Pergamon

Computers Math. Applic. Vol. 32, No. 12, pp. 77–91, 1996

0898-1221/96 $15.00 + 0.00

PII: S0898-1221(96)00208-8

A Systolic Algorithm for Solving Dense Linear Systems

CHAU-JY LIN
Department of Applied Mathematics, National Chiao-Tung University
Hsinchu, 30050, Taiwan, R.O.C.
cjlin@math.nctu.edu.tw

(Received and accepted May 1996)

Abstract—For an arbitrary $n \times n$ matrix A and an $n \times 1$ column vector b, we present a systolic algorithm to solve the dense linear equations $Ax = b$. An important consideration is that the pivot row can be changed during the execution of our systolic algorithm. The computational model consists of n linear systolic arrays. For $1 \leq i \leq n$, the i^{th} linear array is responsible to eliminate the i^{th} unknown variable x_i of x. This algorithm requires $4n$ time steps to solve the linear system. The elapsed time unit within a time step is independent of the problem size n. Since the structure of a PE is simple and the same type PE executes the identical instructions, it is very suitable for VLSI implementation. The design process and correctness proof are considered in detail. Moreover, this algorithm can detect whether A is singular or not.

Keywords—Parallel computer, Linear array, Systolis algorithm, Dense linear system.

1. INTRODUCTION

Parallel computers have been used to solve many problems in the fields of sciences and engineering. Systolic array is one of parallel computers. An algorithm which can be executed on a systolic array is called a *systolic algorithm*. The systolic array has been widely used to solve various problems because of its regular structure, simple interconnection, and feasibility for VLSI implementation [1–5]. Some useful discussions of systolic arrays and systolic algorithms can be referred to the papers in [6–8].

Given an arbitrary $n \times n$ matrix $A = (a_{ij})$ and an $n \times 1$ column vector $b = (b_i)$, the solution of the linear system $Ax = b$ is one of major problems in computational and applied mathematics. For solving $Ax = b$, under the elementary row operation on A, a sequential algorithm is always required to find an i^{th} pivot row such that this row possesses the largest absolute value among the i^{th} column of A. This partial pivoting method is not easy to be accomplished within a parallel algorithm. Thus, many parallel algorithms to solve $Ax = b$ require some assumptions, such as A is nonsingular that the diagonal entries of A are nonzero, and that the problem relating to pivoting is not considered [9–15].

When A is a nonsingular matrix, the LU decomposition is a very useful method to solve $Ax = b$. This method obtains a triangular matrix from A followed by the back substitution. That is, the solution of $Ax = b$ comes from the solutions of two triangular systems $Ly = b$

This work was supported by the National Science Council, Taiwan, R.O.C., under the contract number: NSC 84-2121-M009-012.

and $Ux = y$. However, it is possible that there exist many nonsingular matrices in which the LU decomposition is not easy to do. For example, a zero will appear in the diagonal of A when the LU decomposition is in progress. In this article, without any assumption on the given matrix A, we present a systolic algorithm to obtain the solution of $Ax = b$. Under our method, an existing pivot row can be replaced by a new pivot row during the execution of the elementary row operation on A. If A is a singular matrix, then we can find a row or a column such that it has all zero entries.

The computational model used to solve $Ax = b$ is a two-dimensional systolic array which consists of n linear systolic arrays. Each linear array is designed by the same consideration. Thus, these n linear arrays have similar structure and execute identical instructions. Every linear array has n equations as input data and also has n equations as output data. For $1 \leq i \leq n$, the i^{th} linear array is responsible to eliminate the i^{th} unknown variable x_i of x. This i^{th} linear array deletes $(n-1)$ coefficients of x_i and remains the value of 1 as the coefficient of x_i in the last equation. But this coefficient 1 of x_i is not involved into the work of the following $(n-i)$ linear systolic arrays which are used to eliminate the unknown variables x_k for $i+1 \leq k \leq n$, respectively. Hence, when we perform the instructions on the i^{th} linear array, the unknown variables x_l for $1 \leq l \leq i-1$ are all ignored. This design consideration implies that the back substitution which is followed the LU decomposition is unnecessary in our systolic algorithm.

2. AN OVERVIEW OF SYSTOLIC ARRAY

A systolic array consists of many simple structure PEs (processing elements) such that the same type PE executes the same instructions. Each PE only can communicate data with its neighboring PEs. Suppose that PE1 and PE2 are two PEs in a systolic array. If it is necessary to transfer data from PE1 to PE2, then there exists a communication link, say ξ-link, joining PE1 to PE2. The data sending out by PE1 on ξ-link is denoted as ξ_{out} of PE1. The data receiving by PE2 from ξ-link is denoted as ξ_{in} of PE2. This ξ-link is also considered as an output link of PE1 and an input link of PE2.

In a systolic array, a *time step* is considered as an enough large elapsed time unit such that all PEs can perform the following three tasks.

(α) The PE reads data from its input links.
(β) The PE executes the designed algorithm exactly once loop.
(γ) The PE sends out data to its output links.

In our algorithm, the elapsed time unit within a time step is independent of the problem size n.

If a ξ-link from PE1 to PE2 has a delay symbol δD, then the ξ_{out} sending out by PE1 at a time step t will be the ξ_{in} of PE2 at the time step $t+\delta$. In our systolic array, each link has only one delay, that is, $\delta = 1$. Thus, we omit the delay symbol in our systolic array. The condition of $\delta \neq 0$ means that the behavior of data broadcasting is not allowed within our systolic array.

3. THE DESIGN CONSIDERATION OF A LINEAR SYSTOLIC ARRAY

For a fixed integer i such that $1 \leq i \leq n$, we design the i^{th} linear systolic array to eliminate the i^{th} unknown variable x_i of x. The i^{th} linear array consists of $(n-i+2)$ PEs and three communication links with names a-link, d-link, and c-link. See Figure 1. These PEs are indexed as PE(i,j) for $i \leq j \leq n+1$. The d-link and c-link are used to send data from PE(i,j) to PE$(i,j+1)$. The input a-link of PE(i,j) is used to receive data from the $(i-1)^{\text{th}}$ linear array. The output a-link of PE(i,j) is used to send data to the $(i+1)^{\text{th}}$ linear array.

We classify our PEs into two types. The PE(i,i) is in the type I and the remaining PEs are in the type II. The structures of PEs are depicted in Figure 2. Each PE contains a register R. The type I PE has a more register P. The PE(i,i) is responsible to find a pivot row. When a pivot row had found, the type II PEs update the entries of another rows.

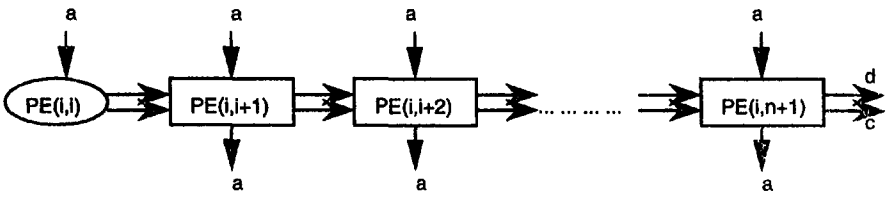


Figure 1. The i^{th} linear systolic array.

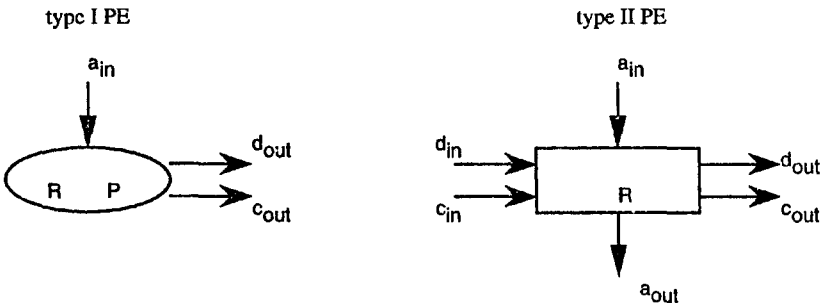


Figure 2. The structure of PEs.

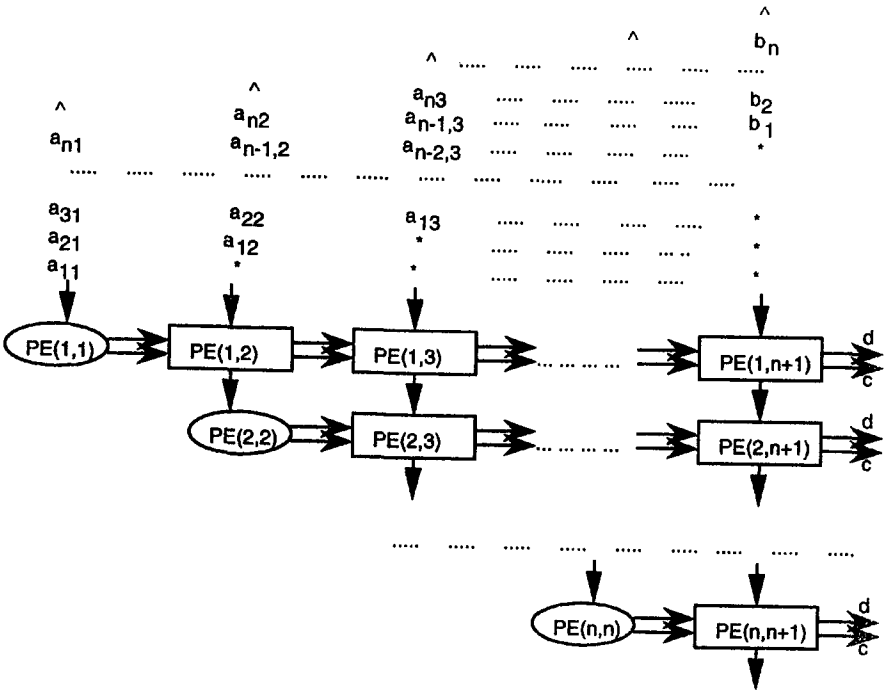


Figure 3. The two-dimensional systolic array.

From the above discussion, we obtan n linear systolic arrays. These n linear arrays are connected to form a two-dimensional array as shown in Figure 3, where the k^{th} column of A will be arranged to meet PE$(1,k)$ for $1 \leq k \leq n$ and the column b will be arranged to meet PE$(1,n+1)$. We use the symbols "$*$" and "$\hat{}$" to mean a waiting and a stopping signal, respectively.

Let $[A^{(0)}, b^{(0)}]$ be the $n \times (n+1)$ augmented matrix by adjoining b to A. At the beginning of our algorithm, the matrix $[A^{(0)}, b^{(0)}]$ is the input values of the a-links on the first linear array. The output data, except the symbol "$*$," which comes from the a-links of the i^{th} linear array will be formed and denoted as the matrix $[A^{(i)}, b^{(i)}]$. According to the order of the values appearing on the output a-links of the i^{th} linear array, the row indexes in $[A^{(i)}, b^{(i)}]$ are numbered sequentially as $i+1,\ i+2, \ldots, n,\ 1, 2, \ldots, i$.That is, the first a_{out} of PE(i,j) has the row index $(i+1)$ in

$[A^{(i)}, b^{(i)}]$ and the last a_{out} of PE(i,j) has the row index i in $[A^{(i)}, b^{(i)}]$. Of couse, this matrix $[A^{(i)}, b^{(i)}]$ will be the input data of the $(i+1)^{\text{th}}$ linear array.

Note that the matrix $[A^{(i)}, b^{(i)}]$ is an $n \times (n-i+1)$ matrix since PE(i,i) has no output a-link. The absence of output a-link in PE(i,i) causes the i^{th} unknown variable x_i to be deleted. The solution of $Ax = b$ will appear on the a_{out} of PE$(n, n+1)$ which is on the n^{th} linear array.

In our systolic array, the d-link is only used to carry the a_{in} of PE(i,i) to PE(i,j), for all $j > i$. The major work of the c-link is to indicate whether a new pivot row is found or not. At the initial state, we assign $(n-i+1)$ to the register P of PE(i,i). When the first pivot row is found, we reset P as $1-P$. This value of $1-P$ indicates how many remaining rows will be tested as a pivot row. The register R of PE(i,j) contains an entry of the current pivot row. At the initial state, we set the symbol "$*$" into the register R of all PE(i,j).

4. THE INSTRUCTIONS OF PES

For $1 \le i \le n$ and $i \le j \le n+1$, we will present the major instructions of PE(i,j). All PEs perform their instructions until the signal "$\hat{}$" appears on their input a-links. First, we consider the work of type I PE.

(1) The main purpose of PE(i,i) is to find the first pivot row by detecting its first nonzero value of a_{in}. Once PE(i,i) has found its first pivot row, PE(i,i) performs the following three tasks.
   (α) PE(i,i) sets its $P = 1-P$.
   (β) PE(i,i) assigns $|a_{\text{in}}|$, the absolute value of a_{in}, into its R.
   (γ) PE(i,i) sends a_{in} to its d-link.

(2) If PE(i,i) has its $P > 0$ and $a_{\text{in}} = 0$, then PE(i,i) is still on the state of finding the first pivot row. That is, the first pivot row is not appeared until now. In this case, PE(i,i) decreases one from P and PE(i,i) continues its search of the first pivot row. At any time step, once PE(i,i) has its $a_{\text{in}} = 0$, PE(i,i) sets its $c_{\text{out}} = 1$ in order to detect whether there exists a zero row in the matrix $A^{(i-1)}$. The existence of a zero row in $A^{(i-1)}$ causes the singularity of A.

(3) If there exists no pivot row, we obtain $P = 0$ and $R = *$ in PE(i,i). That is, the first column of $A^{(i-1)}$ will be appeared as a zero column. In this case, PE(i,i) announces that A is a singular matrix by sending the message of its $c_{\text{out}} = 3$.

(4) After PE(i,i) had found a pivot row and PE(i,i) has its $P < 0$, if PE(i,i) has $a_{\text{in}} \neq 0$ again, then PE(i,i) tries to replace the existing pivot row. When $|a_{\text{in}}| > R$, the action of exchanging two pivot rows will be performed. This exchanged message is transferred by a value of 2 on the c-link.

Now we consider the major work of the type II PE. These PEs will be used to modify the entries of $[A^{(i-1)}, b^{(i-1)}]$ into the entries of the matrix $[A^{(i)}, b^{(i)}]$.

(5) PE(i,j) always sends its d_{in} to d_{out}.

(6) If PE(i,j) knows that PE(i,i) had found its first pivot row, then PE(i,j) assigns the value of $a_{\text{in}}/d_{\text{in}}$ to its R. At this same time step, PE(i,j) sends out the symbol "$*$" to its a_{out}.

(7) If PE(i,j) has its $c_{\text{in}} = 0$ and $R \neq *$, then PE(i,j) modifies the value of its a_{in} into the value of its a_{out}.

(8) When PE(i,j) has its $c_{\text{in}} = 1$, PE(i,j) detects whether its $a_{\text{in}} = 0$. If $a_{\text{in}} = 0$, then PE(i,j) passes its c_{in} to c_{out}. Otherwise, if $a_{\text{in}} \neq 0$, PE(i,j) sets its $c_{\text{out}} = 0$. This work detects whether there exists a zero row in $A^{(i-1)}$. When PE(i,n) has its $c_{\text{out}} = 1$, we know that A is a singular matrix because of a zero row in the matrix $A^{(i-1)}$.

(9) If PE(i,j) has its $c_{\text{in}} = 2$, then the action of exchanging two pivot rows is performed. The PE(i,j) retrieves the content of R to be modified into its a_{out} and PE(i,j) stores the new pivot entry $a_{\text{in}}/d_{\text{in}}$ into its register R.

(10) When PE(i,j) has $a_{\text{in}} = \hat{}$, PE(i,j) sends the content of its R to its a_{out} and PE(i,j)

resets R as a special symbol "&" in order to send the signal " ^ " to its a_{out} at the next time step.

5. THE SYSTOLIC ALGORITHM

First of all, we define nine procedures. The first five procedures are used in the type I PE. The last four procedures are used for the type II PE.

procedure *testing-first-pivot* $\equiv$
  if $P = 1$ **then** *sending-singular-matrix* **else** *searching-first-pivot.*
procedure *sending-singular-matrix* $\equiv c_{\text{out}} = 3$; $R = \hat{}$.
procedure *searching-first-pivot* $\equiv P = P - 1$; $c_{\text{out}} = 1$.
procedure *finding-first-pivot* $\equiv P = 1 - P$; $R = |a_{\text{in}}|$; $c_{\text{out}} = 0$.
procedure *trying-new-pivot* $\equiv$
  $P = P + 1$;
  if $|a_{\text{in}}| > R$ **then** $\{R = |a_{\text{in}}|; c_{\text{out}} = 2\}$ **else** { **if** $a_{\text{in}} = 0$ **then** $c_{\text{out}} = 1$ **else** $c_{\text{out}} = 0$.}
procedure *sending-last-element* $\equiv a_{\text{out}} = R$; $R = \&$.
procedure *exchanging-two-pivots* $\equiv a_{\text{out}} = R - a_{\text{in}}/d_{\text{in}}$; $R = a_{\text{in}}/d_{\text{in}}$.
procedure *modifying-a-link* $\equiv$
  if $R = *$ **then** *storing-first-pivot* **else** $a_{\text{out}} = a_{\text{in}} - R * d_{\text{in}}$.
procedure *storing-first-pivot* $\equiv$ **if** $d_{\text{in}} \neq 0$ **then** $\{R = a_{\text{in}}/d_{\text{in}}; a_{\text{out}} = *\}$ **else** $a_{\text{out}} = a_{\text{in}}$.

ALGORITHM. **LINEAR-SOLVER** $(A, b, n) \equiv$

Initial state:

For $1 \leq i \leq n$ and $i \leq j \leq (n+1)$, we set $R = *$ in all PEs. Set $P = n - i + 1$ in PE(i, i). The entry $a_{i,j}$ of A meets PE$(1, j)$ at the time step $t = j + i - 1$. The entry b_i of b meets PE$(1, n+1)$ at the time step $t = n + i$. A stopping signal " ^ " will meet PE$(1, j)$ at the time step $t = n + j$. All the remaining links are denoted by the symbol "*."

Executive state:
repeat
  /* **do parallel for all PEs of type I.** */
    $d_{\text{out}} = a_{\text{in}}$;
    if $a_{\text{in}} = *$ **then** $\{c_{\text{out}} = *$; **break**$\}$;
    if $a_{\text{in}} = \hat{}$ **then** $\{R = \hat{}\,; c_{\text{out}} = \hat{}\,$; **break** $\}$;
    if $P = 0$ **then** $c_{\text{out}} = 0$;
    if $P < 0$ **then** *trying-new-pivot;*
    if $P > 0$ **then** { **if** $a_{\text{in}} = 0$ **then** *testing-first-pivot* **else** *finding-first-pivot*$\}$;
  /* **do parallel for all PEs of type II.** */
    $d_{\text{out}} = d_{\text{in}}$;
    if ($c_{\text{in}} = 1$ **and** $a_{\text{in}} \neq 0$) **then** $c_{\text{out}} = 0$ **else** $c_{\text{out}} = c_{\text{in}}$;
    if $a_{\text{in}} = *$ **then** $\{a_{\text{out}} = *$; **break;**$\}$
    if ($R = \&$ **or** $c_{\text{in}} = 3$) **then** $\{R = \hat{}\,; a_{\text{out}} = \hat{}\,$; **break;**$\}$
    if $a_{\text{in}} = \hat{}$ **then** { *sending-last-element;* **break;**$\}$
    if $c_{\text{in}} = 2$ **then** *exchanging-two-pivots;*
    if ($c_{\text{in}} = 0$ **or** $c_{\text{in}} = 1$) **then** *modifying-a-link;*
until $R = \hat{}$.

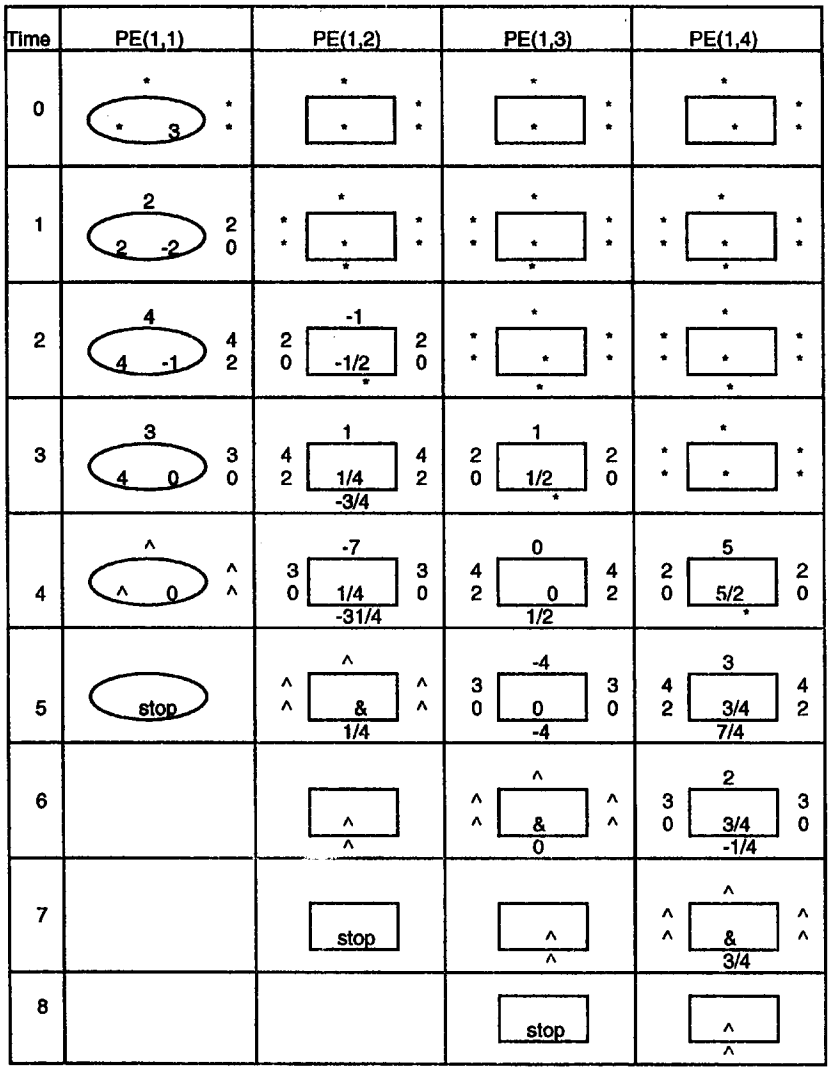
6. AN ILLUSTRATIVE EXAMPLE

We give an example with

$$[A, b] = \left[A^{(0)}, b^{(0)}\right] = \begin{pmatrix} 2 & -1 & 1 & 5 \\ 4 & 1 & 0 & 3 \\ 3 & -7 & -4 & 2 \end{pmatrix}$$

to illustrate our algorithm as shown in Table 1. The related values of Table 1 are corresponding to the positions of links and registers as shown in Figures 1 and 2, where the arrows are omitted. In what follows, we use the notation "PE$(i,j)[a_{\text{in}} = 2, R = 3, \ldots]t = 4$" to indicate that PE$(i,j)$ has its $a_{\text{in}} = 2$, $R = 3$, and so on at the time step $t = 4$. The symbol "S1 $\Longrightarrow$ S2" means that the statement S1 implies the statement S2.

Table 1. An illustrative for $n = 3$ (the first linear array).



There are three linear equations which correspond to the linear system $Ax = b$.

$$2x_1 - x_2 + x_3 = 5, \tag{1}$$

$$4x_1 + x_2 = 3, \tag{2}$$

$$3x_1 - 7x_2 - 4x_3 = 2. \tag{3}$$

In Table 1, three linear systolic arrays are considered in the following three cases, respectively.

CASE (α). On the first linear array, since PE$(1,1)[a_{\text{in}} = 2, P = 3]t = 1$, the equation (1) is a pivot row. From the procedures *finding-first-pivot* and *storing-first-pivot*, we have

$$\text{PE}(1,1)\,[P = -2, R = 2, d_{\text{out}} = 2, c_{\text{out}} = 0]\,t = 1,$$

Table 1. (cont.) The second linear array.

Time	PE(2,2)	PE(2,3)	PE(2,4)
3	* / * 2 * / *	* / * * * / *	* / * * * / *
4	-3/4 / 3/4 -1 -3/4 / 0	* / * * * / *	* / * * * / *
5	-31/4 / 31/4 0 -31/4 / 2	1/2 / -3/4 0 -2/3 -3/4 0	* / * * * / *
6	1/4 / 31/4 0 1/4 / 0	-4 / -31/4 2 16/31 -31/4 2 / -110/93	7/4 / -3/4 0 -7/3 -3/4 0
7	^ / ^ 0 ^ / ^	0 / 1/4 0 16/31 1/4 0 / -4/31	-1/4 / -31/4 2 1/31 -31/4 2 / -220/93
8	stop	^ / ^ ^ & ^ / 16/31	3/4 / 1/4 0 1/31 1/4 0 / 23/31
9		^ / ^ ^ / ^	^ / ^ ^ & ^ / 1/31
10		stop	^ / ^

$$\Longrightarrow \mathrm{PE}(1,2)\left[c_{\mathrm{in}}=0, a_{\mathrm{in}}=-1, d_{\mathrm{in}}=2, R=-\frac{1}{2}, a_{\mathrm{out}}=*, d_{\mathrm{out}}=2, c_{\mathrm{out}}=0\right] t=2,$$

$$\Longrightarrow \mathrm{PE}(1,3)\left[c_{\mathrm{in}}=0, a_{\mathrm{in}}=1, d_{\mathrm{in}}=2, R=\frac{1}{2}, a_{\mathrm{out}}=*, d_{\mathrm{out}}=2, c_{\mathrm{out}}=0\right] t=3,$$

$$\Longrightarrow \mathrm{PE}(1,4)\left[c_{\mathrm{in}}=0, a_{\mathrm{in}}=5, d_{\mathrm{in}}=2, R=\frac{5}{2}, a_{\mathrm{out}}=*, d_{\mathrm{out}}=2, c_{\mathrm{out}}=0\right] t=4.$$

From equation (2), we have $\mathrm{PE}(1,1)[P=-2, a_{\mathrm{in}}=4, R=2]t=2$. By the executions of *trying-new-pivot* and the fact $|a_{\mathrm{in}}|>R$, the message of exchanging two pivot rows must be carried to $\mathrm{PE}(1,j)$ for $j>1$. Thus, by the assignment of $a_{\mathrm{out}}=R-a_{\mathrm{in}}/d_{\mathrm{in}}$ within the procedure *exchanging-two-pivots*, we obtain

$$\mathrm{PE}(1,1)\,[R=4, d_{\mathrm{out}}=4, c_{\mathrm{out}}=2]\,t=2,$$

$$\Longrightarrow \mathrm{PE}(1,2)\left[d_{\mathrm{in}}=4, c_{\mathrm{in}}=2, a_{\mathrm{in}}=1, a_{\mathrm{out}}=-\frac{3}{4}, R=\frac{1}{4}, d_{\mathrm{out}}=4, c_{\mathrm{out}}=2\right] t=3,$$

$$\Longrightarrow \mathrm{PE}(1,3)\left[d_{\mathrm{in}}=4, c_{\mathrm{in}}=2, a_{\mathrm{in}}=0, a_{\mathrm{out}}=\frac{1}{2}, R=0, d_{\mathrm{out}}=4, c_{\mathrm{out}}=2\right] t=4,$$

$$\Longrightarrow \mathrm{PE}(1,4)\left[d_{\mathrm{in}}=4, c_{\mathrm{in}}=2, a_{\mathrm{in}}=3, a_{\mathrm{out}}=\frac{7}{4}, R=\frac{3}{4}, d_{\mathrm{out}}=4, c_{\mathrm{out}}=2\right] t=5.$$

Table 1. (cont.) The third linear array.

Time	PE(3,3)	PE(3,4)
5	* 1	*
6	* 1	*
7	-110/93 110/93 0 -110/93 0	*
8	-4/31 110/93 0 -4/31 0	-110/93 0 -220/93 2 -110/93 0
9	16/31 110/93 0 16/31 0	-4/31 0 23/31 2 1 -4/31 0
10	^ ^ 0 ^	16/31 0 1/31 2 -1 16/31 0
11	stop	^ ^ & 2 ^
12		^ ^

The above three values of a_{out} form the first row of $[A^{(1)}, b^{(1)}]$ with row index 2. For equation (3), by the procedure *trying-new-pivot* with $|a_{\text{in}}| = 3 < R$ and the execution of the assignment $a_{\text{out}} = a_{\text{in}} - R * d_{\text{in}}$ within the procedure *modifying-a-link*, we have

$$\begin{aligned}
&\text{PE}(1,1)\,[P=-1, a_{\text{in}}=3, R=4, d_{\text{out}}=3, c_{\text{out}}=0]\, t=3,\\
\Longrightarrow&\text{PE}(1,2)\left[a_{\text{in}}=-7, d_{\text{in}}=3, c_{\text{in}}=0, R=\frac{1}{4}, a_{\text{out}}=-\frac{31}{4}, d_{\text{out}}=3, c_{\text{out}}=0\right] t=4,\\
\Longrightarrow&\text{PE}(1,3)\,[a_{\text{in}}=-4, d_{\text{in}}=3, c_{\text{in}}=0, R=0, a_{\text{out}}=-4, d_{\text{out}}=3, c_{\text{out}}=0]\, t=5,\\
\Longrightarrow&\text{PE}(1,4)\left[a_{\text{in}}=2, d_{\text{in}}=3, c_{\text{in}}=0, R=\frac{3}{4}, a_{\text{out}}=-\frac{1}{4}, d_{\text{out}}=3, c_{\text{out}}=0\right] t=6.
\end{aligned}$$

The above three values of a_{out} form the second row of $[A^{(1)}, b^{(1)}]$ with row index 3. From the initial state of our algorithm, we have $\text{PE}(1,j)[a_{\text{in}} = \hat{}\,]t = j+3$ for $1 \le j \le 4$. Hence, PE(1,1) stops its execution at $t=4$. From the procedure *sending-last-element*, we have

$$\text{PE}(1,2)\left[a_{\text{in}} = \hat{}, a_{\text{out}}=\frac{1}{4}, R=\&\right] t=5, \qquad \text{PE}(1,3)\,[a_{\text{in}} = \hat{}, a_{\text{out}}=0, R=\&]\, t=6,$$

and

$$\text{PE}(1,4)\left[a_{\text{in}} = \hat{}, a_{\text{out}}=\frac{3}{4}, R=\&\right] t=7.$$

The above three values of a_{out} form the third row of $[A^{(1)}, b^{(1)}]$ with 1 as its row index. Then we have $\text{PE}(1,j)[a_{\text{out}} = \hat{}\,, R = \hat{}\,]t = j + 4$ for $2 \le j \le 4$. Thus, these three PEs stop their executions.

From the above executions, we obtain

$$\left[A^{(1)}, b^{(1)}\right] = \begin{pmatrix} -\frac{3}{4} & \frac{1}{2} & \frac{7}{4} \\ -\frac{31}{4} & -4 & -\frac{1}{4} \\ \frac{1}{4} & 0 & \frac{3}{4} \end{pmatrix},$$

which corresponds to the linear system of

$$\left(-\frac{3}{4}\right) x_2 + \left(\frac{1}{2}\right) x_3 = \frac{7}{4}, \tag{4}$$

$$\left(-\frac{31}{4}\right) x_2 - 4x_3 = -\frac{1}{4}, \tag{5}$$

$$x_1 + \left(\frac{1}{4}\right) x_2 = \frac{3}{4}. \tag{6}$$

CASE (β). On the second linear systolic array, $[A^{(1)}, b^{(1)}]$ is the input data on the input a-links of $\text{PE}(2,j)$ for $2 \le j \le 4$. From equation (4) and $\text{PE}(1,2)[a_{\text{out}} = -3/4]t = 3$, we have

$$\text{PE}(2,2)\left[a_{\text{in}} = -\frac{3}{4}, R = \frac{3}{4}, d_{\text{out}} = -\frac{3}{4}, c_{\text{out}} = 0, P = -1\right] t = 4,$$

$$\Longrightarrow \text{PE}(2,3)\left[c_{\text{in}} = 0, d_{\text{in}} = -\frac{3}{4}, a_{\text{in}} = \frac{1}{2}, R = -\frac{2}{3}, a_{\text{out}} = *, d_{\text{out}} = -\frac{3}{4}, c_{\text{out}} = 0\right] t = 5,$$

$$\Longrightarrow \text{PE}(3,4)\left[c_{\text{in}} = 0, d_{\text{in}} = -\frac{3}{4}, a_{\text{in}} = \frac{7}{4}, R = -\frac{7}{3}, a_{\text{out}} = *, d_{\text{out}} = -\frac{3}{4}, c_{\text{out}} = 0\right] t = 6.$$

From equation (5) and the procedure *exchanging-two-pivots* with $|a_{\text{in}}| > R$, we have

$$\text{PE}(2,2)\left[a_{\text{in}} = -\frac{31}{4}, R = \frac{31}{4}, d_{\text{out}} = -\frac{31}{4}, c_{\text{out}} = 2, P = 0\right] t = 5,$$

$$\Longrightarrow \text{PE}(2,3)\left[c_{\text{in}} = 2, d_{\text{in}} = -\frac{31}{4}, a_{\text{in}} = -4, a_{\text{out}} = -\frac{110}{93}, R = \frac{16}{31}, d_{\text{out}} = -\frac{31}{4}, c_{\text{out}} = 2\right] t = 6,$$

$$\Longrightarrow \text{PE}(2,4)\left[c_{\text{in}} = 2, d_{\text{in}} = -\frac{31}{4}, a_{\text{in}} = -\frac{1}{4}, a_{\text{out}} = -\frac{220}{93}, R = \frac{1}{31}, d_{\text{out}} = -\frac{31}{4}, c_{\text{out}} = 2\right] t = 7.$$

The above two values of a_{out} form the first row of $[A^{(2)}, b^{(2)}]$ with row index 3. From equation (6), we have

$$\text{PE}(2,2)\left[a_{\text{in}} = \frac{1}{4}, R = \frac{31}{4}, d_{\text{out}} = \frac{1}{4}, c_{\text{out}} = 0\right] t = 6,$$

$$\Longrightarrow \text{PE}(2,3)\left[c_{\text{in}} = 0, d_{\text{in}} = \frac{1}{4}, a_{\text{in}} = 0, R = \frac{16}{31}, a_{\text{out}} = -\frac{4}{31}, d_{\text{out}} = \frac{1}{4}, c_{\text{out}} = 0\right] t = 7,$$

$$\Longrightarrow \text{PE}(3,4)\left[c_{\text{in}} = 0, d_{\text{in}} = \frac{1}{4}, a_{\text{in}} = \frac{3}{4}, R = \frac{1}{31}, a_{\text{out}} = \frac{23}{31}, d_{\text{out}} = \frac{1}{4}, c_{\text{out}} = 0\right] t = 8.$$

The above two values of a_{out} form the second row of $[A^{(2)}, b^{(2)}]$ with row index 1. Now $\text{PE}(1,2)[a_{\text{out}} = \hat{}\,]t = 6$ implies $\text{PE}(2,2)[a_{\text{in}} = \hat{}\,]t = 7$. In the same way, we have

$$\text{PE}(1,3)\,[a_{\text{out}} = \hat{}\,]\,t = 7 \quad \text{and} \quad \text{PE}(1,4)\,[a_{\text{out}} = \hat{}\,]\,t = 8,$$

$$\Longrightarrow \text{PE}(2,3)\left[a_{\text{in}} = \hat{}\,, a_{\text{out}} = \frac{16}{31}, R = \&\right] t = 8 \quad \text{and} \quad \text{PE}(2,4)\left[a_{\text{in}} = \hat{}\,, a_{\text{out}} = \frac{1}{31}, R = \&\right] t = 9.$$

These two values of a_{out} form the third row of $[A^{(2)}, b^{(2)}]$ with row index 2. After this time step, PE$(2, j)$ stops its execution.

Until now, we have the matrix

$$\left[A^{(2)}, b^{(2)}\right] = \begin{pmatrix} -\frac{110}{93} & -\frac{220}{93} \\ -\frac{4}{31} & \frac{23}{31} \\ \frac{16}{31} & \frac{1}{31} \end{pmatrix},$$

which corresponds to the linear system of

$$\left(-\frac{110}{93}\right) x_3 = -\frac{220}{93}, \tag{7}$$

$$x_1 - \left(\frac{4}{31}\right) x_3 = \frac{23}{31}, \tag{8}$$

$$x_2 + \left(\frac{16}{31}\right) x_3 = \frac{1}{31}. \tag{9}$$

CASE (γ). We consider the operations on the third linear array. Equation (7) is a pivot row. Hence, it implies

$$\text{PE}(3,3)\left[a_{\text{in}} = -\frac{110}{93}, R = \frac{110}{93}, d_{\text{out}} = -\frac{110}{93}, c_{\text{out}} = 0, P = 0\right] t = 7,$$

$$\Longrightarrow \text{PE}(3,4)\left[c_{\text{in}} = 0, d_{\text{in}} = -\frac{110}{93}, a_{\text{in}} = -\frac{220}{93}, R = 2, a_{\text{out}} = *d_{\text{out}} = -\frac{110}{93}, c_{\text{out}} = 0\right] t = 8.$$

From equation (8), we have

$$\text{PE}(3,3)\left[a_{\text{in}} = -\frac{4}{31}, d_{\text{out}} = -\frac{4}{31}, P = 0, c_{\text{out}} = 0\right] t = 8,$$

$$\Longrightarrow \text{PE}(3,4)\left[c_{\text{in}} = 0, d_{\text{in}} = -\frac{4}{31}, R = 2, a_{\text{in}} = \frac{23}{31}, a_{\text{out}} = 1, d_{\text{out}} = -\frac{4}{31}, c_{\text{out}} = 0\right] t = 9.$$

This output of $a_{\text{out}} = 1$ forms the first row of $[A^{(3)}, b^{(3)}]$ with row index 1. Note that in this last linear array, the matrix $A^{(3)}$ is empty. From equation (9), we have

$$\text{PE}(3,3)\left[a_{\text{in}} = \frac{16}{31}, R = \frac{110}{93}, d_{\text{out}} = \frac{16}{31}, P = 0, c_{\text{out}} = 0\right] t = 9,$$

$$\Longrightarrow \text{PE}(3,4)\left[c_{\text{in}} = 0, d_{\text{in}} = \frac{16}{31}, a_{\text{in}} = \frac{1}{31}, R = 2, a_{\text{out}} = -1, d_{\text{out}} = \frac{16}{31}, c_{\text{out}} = 0\right] t = 10.$$

This $a_{\text{out}} = -1$ is the only entry in the second row of $[A^{(3)}, b^{(3)}]$ with row index 2. Since PE$(2,3)$ has its $a_{\text{out}} = \hat{}$ at $t = 9$, we have PE$(3,3)[a_{\text{in}} = \hat{}, R = \hat{}]t = 10$. At this same time step, PE$(3,3)$ stops its execution. Also we have PE$(2,4)[a_{\text{out}} = \hat{}]t = 10$ which implies the result of PE$(3,4)[a_{\text{in}} = \hat{}, a_{\text{out}} = 2, R = \&]t = 11$.

This $a_{\text{out}} = 2$ is in the third row of $[A^{(3)}, b^{(3)}]$ with row index 3. Finally, we have PE$(3,4)$ has its $R = \hat{}, a_{\text{out}} = \hat{}$ at $t = 12$, and thus, all PEs stop their executions. The solution of $Ax = b$ is in the matrix $[A^{(3)}, b^{(3)}]$ which corresponds to the linear equations of $x_1 = 1$, $x_2 = -1$, and $x_3 = 2$.

7. THE CORRECTNESS PROOF

From the procedure *sending-first-pivot*, we know that when the first pivot row is found by PE(i,i), the type II PE(i,j) sends out a symbol "*" to its a_{out}. These symbols "*" which are generated by the i^{th} linear array will propagate to the PEs of the following k^{th} linear array for $k > i$. These synbols "*" and the waiting symbol "*" given in the initial state of our algorithm will cause some PEs to be idle.

LEMMA 1. *On the i^{th} linear array, when we ignore the "*" due to the initial state, but we include the "*" generated by the previous linear array, the first entry on the input a-link of PE(i,j) appears at the time step $t = i + j - 1$.*

PROOF. From the initial state, for $1 \le j \le n$, we obtain $\text{PE}(1,j)[a_{\text{in}} = a_{1,j}^{(0)}]t = j$. Also we have $\text{PE}(1,n+1)[a_{\text{in}} = b_1^{(0)}]t = n+1$. Since there is one time delay on each a-link, by induction on the integer i, PE(i,j) has an entry on its a-link at the time step $t = j + i - 1$ for $i \le j \le n+1$. ∎

LEMMA 2. *On the i^{th} linear array, if we have the entry $a_{k,i}^{(i-1)} = 0$, for all $i \le k \le n$, then PE(i,i) send out its $c_{\text{out}} = 3$ before the time step $t = n + 2i - 2$.*

PROOF. By the procedure *testing-first-pivot*, we know that the case of all $a_{k,i}^{(i-1)} = 0$ will be detected by PE(i,i) on the i^{th} linear array. From Lemma 1, PE(i,i) has its first entry on its input a-link at $t = 2i - 1$. Since $a_{i,i}^{(i-1)}$ is the first entry of the first column of $A^{(i-1)}$ and there are at most $(i-1)$ symbols "*" generated from the previous $(i-1)$ linear arrays, PE(i,i) has the entry $a_{i,i}^{(i-1)}$ on its input a-link before the time step $t = 2i - 1 + (i-1) = 3i - 2$. Thus, PE$(i,i)$ has its $a_{\text{in}} = a_{n,i}^{(i-1)}$ before $t = (3i-2) + (n-i) = n + 2i - 2$. At this time step, PE$(i,i)$ has $P = 1$ and $a_{\text{in}} = 0$. The procedure *sending-singular-matrix* implies that PE(i,i) sends its $c_{\text{out}} = 3$ to indicate the singularity of A. ∎

LEMMA 3. *During the execution on the i^{th} linear array, if there exists an integer k, for $i \le k \le n$ such that the entries $a_{k,j}^{(i-1)} = 0$, for all $i \le j \le n$, then PE(i,n) has its $c_{\text{out}} = 1$ before the time step $t = n + k + i - 2$.*

PROOF. By Lemma 1, the first entry a_{in} of PE(i,i) appears at $t = 2i-1$. Since the first $(k-i+1)$ rows of $[A^{(i-1)}, b^{(i-1)}]$ are indexed sequentially from i to k, and there are at most $(i-1)$ symbols "*" generated by the previous $(i-1)$ linear arrays, the entry $a_{k,i}^{(i-1)}$ will meet PE(i,i) before the time step $t = 2i-1+(k-i)+(i-1) = 2i+k-2$. By $a_{k,i}^{(i-1)} = 0$, PE(i,i) has its $c_{\text{out}} = 1$ by the execution of procedure *searching-first-pivot* or the procedure *trying-new-pivot*. Thus, we have

$$\begin{aligned}
&\text{PE}(i,i)\,[a_{\text{in}} = 0, c_{\text{out}} = 1]\, t \le 2i + k - 2, \\
\Longrightarrow &\text{PE}(i,i+1)\left[c_{\text{in}} = 1, a_{\text{in}} = 0 = a_{k,i+1}^{(i-1)}, c_{\text{out}} = 1\right] t \le 2i + k - 1, \\
\Longrightarrow &\text{PE}(i,n)\,[c_{\text{in}} = 1, a_{\text{in}} = 0, c_{\text{out}} = 1]\, t \le 2i + k - 2 + (n-i) = n + k + i - 2.
\end{aligned}$$

∎

The $c_{\text{out}} = 1$ on PE(i,n) indicates that $A^{(i-1)}$ has a zero row. Therefore, A is a singular matrix. From Lemmas 2 and 3, we know that if there exists an integer i such that PE(i,n) has its $c_{\text{out}} = 1$ or $c_{\text{out}} = 3$, then A is singular. In the following three lemmas, we assume that A is nonsingular.

LEMMA 4. *The type I PE(i,i) stops its execution at the time step $t = n + 3i - 2$ and the type II PE(i,j) stops its execution at the time step $t = n + 2i + j - 1$. That is, we have PE$(i,i)[R = \hat{}\,]t = n + 3i - 2$ and PE$(i,j)[R = \hat{}\,, a_{\text{out}} = \hat{}\,]t = n + 2i + j - 1$, for $1 \le i \le n$ and $i < j \le n+1$.*

PROOF. By induction on i.

Basis: For $i = 1$. From the initial state, we have

$$\begin{aligned}
&\text{PE}(1,1)\left[a_{\text{in}} = a_{1,1}^{(0)}\right] t = 1, \\
\Longrightarrow &\text{PE}(1,1)\left[a_{\text{in}} = a_{1,n}^{(0)}\right] t = n, \\
\Longrightarrow &\text{PE}(1,1)\,[a_{\text{in}} = \hat{}\,, R = \hat{}\,]\, t = n + 1.
\end{aligned}$$

From the initial state, we know that the entry of the first row of $[A^{(0)}, b^{(0)}]$ meets $\mathrm{PE}(1,j)$ at the time step $t = j + i - 1$. Since each column in $[A^{(0)}, b^{(0)}]$ has n entries, we have

$$\begin{aligned}
&\mathrm{PE}(1,j)\left[a_{\mathrm{in}} = a_{n,j}^{(0)}\right] t = n + j - 1,\\
\Longrightarrow &\mathrm{PE}(1,j)\,[a_{\mathrm{in}} = \hat{}\,, R = \&]\, t = n + j,\\
\Longrightarrow &\mathrm{PE}(1,j)[R = \hat{}\,, a_{\mathrm{out}} = \hat{}\,]t = n + j + 1.
\end{aligned}$$

Thus, it is true for $i = 1$.

Assumption: For $i = k$, it is true. Thus, we have

$$\mathrm{PE}(k,k)\,[R = \hat{}\,]\, t = n + 3k - 2,$$

and

$$\mathrm{PE}(k,j)\,[R = \hat{}\,, a_{\mathrm{out}} = \hat{}\,]\, t = n + 2k + j - 1, \qquad \text{for } j \geq k + 1.$$

Induction: For $i = k + 1$.

From the above assumption with $j = k + 1$, we have

$$\begin{aligned}
&\mathrm{PE}(k,k+1)\,[a_{\mathrm{out}} = \hat{}\,]\, t = n + 2k + (k+1) - 1 = n + 3k,\\
\Longrightarrow &\mathrm{PE}(k+1,k+1)\,[a_{\mathrm{in}} = \hat{}]\, t = n + 3k + 1 = n + 3(k+1) - 2,\\
\Longrightarrow &\mathrm{PE}(k+1,k+1)\,[R = \hat{}\,]\, t = n + 3(k+1) - 2.
\end{aligned}$$

From the above assumption with $j > k + 1$, we have

$$\begin{aligned}
&\mathrm{PE}(k,j)\,[R = \hat{}\,]\, t = n + 2k + j - 1,\\
\Longrightarrow &\mathrm{PE}(k,j)\,[a_{\mathrm{out}} = \hat{}\,]\, t = n + 2k + j - 1,\\
\Longrightarrow &\mathrm{PE}(k+1,j)\,[a_{\mathrm{in}} = \hat{}\,, R = \&]\, t = n + 2k + j,\\
\Longrightarrow &\mathrm{PE}(k+1,j)\,[R = \hat{}\,, a_{\mathrm{out}} = \hat{}\,]\, t = n + 2k + j + 1 = n + 2(k+1) + j - 1.
\end{aligned}$$

Thus, $i = k + 1$ is true. Therefore, the lemma is proved by induction. ∎

LEMMA 5. *The time complexity of our systolic algorithm is* $4n$.

PROOF. Let $i = n$ and $j = n+1$, in Lemma 4. We have $\mathrm{PE}(n, n+1)[R = \hat{}\,]t = n+2i+j-1 = 4n$. Thus, $\mathrm{PE}(n, n+1)$ stops its execution at the time step $t = 4n$. ∎

LEMMA 6. *The* i^{th} *linear array produces the values of* a_{out} *to form the matrix* $[A^{(i)}, b^{(i)}]$ *which corresponds to the linear system of*

$$\begin{aligned}
a_{i+1,i+1}^{(i)} x_{i+1} + a_{i+1,i+2}^{(i)} x_{i+2} + \cdots + a_{i+1,n}^{(i)} x_n &= b_{i+1}^{(i)},\\
a_{i+2,i+1}^{(i)} x_{i+1} + a_{i+2,i+2}^{(i)} x_{i+2} + \cdots + a_{i+2,n}^{(i)} x_n &= b_{i+2}^{(i)},\\
&\cdots\cdots\cdots\cdots\\
a_{n,i+1}^{(i)} x_{i+1} + a_{n,i+2}^{(i)} x_{i+2} + \cdots + a_{n,n}^{(i)} x_n &= b_n^{(i)},\\
x_1 + a_{1,i+1}^{(i)} x_{i+1} + a_{1,i+2}^{(i)} x_{i+2} + \cdots + a_{1,n}^{(i)} x_n &= b_1^{(i)},\\
&\cdots\cdots\cdots\cdots\\
x_i + a_{i,i+1}^{(i)} x_{i+1} + a_{i,i+2}^{(i)} x_{i+2} + \cdots + a_{i,n}^{(i)} x_n &= b_i^{(i)}.
\end{aligned}$$

PROOF. By the mathematical induction on i.

Basis: For $i = 1$.

From the initial state, the matrix $[A, b] = [A^{(0)}, b^{(0)}]$ is assigned to the input a-links of $\mathrm{PE}(1, j)$, $1 \leq j \leq (n+1)$. Thus, we have $\mathrm{PE}(1,1)[a_{\mathrm{in}} = a_{k,1}^{(0)}]t = k$ for $1 \leq k \leq n$..

Suppose that $a_{u,1}^{(0)}$ is the first nonzero value meeting $\mathrm{PE}(1,1)$ at the time step $t = u$. Since $\mathrm{PE}(1,1)[P = n]t = 1$, the procedure *testing-first-pivot* causes the *searching-first-pivot* to be executed for $(u-1)$ times. Thus, we have $\mathrm{PE}(1,1)[P = n-u+1]t = u-1$. At the next time step, we have $\mathrm{PE}(1,1)[a_{\mathrm{in}} \neq 0]t = u$. From the execution of *finding-first-pivot*, we have $\mathrm{PE}(1,1)[P = -n+u]t = u$.

From the time step $t = 1$ to the time step $t = u-1$, $\mathrm{PE}(1,1)$ sets its $c_{\mathrm{out}} = 1$ and $d_{\mathrm{out}} = 0$. This causes the type II $\mathrm{PE}(1, j)$ assigning its a_{in} to its a_{out} by the assignment of $a_{\mathrm{out}} = a_{\mathrm{in}}$ within the procedure *storing-first-pivot* under the condition of $d_{\mathrm{in}} = 0$. That is, for $2 \leq j \leq (n+1)$, $\mathrm{PE}(1, j)$ carries the first $(u-1)$ rows of $[A^{(0)}, b^{(0)}]$, under deleting the first column of $A^{(0)}$, to form the first $(u-1)$ rows of $[A^{(1)}, b^{(1)}]$. These $(u-1)$ rows in $[A^{(1)}, b^{(1)}]$ are numbered as the row indexes from 2 to u.

Since the u^{th} row of $[A^{(0)}, b^{(0)}]$ is the first pivot row found by $\mathrm{PE}(1,1)$, we obtain $d_{\mathrm{in}} \neq 0$ on $\mathrm{PE}(1, j)$ for $2 \leq j \leq (n+1)$. From the procedure *storing-first-pivot*, we have

$$\mathrm{PE}(1, j)\left[R = \frac{a_{\mathrm{in}}}{d_{\mathrm{in}}}, a_{\mathrm{out}} = *\right] t = u + j - 1, \qquad \text{for } 2 \leq j \leq (n+1).$$

From the time step $t = u+1$ to $t = n$, $\mathrm{PE}(1,1)$ sends the message to indicate whether the action of exchanging pivot row occurs or not. This implies that the type II $\mathrm{PE}(1, j)$ modifies the last $(n-u)$ rows of $[A^{(0)}, b^{(0)}]$ to form the rows of $[A^{(1)}, b^{(1)}]$ with the row indexes from $u+1$ to n. Until now, we obtain $(n-1)$ rows in $[A^{(1)}, b^{(1)}]$. These $(n-1)$ rows of $[A^{(1)}, b^{(1)}]$ correspond to $(n-1)$ linear equations such that all the coefficients of x_1 are deleted.

At the time step $t = n+1$, PE(1,1) receives the stooping signal $a_{\mathrm{in}} = \hat{}$. So PE(1,1) stops its execution. Similarly, we have $\mathrm{PE}(1, j)[a_{\mathrm{in}} = \hat{}\,]t = n + j$, for $2 \leq j \leq (n+1)$. The procedure *sending-last-element* causes $\mathrm{PE}(1, j)$ to assign the content of R to a_{out} to form the last row of $[A^{(1)}, b^{(1)}]$. This last row has its index 1 and it corresponds to a linear equation such that its coefficient of the unknown x_1 is 1. Therefore, the case of $i = 1$ is true.

Assumption: Suppose that this lemma is true for $i = k$.

Induction: For $i = k+1$.

Since the values on the output a-links of the k^{th} linear array form the matrix $[A^{(k)}, b^{(k)}]$, this matrix is the input of a-links of the $(k+1)^{\mathrm{th}}$ linear array. By Lemma 1, we know that the first entry generated by the k^{th} linear array meets $\mathrm{PE}(k+1, k+1)$ at $t = 2k+1$.

For $k+1 \leq r \leq n$, let $a_{r,k+1}^{(k)}$ meet $\mathrm{PE}(k+1, k+1)$ at the time step $t = t_r$. Since there are k symbols "*" generated by the previous k linear arrays, the t_r has to satisfy the condition of $2k+1 \leq t_r \leq n+2k$. Suppose that $a_{v,k+1}^{(k)}$ is the first nonzero value meeting $\mathrm{PE}(k+1, k+1)$ at the time step $t = t_v$, for $k+1 \leq v \leq n$. That is, we have $a_{s,k+1}^{(k)} = 0$, for s satisfing $k+1 \leq s \leq v-1$. Hence, from the time step $t = t_s + j - k - 1$ to $t = t_s + j + v - 2k - 2$, the type II $\mathrm{PE}(k+1, j)$ carries the first $(v-k-1)$ rows of $[A^{(k)}, b^{(k)}]$, under deleting the first column in $A^{(k)}$, to form the first $(v-k-1)$ rows of $[A^{(k+1)}, b^{(k+1)}]$ with the row indexes from $k+2$ to v.

Since the first pivot row of $[A^{(k)}, b^{(k)}]$ is found by $\mathrm{PE}(k+1, k+1)$ at $t = t_v$ by $a_{v,k+1}^{(k)} \neq 0$, we have $\mathrm{PE}(k+1, j)[R = a_{\mathrm{in}}/d_{\mathrm{in}}, a_{\mathrm{out}} = *]t = t_v + j - k - 1$, for $k+2 \leq j \leq n+1$. After sending $a_{\mathrm{out}} = *$, $\mathrm{PE}(k+1, j)$ modifies the following $(n-v+k)$ rows of $[A^{(k)}, b^{(k)}]$ to form the $(n-v+k)$ rows of $[A^{(k+1)}, b^{(k+1)}]$ with row indexes as $v+1, v+2, \ldots, n-1, n, 1, 2, \ldots, k-2, k-1, k$.

At the initial state, we set $\mathrm{PE}(i, i)[P = n-i+1]t = 0$. Thus, on the $(k+1)^{\mathrm{th}}$ linear array, there are only the first $(n-k)$ rows of $[A^{(k)}, b^{(k)}]$ to be detected as a pivot row. The last k rows of $[A^{(k)}, b^{(k)}]$, with row index l for $1 \leq l \leq k$, correspond to the k linear equations with the coefficient 1 on the unknown variables x_l, respectively. On the $(k+1)^{\mathrm{th}}$ linear array, these k

equations are not involved into the work for detecting a pivot row. Since the pivot row on the $(k+1)^{\text{th}}$ linear array does not contain the unknowns x_l for $1 \leq l \leq k$, the assignment of $a_{\text{out}} = a_{\text{in}} - R * d_{\text{in}}$ cannot influence the coefficients of x_l. Thus, the coefficient of x_l preserves as 1 under the executions of the $(k+1)^{\text{th}}$ linear array. From Lemma 4, for $k+2 \leq j \leq n+1$, we have

$$\begin{aligned} &\text{PE}(k, k+1)\,[a_{\text{out}} = \hat{}\,]\,t = n + 3k \quad \text{and} \quad \text{PE}(k, j)\,[a_{\text{out}} = \hat{}\,]\,t = n + 2k + j - 1, \\ &\Longrightarrow \text{PE}(k+1, k+1)\,[a_{\text{in}} = \hat{}\,, R = \hat{}\,]\,t = n + 3k + 1, \end{aligned}$$

and

$$\text{PE}(k+1, j)\,[a_{\text{in}} = \hat{}\,, a_{\text{out}} = R, R = \&] = n + 2k + j.$$

These $(n-k)$ values of a_{out} form the last row of $[A^{(k+1)}, b^{(k+1)}]$ with row index $k+1$. This last row corresponds to a linear equation with the coefficient 1 on the unknown x_{k+1}. At the next time step, we have $\text{PE}(k+1, j)[R = \hat{}\,, a_{\text{out}} = \hat{}\,]t = n + 2k + j + 1$. The case of $i = k+1$ is true. Therefore, this lemma is proved by induction. ∎

THEOREM. *The systolic algorithm LINEAR-SOLVER (A, b, n) is correct to solve the dense linear system $Ax = b$ within $4n$ time steps.*

PROOF. From the results of Lemmas 5 and 6 the following concludes. ∎

8. CONCLUSIONS

We present a systolic algorithm to solve the linear systems $Ax = b$. An important feature in our algorithm is that, during the elementary row operation on A, the action of exchanging two pivot rows can performed. Since we need to preserve the coefficient of x_l, for $1 \leq l \leq (i-1)$, within a linear equation during the execution of the i^{th} linear systolic array, the elementary row operation on the i^{th} linear array has to be accomplished by the way of $R_j - c * R_i$, where R_j is the row to be modified, c is a scale value, and R_i is the current pivot row. This requirement is achieved by the assignment of $a_{\text{out}} = a_{\text{in}} - R * d_{\text{in}}$ within the procedure *modifying-a-link*.

This algorithm can be used to solve the linear systems $AX = B$ for X, B being $n \times m$ matrices. In this case, the algorithm requires $(4n + m - 1)$ time steps. When B is an n by n identity matrix and A is nonsingular, the solution is the inverse matrix of A. Moreover, it seems that the absolute value of the determinant of A is the product of R in $\text{PE}(i, i)$ for $1 \leq i \leq n$.

During the execution of our algorithm, if we have $a_{i,i}^{(i-1)} \neq 0$, for all $1 \leq i \leq n$ and the action of exchanging pivot row does not occur, then our algorithm has the same work as the LU decomposition to do. In this case, the c-link is redundant. In fact, the major purpose of c-link is used to carry the message of exchanging pivot row. However, note that the LU decomposition only obtains the upper triangular matrix of A, but our algorithm solves the linear system $Ax = b$.

We hope that this method of designing a systolic algorithm can be applied to solve some NP-complete problems, such as the knapsack and travel-salesman problems.

REFERENCES

1. C.J. Lin, Generating subsets on a systolic array, *Computers Math. Applic.* **21** (1/2), 103–109, (1991).
2. C.J. Lin, A systolic algorithm for dynamic programming, *Computers Math. Applic.* **27** (1), 1–10, (1994).
3. J.A. Mchugh, *Algorithmic Graph Theory*, Prentice-Hall International, (1990).
4. M. Zubair, Efficient systolic algorithm for find bridges in a connected graph, *Parallel Computing* **6**, 57–61, (1988).
5. S.H. Zak and K. Hwang, Polynomial division on systolic arrays, *IEEE Trans. on Computers* **34** (6), 577–578, (1985).
6. L. Convay and C. Mead, *Introduction to VLSI System*, Addison-Wesley, Reading, MA, (1980).
7. D.J. Evans, Systolic algorithms, *Intern. J. Computer Math.* **25**, 155–172, (1988).
8. H.T. Kung, Why systolic architecture?, *IEEE Computer* **15**, 37–46, (1982).

9. E.A. Ahmed, Solution of dense linear systems on an optimal systolic architecture, *Computer & Elect. Engng.* **13** (3/4), 177–193, (1987).
10. J.R. Gilbert, Parallel symbolic factorization of sparse linear systems, *Parallel Computing* **14**, 151–162, (1990).
11. C.K. Koc and R.M. Piedra, A parallel algorithm for exact solution of linear equations, In *International Conference on Parallel Processing III*, pp. 1–8, (1991).
12. R. Melham, Parallel Gauss-Jordan elimination for solution of dense linear systems, *Parallel Computing* **4**, 339–343, (1987).
13. P.I. Piskoulijski, Error analysis of parallel algorithm for the solution of a tridiagonal toeplitz linear system of equations, *Parallel Computing* **18**, 431–438, (1992).
14. M.K. Sridhar, R. Srinath and K. Parthasarsthy, On the direct parallel solution of system of linear equations: New algorithms and systolic structures, *Information Sciences* **43**, 25–53, (1987).
15. O. Wing and W. Huang, A computation model of parallel solution of linear equations, *IEEE Trans. on Comput.* **C-29**, 632–638, (1980).