,1\;\;j\;\;k\,]^T \;|\, 1 \leqslant j \leqslant N_2,\; 1 \leqslant k \leqslant N_3\right\}$$

$$D_5 = \begin{bmatrix} \vec{e}_b^{\,3} & \vec{e}_a^{\,3} & \vec{e}_c^{\,3} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

One of the valid transformation matrices for projection vector $\vec{\mu} = [0\;1\;0]^T$ is

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Based on expression (3.1), directed graph $G_5 = (P_5, D_5)$ is mapped into directed graph with loops $\Gamma_5 = (Q_5, \Delta_5)$, defined by

$$\text{(3.6)} \qquad PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ k \end{bmatrix}, \ \Delta_5 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

for each $k = 1, 2, \ldots, N_3$. According to (3.6), the corresponding systolic array, SA5, consists of $\Omega = N_3$ PEs (see [3]). Since $\vec{e}_a^{\,2} = [0\ 0]^T$, the array SA5 is static (type I). During realization of **Algorithm 2**, the elements of row-vector $\vec{A}_{1\bullet}$ are resident in the array, while the elements of column-vector $\vec{C}_{1\bullet}$ are pipelined.

For projection vector $\vec{\mu} = [0\ 0\ 1]^T$ one of the valid transformations is

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Using this transformation, according to (3.1), graph $G_5 = (P_5, D_5)$ is mapped into directed graph with loops $\Gamma_6 = (Q_6, \Delta_6)$, defined by

$$\text{(3.7)} \qquad PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ j \end{bmatrix}, \ \Delta_6 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

for each $j = 1, 2, \ldots, N_2$. The corresponding systolic array, SA6, consists of $\Omega = N_2$ PEs (see [3]). Since $\vec{e}_c^{\,2} = [0\ 0]^T$, the array is static (type I). During implementation of **Algorithm 2** on the SA6, the elements of column-vector $\vec{C}_{1\bullet}$ are resident while the elements of row-vector $\vec{A}_{1\bullet}$ are pipelined through the array.

Similarly as in the case of the array SA3, for projection vector $\vec{\mu} = [0\ 1\ 1]^T$ we consider the following systolic algorithm, equivalent to **Algorithm 2**:

Algorithm 6
for $k := 1$ **to** N_3 **do**
for $j := 1$ **to** N_2 **do**
$a(1, j, k + j - 1) := a(1, j - 1, k + j - 1)$
$b(1, j, k + j - 1) := b(0, j, k + j - 1)$
$c(1, j, k + j - 1) := c(1, j, k + j - 2) + a(1, j, k + j - 1)b(1, j, k + j - 1)$

where $a(1, 0, t + N_3) \equiv a(1, 0, t)$, $b(0, j, t + N_3) \equiv b(0, j, t)$, for $j = 1, 2, \ldots, N_2$, $k = 1, 2, \ldots, N_3$.

The corresponding directed graph $G_6 = (P_6, D_6)$ is defined by

$$P_6 = \left\{ \vec{p} = [\ 1\ \ j\ \ k + j - 1\ ]^T \mid 1 \le j \le N_2,\ 1 \le k \le N_3 \right\}.$$

One of the valid transformations that corresponds to the projection vector $\vec{\mu} = [0\ 1\ 1]^T$ is

$$S = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Using this transformation, directed graph $G_6 = (P_6, D_5)$ is mapped into directed multi-graph $\Gamma_7 = (Q_7, \Delta_7)$, defined with

$$(3.8)\qquad PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1-k \\ 1 \end{bmatrix}, \ \Delta_7 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \end{bmatrix},$$

for each $k = 1, 2, \ldots, N_3$. From (3.8) we conclude that the corresponding systolic array, SA7, consists of $\Omega = N_3$ PEs (see [3]). Since $\vec{e}_a^{\,2} = -\vec{e}_c^{\,2}$, SA7 is bidirectional (type II). Elements $\vec{A}_{\bullet 1}$ and $\vec{C}_{1\bullet}$ are pipelined through the array in opposite directions.

Similarly to systolic array SA4, for projection vector $\vec{\mu} = [0\ 1\ -1]^T$, we consider the following systolic algorithm, equivalent to **Algorithm 2**:

Algorithm 7

for $k := 1$ **to** N_3 **do**

for $j := 1$ **to** N_2 **do**

$a(1, j-\frac{1}{2}, -j+k+N_2) := a(1, j-1, -j+k+N_2)$

$a(1, j, -j+k+N_2) := a(0, j, -j+k+N_2)$

$b(1, j, -j+k+N_2) := b(0, j, -j+k+N_2)$

$c(1, j, -j+k+N_2) := c(1, j, -j+k+N_2-1) + a(1, j, -j+k+N_2)b(1, j, -j+k+N_2)$

where $a(1, 0, t+N_3) \equiv a(1, 0, t)$, $b(0, j, t+N_3) \equiv b(0, j, t)$, for each $j = 1, 2, \ldots, N_2$, $t = 1, 2, \ldots, N_3$.

The corresponding directed graph $G_7 = (P_7, D_7)$ is defined by

$$P_7 = \left\{ \vec{p} = [\,1\ \ j\ \ -j+k-N_2\,]^T \mid 1 \leqslant j \leqslant N_2,\ 1 \leqslant k \leqslant N_3 \right\}$$

$$D_7 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

One of valid transformations that corresponds to projection vector $\vec{\mu} = [0\ 1\ -1]^T$ is

$$S = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Using this transformation and (3.1), directed graph $G_7 = (P_7, D_7)$ is mapped into directed multigraph $\Gamma_8 = (Q_8, \Delta_8)$, defined by

$$(3.9)\qquad PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k+N_2 \\ 1 \end{bmatrix}, \ \Delta_8 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 0 & \frac{1}{2} & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

for each $k = 1, 2, \ldots, N_3$. Based on (3.9), we conclude that the array SA8 consists of $\Omega = N_3$ PEs (see [3]). Since $\vec{e}_a^{\,2} = \frac{1}{2}\vec{e}_c^{\,2}$, the array SA8 is unidirectional (type III). The elements of vectors $\vec{A}_{1\bullet}$ and $\vec{C}_{1\bullet}$ are pipelined through the array in the same direction.

For the **Algorithm 3**, possible projection vectors are $\vec{\mu} = [1\,0\,0]^T$, $\vec{\mu} = [0\,1\,0]^T$, $\vec{\mu} = [1\,1\,0]^T$ and $\vec{\mu} = [1\,-1\,0]^T$. Directed graph $G_8 = (P_8, D_8)$ that corresponds to this algorithm is defined by

$$P_8 = \left\{\vec{p} = [\, i \;\; j \;\; 1\,]^T \mid 1 \leqslant i \leqslant N_1,\ 1 \leqslant j \leqslant N_2\right\}$$

$$D_8 = \begin{bmatrix} \vec{e}_b^{\,3} & \vec{e}_a^{\,3} & \vec{e}_c^{\,3} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

One of the valid transformation matrices for projection vector $\vec{\mu} = [1\,0\,0]^T$ is

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

According to (3.1), directed graph $G_8 = (P_8, D_8)$ is mapped into directed graph with loops $\Gamma_9 = (Q_9, \Delta_9)$, determined by

$$PE \to \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} j \\ 1 \end{bmatrix},\quad \Delta_9 = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.10}$$

for each $j = 1, 2, \ldots, N_3$. The corresponding systolic array, SA9, consists of $\Omega = N_2$ PEs (see [3]). Since $\vec{e}_b^{\,2} = [0\,0]^T$, the array SA9 is static (type I). The elements of vector $\vec{A}_{\bullet 1}$ are pipelined through the array.

For the projection vector $\vec{\mu} = [0\,1\,0]^T$, one of the valid transformations is

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Based on (3.1), directed graph $G_8 = (P_8, D_8)$ is mapped into directed graph with loops $\Gamma_{10} = (Q_{10}, \Delta_{10})$, defined by

$$PE \to \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} i \\ 1 \end{bmatrix},\quad \Delta_{10} = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.11}$$

for $i = 1, 2, \ldots, N_1$. The corresponding systolic array, SA10, consists of $\Omega = N_1$ PEs (see, [3]). Since $\vec{e}_a^{\,2} = [0\,0]^T$, array is static (type I). The elements of vector $\vec{B}_{1\bullet}$ are pipelined through the array.

For projection vector $\vec{\mu} = [1\,1\,0]^T$, we will consider the following two algorithms, equivalent to **Algorithm 3**, in order to minimize the corresponding arrays.

Algorithm 8
for $j := 1$ **to** N_2 **do**
for $i := 1$ **to** N_1 **do**

$b(i, i+j-1, 1) := b(i-1, i+j-1, 1)$
$a(i, i+j-1, 1) := a(i, i+j-2, 1)$
$c(i, i+j-1, 1) := c(i, i+j-1, 0) + a(i, i+j-1, 1)b(i, i+j-1, 1)$

where $b(0, t+N_2, 1) \equiv b(0, t, 1)$, $c(0, t+N_2, 1) \equiv c(0, t, 1)$, for each $i = 1, 2, \ldots, N_1$, $t = 1, 2, \ldots, N_2$.

Algorithm 9
for $j := 1$ **to** N_2 **do**
for $i := 1$ **to** N_1 **do**
$b(i+j-1, j, 1) := b(i+j-2, j, 1)$
$a(i+j-1, j, 1) := a(i+j-1, j-1, 1)$
$c(i+j-1, j, 1) := c(i+j-1, j, 0) + a(i+j-1, j, 1)b(i+j-1, j, 1)$

where $a(t+N_1, 0, 1) \equiv a(t, 0, 1)$, $c(t+N_1, j, 1) \equiv c(t, j, 1)$, for each $t = 1, 2, \ldots, N_1$, $j = 1, 2, \ldots, N_2$.

For **Algorithm 8** the corresponding directed graph, $G_9 = (P_9, D_9)$, is determined by

$$P_9 = \left\{\vec{p} = [\, i \quad i+j-1 \quad 1\,]^T \mid 1 \leqslant i \leqslant N_1,\ 1 \leqslant j \leqslant N_2\right\}$$

$$D_9 = \begin{bmatrix} \vec{e}_b^{\,3} & \vec{e}_a^{\,3} & \vec{e}_c^{\,3} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

One of the valid transformation matrices for projection vector $\vec{\mu} = [1\ 1\ 0]^T$ is

$$S = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Having in mind (3.1), directed graph $G_9 = (P_9, D_9)$ is mapped into directed multigraph $\Gamma_{11} = (Q_{11}, \Delta_{11})$, defined by

$$PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1-j \\ 1 \end{bmatrix},\ \Delta_{10} = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.12}$$

for $j = 1, 2, \ldots, N_2$. From (3.12) we conclude that the corresponding array, SA11, consists of $\Omega = N_2$ PEs (see [3]). Since $\vec{e}_b^{\,2} = -\vec{e}_a^{\,2}$, the array SA11 is bidirectional (type II). The elements of vectors $\vec{A}_{\bullet 1}$ and $\vec{B}_{1\bullet}$ are pipelined through the array in opposite directions.

For **Algorithm 9**, the corresponding directed graph, $G_{10} = (P_{10}, D_{10})$, is determined by

$$P_{10} = \left\{\vec{p} = [\, i+j-1 \quad j \quad 1\,]^T \mid 1 \leqslant i \leqslant N_1,\ 1 \leqslant j \leqslant N_2\right\}$$

$$D_{10} = \begin{bmatrix} \vec{e}_b^{\,3} & \vec{e}_a^{\,3} & \vec{e}_c^{\,3} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Using valid transformation

$$S = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

according to (3.1), directed graph $G_{10} = (P_{10}, D_{10})$ is mapped into directed multigraph $\Gamma_{12} = (Q_{12}, \Delta_{12})$, defined by

$$\text{(3.13)} \qquad PE \to \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} i-1 \\ 1 \end{bmatrix}, \ \Delta_{10} = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

for $i = 1, 2, \ldots, N_1$. From (3.13) we conclude that the corresponding systolic array, SA12, consists of $\Omega = N_1$ PEs (see [3]). Since $\vec{e}_b^{\,2} = -\vec{e}_a^{\,2}$, the array SA12 is bidirectional (type II). The elements of vectors $\vec{A}_{\bullet 1}$ and $\vec{B}_{1\bullet}$ are pipelined through the array in opposite directions.

The arrays SA11 and SA12 were designed in [4].

For the projection vector $\vec{\mu} = [1\ -1\ 0]^T$, instead of **Algorithm 3**, we use the following two equivalent algorithms:

Algorithm 10

for $j := 1$ **to** N_2 **do**

for $i := 1$ **to** N_1 **do**

$b(i - \frac{1}{2}, -i + j + N_1, 1) := b(i - 1, -i + j + N_1, 1)$

$b(i, -i + j + N_1, 1) := b(i - \frac{1}{2}, -i + j + N_1, 1)$

$a(i, -i + j + N_1, 1) := a(i, -i + j + N_1 - 1, 1)$

$c(i, -i + j + N_1, 1) := c(i, -i + j + N_1, 0) + a(i, -i + j + N_1, 1)b(i, -i + j + N_1, 1)$

where $b(i, t + N_2, 1) \equiv b(i, t, 1)$, $c(i, t + N_2, 1) \equiv c(i, t, 1)$, for each $i = 1, 2, \ldots, N_1$, and $t = 1, 2, \ldots, N_2$.

Algorithm 11

for $j := 1$ **to** N_2 **do**

for $i := 1$ **to** N_1 **do**

$a(i - j + N_2, j - \frac{1}{2}, 1) := a(i - j + N_2, j - 1, 1)$

$a(i - j + N_2, j, 1) := a(i - j + N_2, j - \frac{1}{2}, 1)$

$b(i - j + N_2, j, 1) := b(i - j + N_2 - 1, j, 1)$

$c(i - j + N_2, j, 1) := c(i - j + N_2, j, 0) + a(i - j + N_2, j, 1)b(i - j + N_2, j, 1)$

where $a(t + N_1, 0, 1) \equiv a(t, 0, 1)$, $c(t + N_1, j, 1) \equiv c(t, j, 1)$, for each $t = 1, 2, \ldots, N_1$, and $j = 1, 2, \ldots, N_2$.

Directed graph, $G_{11} = (P_{11} \cup \bar{P}_{11}, D_{11})$, that corresponds to **Algorithm 10** is defined by

$$P_{11} = \left\{\vec{p} = [\, i \quad -i + j + N_1 \quad 1\,]^T\right\},$$

$$\bar{P}_{11} = \left\{\vec{p} = \left[\, i - \frac{1}{2} \quad -i + j + N_1 \quad 1\,\right]^T\right\},$$

$$D_{11} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

For projection direction $\vec{\mu} = [1 \ -1 \ 0]^T$ a valid transformation is

$$S = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

According to this transformation and (3.1), directed graph $G_{11} = (P_{11} \cup \bar{P}_{11}, D_{11})$ is mapped into directed multigraph $\Gamma_{13} = (Q_{13} \cup \bar{Q}_{13}, \Delta_{13})$, defined by

$$PE \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} j + N_1 \\ 1 \end{bmatrix}, \ d \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} j + N_1 - \frac{1}{2} \\ 1 \end{bmatrix},$$

$$\Delta_{13} = [\ \vec{e}_b^{\,2} \quad \vec{e}_a^{\,2} \quad \vec{e}_c^{\,2}\ ]^T = \begin{bmatrix} \frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.14}$$

for $j = 1, 2, \ldots, N_2$. From (3.14) we conclude that corresponding systolic array, SA13, consists of $\Omega = N_2$ PEs (see [3]). Since $\vec{e}_b^{\,2} = \frac{1}{2}\vec{e}_a^{\,2}$, the array SA13 is unidirectional (type III). Elements of vectors $\vec{A}_{\bullet 1}$ and vector-row $\vec{B}_{1\bullet}$ are pipelined through the array in the same direction.

Directed graph, $G_{12} = (P_{12} \cup \bar{P}_{12}, D_{12})$, that corresponds to **Algorithm 11** is defined by

$$P_{12} = \left\{\vec{p} = [\, -i + j + N_2 \quad j \quad 1\,]^T\right\},$$

$$\bar{P}_{12} = \left\{\vec{p} = \left[\, -i + j + N_2 \quad j - \frac{1}{2} \quad 1\,\right]^T\right\},$$

$$D_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Using transformation matrix

$$S = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

according to (3.1), this graph is mapped into directed multigraph $\Gamma_{14} = (Q_{14} \cup \bar{Q}_{14}, \Delta_{14})$, defined by

$$PE \to \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} i+N_2 \\ 1 \end{bmatrix}, \ d \to \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} i+N_2-\frac{1}{2} \\ 1 \end{bmatrix},$$

$$\Delta_{14} = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix}^T = \begin{bmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.15}$$

for $i = 1, 2, \ldots, N_1$. From (3.15) we conclude that the corresponding systolic array SA14, consists of $\Omega = N_1$ PEs ([3]). Since $\vec{e}_a^{\,2} = \frac{1}{2}\vec{e}_b^{\,2}$, the SA14 is unidirectional (type III). The elements of vectors $\vec{A}_{\bullet 1}$ and $\vec{B}_{\bullet 1}$ are pipelined through the array in the same direction.

4. Discussion

Table 4. lists the arrays obtained in this paper, the algorithm used to design the corresponding array, type of the array, the number of PE and the used projection vector. It can be concluded that, regardless of the mutual relation between N_1, N_2 and N_3, it is always possible to derive a systolic algorithm which can be used to design a space-optimal systolic array of either of the three types.

SA	Algorithm	Type	$\vec{\mu}$	Ω
SA1	1	I	$\vec{\mu} = [1\ 0\ 0]^T$	N_3
SA2	1	I	$\vec{\mu} = [0\ 0\ 1]^T$	N_1
SA5	2	I	$\vec{\mu} = [0\ 1\ 0]^T$	N_3
SA6	2	I	$\vec{\mu} = [0\ 0\ 1]^T$	N_2
SA9	3	I	$\vec{\mu} = [1\ 0\ 0]^T$	N_2
SA10	3	I	$\vec{\mu} = [0\ 1\ 0]^T$	N_1
SA3	4	II	$\vec{\mu} = [1\ 0\ 1]^T$	N_3
SA7	6	II	$\vec{\mu} = [0\ 1\ 1]^T$	N_3
SA11	8	II	$\vec{\mu} = [1\ 1\ 0]^T$	N_2
SA12	9	II	$\vec{\mu} = [1\ 1\ 0]^T$	N_1
SA4	5	III	$\vec{\mu} = [1\ 0\ -1]^T$	N_3
SA8	7	III	$\vec{\mu} = [0\ 1\ -1]^T$	N_3
SA13	10	III	$\vec{\mu} = [1\ -1\ 0]^T$	N_2
SA14	11	III	$\vec{\mu} = [1\ -1\ 0]^T$	N_1

Table 4.1: Survey of the systolic arrays

Thus, for example, when $N_2 \geqslant \min\{N_1, N_3\}$, space-optima SA of type I is obtained according to **Algorithm 2** and **Algorithm 3** for projection vectors $\vec{\mu} = [0\ 0\ 1]^T$ and $\vec{\mu} = [1\ 0\ 0]^T$, respectively; a space-optimal bidirectional array of type II is obtained from **Algorithm 8** using projection vector $\vec{\mu} = [1\ 1\ 0]^T$; a space-optimal unidirectional array of type III is obtained from **Algorithm 10** using projection vector $\vec{\mu} = [1\ -1\ 0]^T$.

REFERENCES

1. M. P. Bekakos, E. I. Milovanović, N. M. Stojanović, T. I. Tokić and I. Ž. Milovanović: *Transformation matrices for systolic array synthesis*. J. Elecrotechn. Math. **7(1)** (2003), 9–15.
2. H.T. Kung: *Why systolic architecture?*. Computer **15(1)** (1982), 37–46.
3. I. Ž. Milovanović, T. I. Tokić, E. I. Milovanović and M. K. Stojčev: *Determining the Number of Processing Elements in Systolic Array*. Facta Univ. Ser. Math. Inor. **15(1)** (2000), 123–132.
4. I. Ž. Milovanović, E. I. Milovanović, B. M. Randjelović and I. Č. Jovanović: *Matrix multiplication on bidirectional linear systolic arrays*. FILOMAT **17(1)** (2003), 135–141.
5. I. Ž. Milovanović, M. P. Bekakos, I. N. Tselepis, and E. I. Milovanović: *Forty-three ways of systolic matrix multiplication*. J. Comput. Math. **87(6)** (2010), 164–176.
6. I. Ž. Milovanović, E. I. Milovanović and M. P. Bekakos: *Synthesis of a unidirectional systolic array for matrix-vector multiplication*. Math. Comput. Modeling **13(5-6)** (2006), 612–619.
7. E. I. Milovanović, M. P. Bekakos and I. Ž. Milovanović: *Synthesis of space optimal systolic arrays for band matrix-vector multiplication*. J. Supercomputing **49(3)** (2009), 269–290.
8. E. I. Milovanović, M. B. Tošić, I. Ž. Milovanović and I. Z. Milentijević: *Designing processor-time optimal linear systolic arrays for matrix-vector multiplication*. J. Elecrotechn. Math. **3(1)** (1999), 7–19.
9. I. Ž. Milovanović, E. I. Milovanović, I. Z. Milentijević and M. K. Stojčev: *Designing of processor time-optimal systolic arrays for band matrix-vector multiplication*. Comput. Math. Appl. **32(2)** (1996), 1–11.
10. D. I. Moldovan: *On the design of algorithms for VLSI systolic arrays*. IEEE Proc. **71(1)** (1983), 113–120.
11. N. M. Novaković, E. I. Milovanović, M. K. Stojčev, T. I. Tokić and I. Ž. Milovanović: *Optimization of bidirectional systolic arrays for matrix-vector multiplication*. J. Elecrotechn. Math. **4(1)** (2000), 35–40.
12. B. M. Randjelović, E. I. Milovanović, I. Ž. Milovanović and M. P. Bekakos: *Synthesis of regular bidirectional linear systolic arrays for matrix multiplication*. J. Tech. Univ. Gabrovo **11(1)** (2005), 98–102.
13. J. M. Ortega: *Introduction to parallel and vector solution of linear systems*. Plenum Press, New York and London, 1988.
14. S. G. Sedukhin: *The designing and analysis of systolic algorithms and structures*. Programming **2** (1991), 20-40.

15. C. R. Wan and D. J. Evans: *Nineteen ways of systolic matrix multiplication*. J. Comput. Math., **68(2)** (1998), 39–69.

Branislav M. Randjelović
Faculty of Electronic Engineering
Department of Mathematics
Aleksandra Medvedeva 14
18000 Niš, Serbia
branislav.randjelovic@elfak.ni.ac.rs

Emina I. Milovanović
Faculty of Electronic Engineering
Department of Computer Science
Aleksandra Medvedeva 14
18000 Niš, Serbia
emina.milovanovic@elfak.ni.ac.rs

Igor Ž. Milovanović
Faculty of Electronic Engineering
Department of Mathematics
Aleksandra Medvedeva 14
18000 Niš, Serbia
igor.milovanovic@elfak.ni.ac.rs