FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. 21 (2006), 87–92

SYSTOLIC MATRIX VECTOR ITERATIONS*

E. I. Milovanović, M. P. Bekakos,
I. Ž. Milovanović, T. Z. Mirković

Abstract. In this paper we consider computation of iterative process of the form $\vec{x}^{(t)} = A \cdot \vec{x}^{(t-1)}$, where $A = (a_{ik})$ is $n \times n$ dense matrix on unidirectional linear systolic array.

1. Introduction

We consider the systolic implementation of iterative method defined by

$$\vec{x}^{(t)} = A\vec{x}^{(t-1)}, \quad t = 1, 2, \dots, m, \tag{1.1}$$

where $A = (a_{ik})$ is a dense square matrix of order $n \times n$,

$$\vec{x}^{(0)} = [x_1^{(0)} \ x_2^{(0)} \ \dots \ x_n^{(0)}]^T$$

is a given initial vector, and $m \gg 1$. The main building block for the systolic implementation of matrix vector iterations is a systolic array for matrix vector multiplication. The computation task can be expanded either in space, leading to series of cascaded linear systolic arrays performing pipelined iterations, or in time leading to an iterative systolic structure with feedback control [2]. A combination of the two approaches is possible, also.

In this paper we consider implementation of matrix vector iterations on linear unidirectional systolic array (ULSA) for matrix vector multiplication

Received January 10, 2006

2000 *Mathematics Subject Classification.* 68M07, 68Q35.

*The authors were supported in part by the Serbian Ministry of Science and Environmental Protection (Project: #144034: Parallel Methods and Algorithms in Discrete Mathematics).

with feedback connection. We give an explicit formulas to design ULSA which is space optimal, i.e. has the optimal number od processing elements (PE) for a given matrix dimension and minimal execution time for such number of PEs. The iteration process (1.1) is implemented on the ULSA for matrix vector multiplication in m iterations, such that $\vec{x}^{(t-1)}$ represents input for the computation of $\vec{x}^{(t)}$, for $t = 1, 2, \dots, m$. Data schedule is defined such that input time of t-th iteration is overlapped with computation of $(t-1)$-st iteration, for $t = 2, 3, \dots, m$, reducing the overall computation time. The efficiency of the proposed method is considered, also.

The problem of computing matrix vector iterations on the BLSA (bidirectional linear SA) was considered in [1] and on the ULSA in [2]. However, explicit formulas for the synthesis of these arrays are not given. In both cases matrix A was a band matrix with band width less than the dimension of matrix which enables obtaining space optimal arrays. If width of matrix band is greater than matrix dimension or full matrix is considered than obtained arrays are not space optimal.

2. Implementation of Matrix Vector Iterations on the ULSA

The space optimal ULSA that computes product of matrix $A = (a_{ik})$ of order $n \times n$ and vector $\vec{b} = [b_1\ b_2\ \dots\ b_n]^T$ was designed in [3]. It consists of $\Omega = n$ PEs and computes the matrix vector product $\vec{c} = A\vec{b}$ for time $T_{\text{tot}} = 3n - 2$. An example of this array, for the case $n = 4$, is given in Fig. 1.

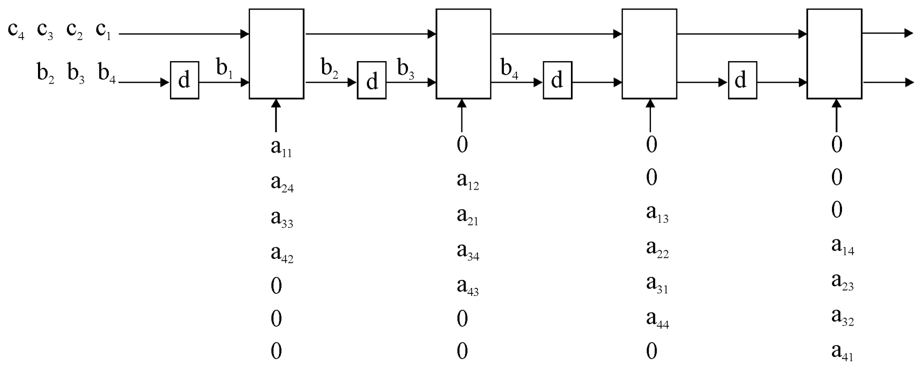

FIG. 1: Data flow in the ULSA synthesized in [3]

In order to compute matrix vector iterations (1.1) on the ULSA from [3] efficiently, we need to reorder data schedule. Namely, elements of vectors

$\vec{c}$ and $\vec{b}$ enter the ULSA in mutually reverse order. Since, in the iterative process elements of resulting vector $\vec{x}^{(t)}$ form t-th iteration will appear as inputs $\vec{x}^{(t-1)}$ in the next iteration, it is desirable that both $\vec{x}^{(t)}$ and $\vec{x}^{(t-1)}$ enter/leave the array in the same order. If opposite, we need to reorder resulting elements $\vec{x}^{(t)}$ before each new iteration. Second concerns the position of delay elements in the ULSA. Namely, there are two alternatives. Either to first delay elements of $\vec{b}$ and then use it in the computation, or use it first, then delay. The second solution is better since it will shorten the total execution time for $m-1$ time units, which in the case $m \gg 1$ is not negligible.

In order to obtain equal data schedule for $\vec{x}^{(t)}$ and $\vec{x}^{(t-1)}$, i.e., $\vec{c}$ and $\vec{b}$, we will use matrix $\hat{I}$ of order $n \times n$ defined by

$$\hat{I} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & & & \\ 1 & \cdots & 0 & 0 \end{bmatrix}.$$

Matrix $\hat{I}$ has some interesting properties which we will use to obtain desired data schedule. First, $\hat{I} \cdot \hat{I} = I$. Second, for each vector $\vec{b} = [b_1 \ b_2 \ \ldots \ b_n]^T$ and matrix $A = (a_{ik})$ of order $n \times n$, the following equalities are valid: $\vec{b}^* = \hat{I}\vec{b} = [b_n \ b_{n-1} \ \ldots \ b_1]^T$, and $A^* = A \cdot \hat{I} = (a_{i,n-k+1})$. In other words vector $\vec{b}^*$ and matrix A^* are mirror symmetric to $\vec{b}$ and A, respectively. According to this properties we have that

$$\vec{c} = A \cdot \vec{b} = (A \cdot \hat{I}) \cdot (\hat{I} \cdot \vec{b}). \tag{2.1}$$

Now, denote with

$$PE \mapsto \begin{bmatrix} x \\ y \end{bmatrix}, \quad d \mapsto \begin{bmatrix} x \\ y \end{bmatrix}, \quad \gamma(\cdot,\cdot,\cdot) \mapsto \begin{bmatrix} x \\ y \end{bmatrix},$$

$\gamma \in \{a, b, c\}$, (x, y) coordinates of the PEs, delay elements and data elements, respectively.

According to the above properties we have

$$PE \mapsto \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k+1 \\ 1 \end{bmatrix},$$

$$d \mapsto \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k+3/2 \\ 1 \end{bmatrix},$$

$$a(i,0,n-k+1) \mapsto \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k+1 \\ 2-i-k \end{bmatrix}, \tag{2.2}$$

$$b(0,1,n-k+i) \mapsto \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (3+k-i)/2 \\ 1 \end{bmatrix},$$

$$c(i,1,0) \mapsto \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2-i \\ 1 \end{bmatrix},$$

for $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$, and

$$a(i,0,t+n) \equiv a(i,0,t) \equiv a_{it}, \qquad b(0,1,t+n) \equiv b(0,1,t) \equiv b_t.$$

The communication links in the ULSA are implemented along the propagation vectors

$$\Delta = \begin{bmatrix} \vec{e}_b^{\,2} & \vec{e}_a^{\,2} & \vec{e}_c^{\,2} \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{2.3}$$

Data schedule in the ULSA synthesized according to (2.2) and (2.3) at the beginning of the computation, for the case $n = 4$, is depicted in Fig. 2.

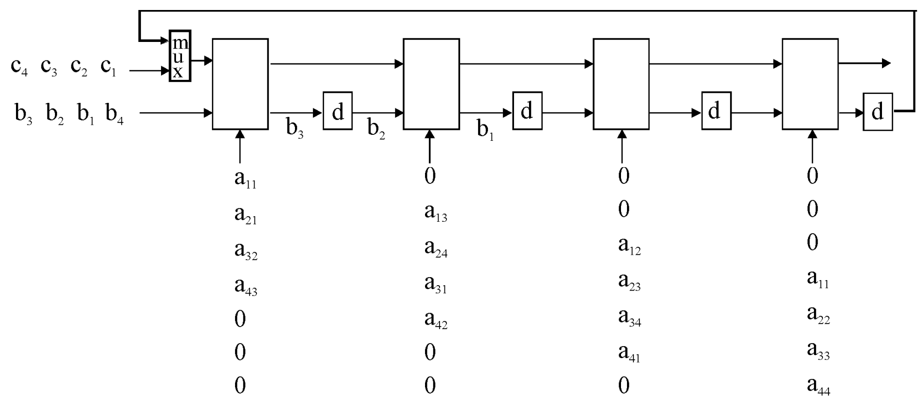

FIG. 2: Data flow in the ULSA synthesized according to (2.2) and (2.3) for $n = 4$

Table 1 shows the complete, step by step, diagram for computing matrix vector iterations on the ULSA for the case $n = 3$ and $m = 3$.

Clk	PE1	d	PE2	d	PE3	d
1	$0 := 0 + 0 \cdot x_1^{(0)}$					
2	$0 := 0 + 0 \cdot x_2^{(0)}$	$x_1^{(0)}$				
3	$x_1^{(1)} := 0 + a_{13}x_3^{(0)}$	$x_2^{(0)}$	$0 := 0 + 0 \cdot x_1^{(0)}$			
4	$x_2^{(1)} := 0 + a_{21}x_1^{(0)}$	$x_3^{(0)}$	$x_1^{(1)} := x_1^{(1)} + a_{12}x_2^{(0)}$	$x_1^{(0)}$		
5	$x_3^{(1)} := 0 + a_{32}x_2^{(0)}$	$x_1^{(0)}$	$x_2^{(1)} := x_2^{(1)} + a_{23}x_3^{(0)}$	$x_2^{(0)}$	$x_1^{(1)} := x_1^{(1)} + a_{11}x_1^{(0)}$	
6	$0 := 0 + 0 \cdot x_1^{(1)}$	$x_2^{(0)}$	$x_3^{(1)} := x_3^{(1)} + a_{31}x_1^{(0)}$	$x_3^{(0)}$	$x_2^{(1)} := x_2^{(1)} + a_{22}x_2^{(0)}$	$x_1^{(0)}$
7	$0 := 0 + 0 \cdot x_2^{(1)}$	$x_1^{(1)}$	$0 := 0 + 0 \cdot x_2^{(0)}$	$x_1^{(0)}$	$x_3^{(0)} := x_3^{(1)} + a_{33}x_3^{(0)}$	$x_2^{(0)}$
8	$x_1^{(2)} := 0 + a_{13}x_3^{(1)}$	$x_2^{(1)}$	$0 := 0 + 0 \cdot x_1^{(1)}$	$x_2^{(0)}$	$0 := 0 + 0 \cdot x_1^{(0)}$	$x_3^{(0)}$
9	$x_2^{(2)} := 0 + a_{21}x_1^{(1)}$	$x_3^{(1)}$	$x_1^{(2)} := x_1^{(2)} + a_{12}x_2^{(1)}$	$x_1^{(1)}$	$0 := 0 + 0 \cdot x_2^{(0)}$	$x_1^{(0)}$
10	$x_3^{(2)} := 0 + a_{32}x_2^{(1)}$	$x_1^{(1)}$	$x_2^{(2)} := x_2^{(2)} + a_{23}x_3^{(1)}$	$x_2^{(1)}$	$x_1^{(2)} := x_1^{(2)} + a_{11}x_1^{(1)}$	$x_2^{(0)}$
11	$0 := 0 + 0 \cdot x_1^{(2)}$	$x_2^{(1)}$	$x_3^{(2)} := x_3^{(2)} + a_{31}x_1^{(1)}$	$x_3^{(1)}$	$x_2^{(2)} := x_2^{(2)} + a_{22}x_2^{(1)}$	$x_1^{(1)}$
12	$0 := 0 + 0 \cdot x_2^{(2)}$	$x_1^{(2)}$	$0 := 0 + 0 \cdot x_2^{(1)}$	$x_1^{(1)}$	$x_3^{(2)} := x_3^{(2)} + a_{33}x_3^{(1)}$	$x_2^{(1)}$
13	$x_1^{(3)} := 0 + a_{13}x_3^{(2)}$	$x_2^{(2)}$	$0 := 0 + 0 \cdot x_1^{(2)}$	$x_2^{(1)}$	$0 := 0 + 0 \cdot x_1^{(1)}$	$x_3^{(1)}$
14	$x_2^{(3)} := 0 + a_{21}x_1^{(2)}$	$x_3^{(2)}$	$x_1^{(3)} := x_1^{(3)} + a_{12}x_2^{(2)}$	$x_1^{(2)}$	$0 := 0 + 0 \cdot x_2^{(1)}$	$x_1^{(1)}$
15	$x_3^{(3)} := 0 + a_{32}x_2^{(2)}$	$x_1^{(2)}$	$x_2^{(3)} := x_2^{(3)} + a_{23}x_3^{(2)}$	$x_2^{(2)}$	$x_1^{(3)} := x_1^{(3)} + a_{11}x_1^{(2)}$	$x_2^{(1)}$
16	$0 := 0 + 0 \cdot x_1^{(3)}$	$x_2^{(2)}$	$x_3^{(3)} := x_3^{(3)} + a_{31}x_1^{(2)}$	$x_3^{(2)}$	$x_2^{(3)} := x_2^{(3)} + a_{22}x_2^{(2)}$	$x_1^{(2)}$
17	$0 := 0 + 0 \cdot x_2^{(3)}$	$x_1^{(3)}$	$0 := 0 + 0 \cdot x_2^{(2)}$	$x_1^{(2)}$	$x_3^{(3)} := x_3^{(3)} + a_{33}x_3^{(2)}$	$x_2^{(2)}$
18	$x_1^{(4)} := 0 + a_{13}x_3^{(3)}$	$x_2^{(3)}$	$0 := 0 + 0 \cdot x_1^{(3)}$	$x_2^{(2)}$	$0 := 0 + 0 \cdot x_1^{(2)}$	$x_3^{(2)}$
19	$x_2^{(4)} := 0 + a_{21}x_1^{(3)}$	$x_3^{(3)}$	$x_1^{(4)} := x_1^{(4)} + a_{12}x_2^{(3)}$	$x_1^{(3)}$	$0 := 0 + 0 \cdot x_2^{(2)}$	$x_1^{(2)}$
20	$x_3^{(4)} := 0 + a_{32}x_2^{(3)}$	$x_1^{(3)}$	$x_2^{(4)} := x_2^{(4)} + a_{23}x_3^{(3)}$	$x_2^{(3)}$	$x_1^{(4)} := x_1^{(4)} + a_{11}x_1^{(3)}$	$x_2^{(2)}$
21	$0 := 0 + 0 \cdot 0$	$x_2^{(3)}$	$x_3^{(4)} := x_3^{(4)} + a_{31}x_1^{(3)}$	$x_3^{(3)}$	$x_2^{(4)} := x_2^{(4)} + a_{22}x_2^{(3)}$	$x_1^{(3)}$
22	$0 := 0 + 0 \cdot 0$	0	$0 := 0 + 0 \cdot x_2^{(3)}$	$x_1^{(3)}$	$x_3^{(4)} := x_3^{(4)} + a_{33}x_3^{(3)}$	$x_2^{(3)}$

TABLE 1: Step by step diagram for computing matrix vector iteration $x^{(t)} = Ax^{(t-1)}$, $t = 1, 2, \ldots m$ on the ULSA for the case $n = 3$ and $m = 4$

3. Performance Analysis

According to (2.2) we conclude that ULSA consists of $\Omega = n$ PEs, which is optimal number for a given dimension of matrix A.

Computation time for one iteration on the ULSA is $t_{\text{tot}} = t_{\text{in}} + t_{\text{exe}} + t_{\text{out}} = 3n - 2$, where $t_{\text{in}} = n - 1$ is the initialization time, $t_{\text{exe}} = 2n - 1$ is active computation time and $t_{\text{out}} = 0$ is output time. The important property of the proposed data schedule is that initialization time of the t-th iteration (i.e., $x^{(t)}$) is overlapped with active computation time of the $(t-1)$-st iteration (i.e., $x^{(t-1)}$), for $t = 1, 2, \ldots, m$. Having this in mind, the time needed to compute $x^{(m)}$ on the ULSA is

$$T_{\text{tot}} = t_{\text{in}} + (m-1)t_{\text{exe}} = m(2n-1) + n - 1 = (2m+1)n - m - 1.$$

The efficiency of the array depends on the relation between number of iterations m and matrix dimension n. The values of the efficiency for various m and n are summarized in Table 2. From Table 2 we conclude that for some fixed m the efficiency decreases as n increases. However, reducing of the efficiency is not a substantial, and all values are grouped around 0.5 even for relatively small n ($n \geq 50$) and $m \geq 20$. Therefore, it is convenient to consider a realization of the matrix vector iterations (1.1) on the fixed size ULSA. If n is fixed on some constant value the efficiency increases as m increases. The efficiency is greater than 0.5 for all $m \geq n-1$. However, for $n \geq 10$ the efficiency is around 0.5 regardless to the number of iterations m.

mn	2	5	10	50	100	1000	$n \to +\infty$
1	0.50	0.38	0.36	0.34	0.34	0.33	0.33
5	0.62	0.51	0.49	0.46	0.46	0.45	0.45
10	0.65	0.53	0.50	0.48	0.48	0.48	0.48
50	0.66	0.55	0.52	0.50	0.50	0.50	0.50
100	0.66	0.55	0.52	0.50	0.50	0.50	0.50
1000	0.67	0.55	0.53	0.50	0.50	0.50	0.50
$m \to +\infty$	0.67	0.55	0.53	0.51	0.50	0.50	0.50

Table 2: Efficiency of the ULSA for various m and n

REFERENCES

1. K. G. Margaritis, D. J. Evans: *Folding techniques for systolic iterations.* Parallel Algorithms Appl. **7** (1995), 87–105.
2. D. J. Evans, K. G. Margaritis: *Systolic designs for eigenvalue-eigenvector computations using matrix powers.* Parallel Comput. **14** (1990), 77–88.
3. I. Ž. Milovanović, E. I. Milovanović, M. P. Bekakos: *Synthesis of a unidirectional systolic array for matrix-vector multiplication.* Math. Comput. Modelling **43**, 5-6 (2006), 612–619.

Faculty of Electronic Engineering
Department of Computer Science
P.O. Box 73
18000 Niš, Serbia
ema@elfak.ni.ac.yu (E.I. Milovanović)

Faculty of Electronic Engineering
Department of Mathematics
P.O. Box 73
18000 Niš, Serbia
igor@elfak.ni.ac.yu (I.Ž. Milovanović)

Faculty of Science and Mathematics
Department of Mathematics
38220 Kosovska Mitrovica, Serbia