Brief Contributions

An Approach to Designing Modular Extensible Linear Arrays for Regular Algorithms

Pen-Yuang Chang and Jong-Chuang Tsay

Abstract—The purpose of this paper is to describe a new method to design unidirectional modular extensible linear arrays for regular algorithms. The time complexity of our method is polynomial and depends only on the number of dimensions of the regular algorithm. The designed linear array is asymptotically optimal in space and time.

Index Terms—Algorithm transformation, conflict-free mapping, data dependency, linear array, modular extensible, optimal spacetime mapping, regular algorithm, systolic array, unimodular matrix, VLSI.

1 Introduction

LINEAR array is one of the most popular and important regular arrays [1]. The attractive features of the linear array include bounded I/O, fault tolerance, minimal communication pattern, and modular extensibility. Previous research done in designing linear arrays for the system of uniform recurrence equations (SURE) fall into two categories. The first one [2], [3], [4], [5] designed linear arrays for different problems case by case. The second category [6], [7], [8], [9], [10] proposed procedures to design systematically linear arrays for SUREs. On the design of *modular extensible* linear arrays, four types of conflict-free conditions should be satisfied. They are dependence, computation, link, and memory conflict-free.

The major problem in designing modular extensible linear arrays is *to check the link conflicts, either the whole computation domain is necessary to be examined or a diophantine equation needs to be solved and enumerated.* To avoid the time-consuming procedure, we propose a transformation matrix in a fixed form that is conflict-free. The time complexity of our method is polynomial and depends only on the number of dimensions of the regular algorithm.

Here is an outline of following sections: Definitions and design concepts are given in Section 2. In Section 3, we propose a conflict-free transformation matrix and prove that the linear arrays designed by our method are asymptotically optimal in space and time. Finally, our concluding remarks are presented in Section 4.

2 Definition and Concept

An SURE [11] is a four-tuple $\mathcal{A}_M = (J^n, V, F, D)$, where J^n is the computation domain, V the set of variables, F the set of functions, and D the set of uniform dependence vectors. For simplicity, we consider only the case of $J^n = \{[j_1, j_2, \cdots, j_n]^T \mid 1 \le j_i \le N, 1 \le i \le n\}$, with $N >> n$. D is represented as a matrix form in which each column $\vec{d}$ corresponds to a dependence vector and $rank(D) = n$ should be satisfied. Without loss of generality, we assume every dependence vector of an SURE is lexicographically positive. Wolf and Lam in [12] showed that an SURE with lexicographically positive dependence vectors can be made fully permutable by skewing. That is, $X_{n\times n}\, D_{n\times m} = F_{n\times m}$, where X is the skew transformation matrix and $\forall f_{ij} \in F, f_{ij} \ge 0$. Thus, we have $D = X^{-1}F \equiv BF$. Since every element in F is greater than or equal to zero, B forms a positive integral coordinate basis of the dependence matrix D. That is, every dependence vector $\vec{d}_j \in D$ can be constructed by positive integral combination of the column vectors $\vec{b}_i$s of B, or $\vec{d}_j = \sum_{i=1}^{n} f_{ij}\vec{b}_i$. The matrix $B_{n\times n}$ can be used to replace $D_{n\times m}$ to form a new dependence matrix of the SURE. Therefore, we consider only that D is an $n \times n$ and full rank matrix in the following.

An SURE $\mathcal{A}_M = (J^n, V, F, D)$ is said to be *fully connected* (FC-SURE) if any two index vectors in J^n are connected [13]. An important property of the FC-SURE is that *the absolute value of the determinant of its dependence matrix is one.* An SURE is said to be *matrix-multiplication-like* (MM-SURE) [14] if its dependence matrix is an identity matrix. The first step of our method is to transform an FC-SURE to an equivalent MM-SURE by a unimodular matrix.

THEOREM 1. *By the unimodular matrix $T_u = [u_{ij}]_{n\times n} = D^{-1}$, an FC-SURE $\mathcal{A}_M = (J^n, V, F, D)$, where $J^n = \{[j_1, j_2, \cdots, j_n]^T \mid 1 \le j_i \le N, 1 \le i \le n\}$, can be transformed to an equivalent MM-SURE $\mathcal{A}_I = (\bar{J}^n, V, F, I)$, where $\bar{J}^n = \{[\acute{j}_1, \acute{j}_2, \ldots, \acute{j}_n]^T \mid \acute{j}_i = \sum_{k=1}^{n} u_{ik} j_k\}$ and I denotes an identity matrix.*

PROOF. See [15].

A *linear array* is a four-tuple $\mathcal{A}_Y = (P, M, C, L)$, where P is a set of PEs that are logically arranged in a sequence, M a set of memory storage in each PE, C a set of computational units in each PE, and L a set of links in each PE in which each link connects two neighboring PEs.

THEOREM 2. *For a transformation matrix $T = [\vec{\lambda}\ \vec{s}]^T$, where $\vec{\lambda}$ and $\vec{s}$ are the time and space mapping, respectively, to be conflict-free on mapping an FC-SURE $\mathcal{A}_M = (J^n, V, F, D)$ to a linear array $\mathcal{A}_Y = (P, M, C, L)$, the necessary and sufficient conditions are as follows: Dependence conflict-free: iff $\forall \vec{d}_i \in D$, $\vec{\lambda}^T\vec{d}_i > 0$. Computation conflict-free: iff $\forall \vec{j}_1, \vec{j}_2 \in J^n$, if $\vec{s}^T\vec{j}_1 = \vec{s}^T\vec{j}_2$, then $\vec{\lambda}^T\vec{j}_1 \ne \vec{\lambda}^T\vec{j}_2$. Link conflict-free: iff $\forall \vec{j}_1, \vec{j}_2 \in J^n$, for each $\vec{d}_j \in D$, if $\vec{\lambda}^T(\vec{j}_1 - \vec{j}_2)\vec{s}^T\vec{d}_j = \vec{s}^T(\vec{j}_1 - \vec{j}_2)\vec{\lambda}^T\vec{d}_j$ then $\vec{j}_1 - \vec{j}_2 = k\vec{d}_j$ or $\vec{s}^T\vec{d}_j = 0$. Memory conflict-free: iff $\forall \vec{j}_1, \vec{j}_2 \in J^n$, for each $\vec{d}_j \in D$, if $\vec{s}^T\vec{j}_1 = \vec{s}^T\vec{j}_2$ and $\vec{\lambda}^T\vec{j}_1 < \vec{\lambda}^T\vec{j}_2$, then $\vec{\lambda}^T\vec{j}_1 + \left|\frac{\vec{\lambda}^T\vec{d}_j}{\vec{s}^T\vec{d}_j}\right| \le \vec{\lambda}^T\vec{j}_2$ when $\vec{s}^T\vec{d}_j \ne 0$, and $\vec{\lambda}^T\vec{j}_1 + \vec{\lambda}^T\vec{d}_j \le \vec{\lambda}^T\vec{j}_2$ when $\vec{s}^T\vec{d}_j = 0$.*

PROOF. The necessary and sufficient conditions of the first three conflict-free have been proven by Lee and Kedem in [7]. Now, we want to prove the necessary and sufficient conditions of memory conflict-free. [If part] From $\vec{s}^T\vec{j}_1 = \vec{s}^T\vec{j}_2$, $\vec{j}_1$ and $\vec{j}_2$ are mapped to the same PE. Case 1 ($\vec{s}^T\vec{d}_j \ne 0$): (Fig. 1a) Since $\vec{s}^T\vec{d}_j \ne 0$, the computation result of an instance of v_j should be propagated to its neighboring PE. Let $t_1 = \vec{\lambda}^T\vec{j}_1$,

• *The authors are with the Institute of Computer Science and Information Engineering, College of Engineering, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China. E-mail: jctsay@ccsie.nctu.edu.tw.*

Manuscript received 21 Dec. 1993; revised 28 Oct. 1997.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 106032.

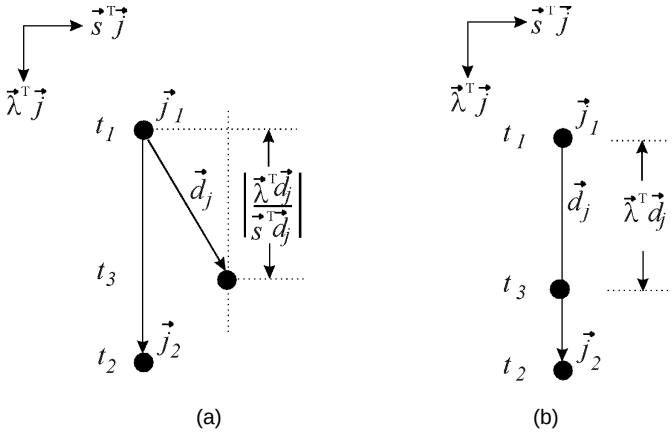

Fig. 1.(a) Case 1 ($s^T d_j \neq 0$) of memory conflict-free mapping, (b) Case 2 ($s^T d_j = 0$) of memory conflict-free mapping.

$t_2 = \vec{\lambda}^T \vec{j}_2$, and $t_3 = \vec{\lambda}^T \vec{j}_1 + \left| \frac{\vec{\lambda}^T \vec{d}_j}{\vec{s}^T \vec{d}_j} \right|$. Since $t_3 \le t_2$, the computation result of $v_j(\vec{j}_1)$ has been propagated out of its PE before that of $v_j(\vec{j}_2)$ is generated. Thus, at every time instance, for each PE, only one memory storage is necessary to store the computation result of an instance of v_j. Case 2 ($\vec{s}^T \vec{d}_j = 0$): (Fig. 1b) It is similar to Case 1. Therefore, if $\vec{s}^T \vec{j}_1 = \vec{s}^T \vec{j}_2$ and $\vec{\lambda}^T \vec{j}_1 < \vec{\lambda}^T \vec{j}_2$, then $\vec{\lambda}^T \vec{j}_1 + \left| \frac{\vec{\lambda}^T \vec{d}_j}{\vec{s}^T \vec{d}_j} \right| \le \vec{\lambda}^T \vec{j}_2$ when $\vec{s}^T \vec{d}_j \neq 0$, and $\vec{\lambda}^T \vec{j}_1 + \vec{\lambda}^T \vec{d}_j \le \vec{\lambda}^T \vec{j}_2$ when $\vec{s}^T \vec{d}_j = 0$. Hence, the condition of memory conflict-free is satisfied. [Only if part] Case 1 ($\vec{s}^T \vec{d}_j = 0$): By contradiction, we assume $\vec{\lambda}^T \vec{j}_1 + \left| \frac{\vec{\lambda}^T \vec{d}_j}{\vec{s}^T \vec{d}_j} \right| > \vec{\lambda}^T \vec{j}_2$. Let $t_1 = \vec{\lambda}^T \vec{j}_2$, $t_2 = \vec{\lambda}^T \vec{j}_2$, and $t_3 = \vec{\lambda}^T \vec{j}_1 + \left| \frac{\vec{\lambda}^T \vec{d}_j}{\vec{s}^T \vec{d}_j} \right|$. Thus, memory conflict occurs during the time interval $[t_2, t_3)$, since another extra memory storage is necessary to store the computation result of $v_j(\vec{j}_2)$ before propagating $v_j(\vec{j}_1)$ to neighboring PE on t_3. Case 2 ($\vec{s}^T \vec{d}_j = 0$): It is similar to Case 1. Therefore, if the condition of memory conflict-free is satisfied, then $\forall \vec{j}_1, \vec{j}_2 \in J^n$, for each $\vec{d}_j \in D$, if $\vec{s}^T \vec{j}_1 = \vec{s}^T \vec{j}_2$ and $\vec{\lambda}^T \vec{j}_1 < \vec{\lambda}^T \vec{j}_2$, then $\vec{\lambda}^T \vec{j}_1 + \left| \frac{\vec{\lambda}^T \vec{d}_j}{\vec{s}^T \vec{d}_j} \right| \le \vec{\lambda}^T \vec{j}_2$ when $\vec{s}^T \vec{d}_j \neq 0$, and $\vec{\lambda}^T \vec{j}_1 + \vec{\lambda}^T \vec{d}_j \le \vec{\lambda}^T \vec{j}_2$ when $\vec{s}^T \vec{d}_j = 0$. □

Our method for designing linear arrays is to find a unimodular matrix T_u to transform the given FC-SURE to an MM-SURE, and to propose a conflict-free transformation matrix in a fixed form, say T_l (Section 3), for mapping the MM-SURE to a linear array.

THEOREM 3. *Designing a modular extensible linear array $\mathcal{A}_Y$ for an FC-SURE $\mathcal{A}_M$ can be achieved by selecting the transformation matrix $T = T_l T_u$, where T_u is the $n \times n$ unimodular matrix for transforming $\mathcal{A}_M$ to an MM-SURE $\mathcal{A}_I$, and $T_l = [\vec{\phi}\ \vec{r}]^T$ is a $2 \times n$ conflict-free transformation matrix for mapping $\mathcal{A}_I$ to $\mathcal{A}_Y$.*

PROOF. From Theorem 1, we know that an FC-SURE $\mathcal{A}_M = (J^n, V, F, D)$ can always be transformed to an equivalent MM-SURE $\mathcal{A}_I = (\bar{J}^n, V, F, I)$ by the unimodular matrix $T_u = D^{-1}$. Now, we want to prove that if T_l is conflict-free for mapping $\mathcal{A}_I$ to $\mathcal{A}_Y$, then $T = T_l T_u \equiv [\vec{\lambda}\ \vec{s}]^T$ is also conflict-free for mapping $\mathcal{A}_M$ to $\mathcal{A}_Y$. Dependence conflict-free: Since T_l is conflict-free, $\forall \vec{e}_i \in I$, $\vec{\phi}^T \vec{e}_i > 0$. Thus, from $\vec{e}_i = T_u \vec{d}_i$, we have $\vec{\phi}^T T_u \vec{d}_i > 0$ or $\vec{\lambda}^T \vec{d}_i > 0$. Computation conflict-free: Since T_l is conflict-free, $\forall \vec{j}_1, \vec{j}_2 \in \bar{J}^n$, if $\vec{r}^T \vec{j}_1 = \vec{r}^T \vec{j}_2$, then $\vec{\phi}^T \vec{j}_1 \neq \vec{\phi}^T \vec{j}_2$. Thus, from $\vec{j}_1 = T_u \vec{j}_1$ and $\vec{j}_2 = T_u \vec{j}_2$, we have if $\vec{r}^T T_u \vec{j}_1 = \vec{r}^T T_u \vec{j}_2$, then $\vec{\phi}^T T_u \vec{j}_1 \neq \vec{\phi}^T T_u \vec{j}_2$, or, if $\vec{s}^T \vec{j}_1 = \vec{s}^T \vec{j}_2$, then $\vec{\lambda}^T j_1 \neq \vec{\lambda}^T \vec{j}_2$. Link conflict-free: Since T_l is conflict-free, iff $\forall \vec{j}_1 \vec{j}_2 \in \bar{J}^n$, for each $\vec{e}_j \in I$, if $\vec{\phi}^T (\vec{j}_1 - \vec{j}_2) \vec{r}^T \vec{e}_j = \vec{r}^T (\vec{j}_1 - \vec{j}_2) \vec{\phi}^T \vec{e}_j$, then $\vec{j}_1 - \vec{j}_2 = k\vec{e}_j$ or $\vec{r}^T \vec{e}_j = 0$. Thus, if $\vec{\phi}^T T_u (\vec{j}_1 - \vec{j}_2) \vec{r}^T T_u \vec{d}_j = \vec{r}^T T_u (\vec{j}_1 - \vec{j}_2) \vec{\phi}^T T_u \vec{d}_j$, then $\vec{j}_1 - \vec{j}_2 = k\vec{d}_j$ or $\vec{r}^T T_u \vec{d}_j = 0$, or, if $\vec{\lambda}^T (\vec{j}_1 - \vec{j}_2) \vec{s}^T \vec{d}_j = \vec{s}^T (\vec{j}_1 - \vec{j}_2) \vec{\lambda}^T \vec{d}_j$, then $\vec{j}_1 - \vec{j}_2 = k\vec{d}_j$ or $\vec{s}^T \vec{d}_j = 0$. Memory conflict-free: Since T_l is conflict-free, $\forall \vec{j}_1, \vec{j}_2 \in \bar{J}^n$, for each $\vec{e}_j \in I$, if $\vec{r}^T \vec{j}_1 = \vec{r}^T \vec{j}_2$ and $\vec{\phi}^T \vec{j}_1 < \vec{\phi}^T \vec{j}_2$, then $\vec{\phi}^T \vec{j}_1 + \left| \frac{\vec{\phi}^T \vec{e}_j}{\vec{r}^T \vec{e}_j} \right| \le \vec{\phi}^T \vec{j}_2$ when $\vec{r}^T \vec{e}_j \neq 0$ and $\vec{\phi}^T \vec{j}_1 + \vec{\phi}^T \vec{e}_j \le \vec{\phi}^T \vec{j}_2$ when $\vec{r}^T \vec{e}_j = 0$. Thus, we have, if $\vec{r}^T T_u \vec{j}_1 = \vec{r}^T T_u \vec{j}_2$ and $\vec{\phi}^T T_u \vec{j}_1 < \vec{\phi}^T T_u \vec{j}_2$, then $\vec{\phi}^T T_u \vec{j}_1 + \left| \frac{\vec{\phi}^T T_u \vec{d}_j}{\vec{r}^T T_u \vec{d}_j} \right| \le \vec{\phi}^T T_u \vec{j}_2$ when $\vec{r}^T T_u \vec{d}_j \neq 0$ and $\vec{\phi}^T T_u \vec{j}_1 + \vec{\phi}^T T_u \vec{d}_j \le \vec{\phi}^T T_u \vec{j}_2$ when $\vec{r}^T T_u \vec{d}_j = 0$. That is, if $\vec{s}^T \vec{j}_1 = \vec{s}^T \vec{j}_2$ and $\vec{\lambda}^T \vec{j}_1 < \vec{\lambda}^T \vec{j}_2$, then $\vec{\lambda}^T \vec{j}_1 + \left| \frac{\vec{\lambda}^T \vec{d}_j}{\vec{s}^T \vec{d}_j} \right| \le \vec{\lambda}^T \vec{j}_2$ when $\vec{\lambda}^T \vec{d}_j \neq 0$ and $\vec{\lambda}^T \vec{j}_1 + \vec{\lambda}^T \vec{d}_j \le \vec{\lambda}^T \vec{j}_2$ when $\vec{\lambda}^T \vec{d}_j = 0$. □

3 DESIGN OF LINEAR ARRAYS

Two topics are discussed in this section: The first one is that we will give a transformation matrix in a fixed form to design a unidirectional modular extensible linear array for an MM-SURE. The correctness of this transformation matrix is also to be proved. The second topic is that we will prove the designed linear array is asymptotically optimal in space and time.

THEOREM 4. *Designing a unidirectional modular extensible linear array $\mathcal{A}_Y$ for an FC-SURE $\mathcal{A}_M$ can be achieved by selecting the transformation matrix $T = T_l T_u$, where $T_u = [u_{ij}]$ is the $n \times n$ unimodular matrix for transforming $\mathcal{A}_M$ to an MM-SURE $\mathcal{A}_I$, and*

$$T_l = \begin{bmatrix} n-1 & (n-2)H & (n-3)H^2 & \cdots & H^{n-2} & \sum_{k=0}^{n-2} H^k \\ 1 & H & H^2 & \cdots & H^{n-2} & 0 \end{bmatrix}$$

is the $2 \times n$ transformation matrix for mapping $\mathcal{A}_I$ to $\mathcal{A}_Y$, where $H = \alpha \max_{1 \le i \le n} \sum_{k=1}^{n} |u_{ik}| N$ ($N >> n$), and, if $n \le 3$, then $\alpha = 1$, else $\alpha = 2$.

PROOF. We want to prove that

$$T_l = \begin{bmatrix} \vec{\phi}^T \\ \vec{r}^T \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_n \\ r_1 & r_2 & \cdots & r_n \end{bmatrix}$$

is conflict-free for mapping $\mathcal{A}_I = (\bar{J}^n, V, F, I)$ to $\mathcal{A}_Y = (P, M, C, L)$. $T_l = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ is a conflict-free mapping when $n = 2$. Now,

we consider the case of $n \geq 3$. Let the index difference between $\vec{j}_1$ and $\vec{j}_2$ be denoted by $\Delta\vec{j} = (\vec{j}_1 - \vec{j}_2)$, and the difference between the ith indexes of $\vec{j}_1$ and $\vec{j}_2$ be denoted by Δj_i. From Theorem 1, we have $\bar{J}^n = \{(j_1, j_2, \cdots, j_n) \mid j_i = \sum_{k=1}^{n} u_{ik} j_k\}$. Thus, $|\Delta j_i| < \sum_{k=1}^{n} |u_{ik}| N$. Since $H = \alpha \max_{1 \leq i \leq n} \sum_{k=1}^{n} |u_{ik}| N$, we have $|\Delta j_i| < \frac{H}{\alpha}, \forall 1 \leq i \leq n$.

- Dependence conflict-free: $\forall \vec{e}_i \in I$, $\vec{\phi}^{\mathrm{T}} \vec{e}_i > 0$, since $\phi_i > 0$.
- Computation conflict-free: Since $|\Delta j_i| < H$, we have $\vec{r}^{\mathrm{T}} \vec{j}_1 = \vec{r}^{\mathrm{T}} \vec{j}_2$, iff $\Delta j_i = 0\ \forall 1 \leq i \leq n-1$ and $\Delta j_n \neq 0$; but, when $\Delta j_i = 0\ \forall 1 \leq i \leq n-1$ and $\Delta j_n \neq 0$, we have $\vec{\phi}^{\mathrm{T}} \Delta\vec{j} \geq \sum_{k=0}^{n-2} H^k \neq 0$. Thus, $\forall \vec{j}_1, \vec{j}_2 \in \bar{J}^n$, if $\vec{r}^{\mathrm{T}} \vec{j}_1 = \vec{r}^{\mathrm{T}} \vec{j}_2$, then $\vec{\phi}^{\mathrm{T}} \vec{j}_1 \neq \vec{\phi}^{\mathrm{T}} \vec{j}_2$.
- Link conflict-free: For the case of e_n, no link would be generated, since $s_n = 0$. For each $e_{i \neq n} \in I$, from $\vec{\phi}^{\mathrm{T}}(\vec{j}_1 - \vec{j}_2)\vec{r}^{\mathrm{T}}\vec{e}_i = \vec{r}^{\mathrm{T}}(\vec{j}_1 - \vec{j}_2)\vec{\phi}^{\mathrm{T}}\vec{e}_i$, we have
$$\sum_{k=1}^{i-1}(k-i)H^{k-1}\Delta j_k + \sum_{k=i+1}^{n-1}(k-i)H^{k-1}\Delta j_k - \sum_{k=1}^{n-1} H^{k-1}\Delta j_n = 0.$$

Its left-hand side is a polynomial in H with $|\Delta j_k| < \frac{H}{\alpha} \leq H$, and $\max\{|k-i|\} = n-2 << N \leq H$. There are two cases to be considered:

- Case 1 ($\Delta j_n = 0$): We have
$$\sum_{k=1}^{i-1}(k-i)H^{k-1}\Delta j_k + \sum_{k=i+1}^{n-1}(k-1)H^{k-1}\Delta j_k = 0.$$
This equation implies that $\sum_{k=1}^{i-1}(k-i)H^{k-1}\Delta j_k = 0$ and $\sum_{k=i+1}^{n-1}(k-i)H^{k-1}\Delta j_k = 0$, since $\Delta j_k < H$ and $|k-i| << H$. To simplify the following descriptions, we assume that i = odd. In the case of i = even, we can prove similarly.
  - $\sum_{k=1}^{i-1}(k-i)H^{k-1}\Delta j_k = 0$: When $n > 3$, the equation implies $(k-i)H^{k-1}\Delta j_k + (k+1-i)H^k \Delta j_{k+1} = 0$, $k = 1, 3, 5, \cdots, i-2$. The reason is that the absolute value of every coefficient $|(k-i)\Delta j_k|$ of the equation is less than H^2. From the above equation, we have $\Delta j_k = -\frac{k+1-i}{k-i} H \Delta j_{k+1}$. Assume $\Delta j_{k+1} \neq 0$, we have
$$|\Delta j_k| = \left|\frac{k+1-i}{k-i}\right| H |\Delta j_{k+1}| \geq \left|\frac{k+1-i}{k-i}\right| H,$$
since $|\Delta j_{k+1}| \geq 1$. From $1 \leq k \leq i-2$, we have $|\Delta j_k| \geq \frac{H}{2}$. It is contradictory to $|\Delta j_k| < \frac{H}{2}$, because $\alpha = 2$ when $n > 3$. Thus, we have $\Delta j_k = 0$, $1 \leq k \leq i-1$. When $n = 3$, the equation is null, since $i = 1$ is the only odd integer less than or equal to $n-1$.
  - $\sum_{k=i+1}^{n-1}(k-i)H^{k-1}\Delta j_k = 0$: When $n > 3$ and n = even, the equation implies $(k-i)H^{k-1}\Delta j_k + (k+1-i)H^k \Delta j_{k+1} = 0$, $k = i+1, i+3, \cdots, n-2$. $\Rightarrow \Delta j_k = \frac{k+1-i}{k-i} H \Delta j_{k+1}$. Assume $\Delta j_{k+1} \neq 0$, we have
$$|\Delta j_k| = \left|\frac{k+1-i}{k-i}\right| H |\Delta j_{k+1}| \geq \left|\frac{k+1-i}{k-i}\right| H,$$
since $|\Delta j_{k+1}| \geq 1$. From $i+1 \leq k \leq n-2$, we have $|\Delta j_k| \geq H$. It is contradictory to $|\Delta j_k| < H$. Thus, we have $\Delta j_k = 0$, $i+1 \leq k \leq n-1$. When $n > 3$ and n = odd, the proof is similar to the case of n = even. When $n = 3$ (with $i = 1$), the equation implies $H\Delta j_2 = 0$. Therefore, $\Delta j_2 = 0$.

  For the case of $\Delta j_n = 0$, we have $\Delta j_k = 0$, $1 \leq k \leq i-1$ and $i+1 \leq k \leq n-1$.
- Case 2 ($\Delta j_n \neq 0$): We have
$$\sum_{k=1}^{i-1}\big((k-i)\Delta j_k - \Delta j_n\big)H^{k-1} - \Delta j_n H^{i-1} + \sum_{k=i+1}^{n-1}\big((k-i)\Delta j_k - \Delta j_n\big)H^{k-1} = 0.$$
  Case 2.1 ($\Delta j_n = \pm c$, $c \geq 1$, c is a constant value): We have $\Delta j_{i-1} = \mp cH$, since the coefficient of H^{i-1} should be zero. Thus, we have $\Delta j_{i-1} \geq H$. It is conflict to $\Delta j_{i-1} < H$.
  Case 2.2 ($\Delta j_n = \pm\frac{H}{c}$, $c \geq 2$): We have $\Delta j_{i+1} = \pm\frac{1}{c}$, since the coefficient of H^i should be zero. It is conflict to Δj_{i+1} should be an integer. In either case, we have $\Delta j_n = 0$.

  Now, we can deduce that $\forall \vec{j}_1, \vec{j}_2 \in \bar{J}^n$, for each $\vec{e}_i \in I$, if $\vec{\phi}^{\mathrm{T}}(\vec{j}_1 - \vec{j}_2)\vec{r}^{\mathrm{T}}\vec{e}_i = \vec{r}^{\mathrm{T}}(\vec{j}_1 - \vec{j}_2)\vec{\phi}^{\mathrm{T}}\vec{e}_i$, then $\vec{j}_1 - \vec{j}_2 = k\vec{e}_i$.
- Memory conflict-free: Since $|\Delta j_i| < H$, we have $\vec{r}^{\mathrm{T}} \vec{j}_1 = \vec{r}^{\mathrm{T}} \vec{j}_2$ iff $\Delta j_i = 0$, $1 \leq i \leq n-1$, $\Delta j_n \neq 0$. Thus, if $\vec{r}^{\mathrm{T}} \vec{j}_1 = \vec{r}^{\mathrm{T}} \vec{j}_2$ and $\vec{\phi}^{\mathrm{T}} \vec{j}_1 < \vec{\phi}^{\mathrm{T}} \vec{j}_2$, then $\vec{\phi}^{\mathrm{T}} \vec{j}_1 + \sum_{k=0}^{n-2} H^k \leq \vec{\phi}^{\mathrm{T}} \vec{j}_2$ and $\forall e_{i \neq n}$, $\vec{\phi}^{\mathrm{T}} \vec{j}_1 + (n-i) < \vec{\phi}^{\mathrm{T}} \vec{j}_2$. Therefore, since $\vec{r}^{\mathrm{T}}\vec{e}_i = 0$ iff $i = n$, and $\left|\frac{\vec{\phi}^{\mathrm{T}}\vec{e}_i}{\vec{r}^{\mathrm{T}}\vec{e}_i}\right| = n - i$ for $i \neq n$, we have $\forall \vec{j}_1, \vec{j}_2 \in \bar{J}^n$, for each $\vec{e}_i \in I$, if $\vec{r}^{\mathrm{T}} \vec{j}_1 = \vec{r}^{\mathrm{T}} \vec{j}_2$ and $\vec{\phi}^{\mathrm{T}} \vec{j}_1 < \vec{\phi}^{\mathrm{T}} \vec{j}_2$, then $\vec{\phi}^{\mathrm{T}} \vec{j}_1 + \left|\frac{\vec{\phi}^{\mathrm{T}}\vec{e}_i}{\vec{r}^{\mathrm{T}}\vec{e}_i}\right| \leq \vec{\phi}^{\mathrm{T}} \vec{j}_2$ when $\vec{r}^{\mathrm{T}}\vec{e}_i \neq 0$ and $\vec{\phi}^{\mathrm{T}} \vec{j}_1 + \vec{\phi}^{\mathrm{T}}\vec{e}_i \leq \vec{\phi}^{\mathrm{T}} \vec{j}_2$ when $\vec{r}^{\mathrm{T}}\vec{e}_i = 0$.
- Unidirection: $\forall \vec{d}_i \in D$, $\vec{s}^{\mathrm{T}}\vec{d}_i = \vec{r}^{\mathrm{T}} T_u \vec{d}_i = \vec{r}^{\mathrm{T}}\vec{e}_i = r_i \geq 0$.

Since T_l is conflict-free for mapping $\mathcal{A}_I$ to $\mathcal{A}_Y$, from Theorem 3, we know that $T = T_l T_u$ is also conflict-free for mapping $\mathcal{A}_M$ to $\mathcal{A}_Y$. □

EXAMPLE 1 (Matrix multiplication). Since its dependence matrix $D = I$, we have $T_u = I$. Thus, $T = T_l = \begin{bmatrix} 2 & N & N+1 \\ 1 & N & 0 \end{bmatrix}$. The execution time is $t_e = (2 + N + N + 1)(N-1) + 1 = 2N^2 + N - 2$. The number of PE used is $\|P\| = (1+N)(N-1) + 1 = N^2$.

EXAMPLE 2 (Transitive Closure, TC). Although its dependence matrix $D = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$ is not a square one, D can be made fully permutable by skewing. That is, $X_{3\times3} D_{3\times5} = F_{3\times5}$, where X is the skew transformation matrix and $\forall f_{ij} \in F, f_{ij} \geq 0$. Here, $X = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ and $F = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$. Thus, we have $D = X^{-1}F \equiv BF$, where $B = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$. The matrix B is used to replace D to form the new dependence matrix of TC, such

that, $T_u = B^{-1} = X = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, $T_l = \begin{bmatrix} 2 & 2N & 2N+1 \\ 1 & 2N & 0 \end{bmatrix}$, and $T = T_l T_u = \begin{bmatrix} 2 & 2N & 4N+3 \\ 1 & 2N & 2N+1 \end{bmatrix}$. The execution time is $t_e = (2 + 2N + 4N + 3)(N-1) + 1 = 6N^2 - N - 4$. The number of PE used is $\|P\| = (1 + 2N + 2N + 1)(N-1) + 1 = 4N^2 - 2N - 1$.

THEOREM 5. *The number of PEs and the execution time of the linear array designed by Theorem 4 are both* $O(N^{n-1})$.

PROOF. From Theorem 4, we have $T = T_l T_u \equiv \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ s_1 & s_2 & \cdots & s_n \end{bmatrix}$ and $H = \alpha \max_{1 \le i \le n} |u_{ik}| N \equiv hN$, where h is an integral number. Thus,

$$\lambda_j = \sum_{i=1}^{n-1} \left(u_{ij}(n-i) + u_{nj}\right) H^{i-1} = \sum_{i=1}^{n-1} \left(u_{ij}(n-i) + u_{nj}\right) (hN)^{i-1} \equiv a_j N^{n-2} + o\left(N^{n-2}\right),$$

where a_j is an integral number. Thus, the execution time of the linear array designed by our method is

$$t_e = \left(\sum_{j=1}^{n} \lambda_j\right)(N-1) = \left(\sum_{j=1}^{n} \left(a_j N^{n-2} + o\left(N^{n-2}\right)\right)\right)(N-1) \equiv c_1 N^{n-1} + o\left(N^{n-1}\right) = O\left(N^{n-1}\right),$$

where c_1 is an integral number. Similarly,

$$s_j = \sum_{i=1}^{n-1} u_{ij} H^{i-1} = \sum_{i=1}^{n-1} u_{ij} (hN)^{i-1} \equiv b_j N^{n-2} + o\left(N^{n-2}\right),$$

where b_j is an integral number. Thus, the number of PEs used of the linear array designed by our method is

$$\|P\| = \left(\sum_{j=1}^{n} s_j\right)(N-1) = \left(\sum_{j=1}^{n} \left(b_j N^{n-2} + o\left(N^{n-2}\right)\right)\right)(N-1) \equiv c_2 N^{n-1} + o\left(N^{n-1}\right) = O\left(N^{n-1}\right),$$

where c_2 is an integral number. □

THEOREM 6. *Executing an FC-SURE* $\mathcal{A}_M = (J^n, V, F, D)$ *on a modular extensible linear array, the lower bound of the execution time is* $\Omega(N^{n-1})$.

PROOF. For a recurrence equation $v_j(\vec{j}_i + \vec{d}_i) = f_j(\cdots, v_i(\vec{j}_i), \cdots)$ in the FC-SURE, if $\vec{j}_i \notin J^n$, then the value of $v_i(\vec{j}_i)$ should be set by the initial value of the algorithm. For an n-dimensional FC-SURE, there are at least N^{n-1} initial values for the variable v_i.

Now, if we execute the FC-SURE on a linear array, because the input ports of the linear array are confined on the boundary PEs, there is only one input port for each variable. Since the array should satisfy the link conflict-free condition, the time for inputting all N^{n-1} initial values of a variable is at least N^{n-1} time steps. Thus, the lower bound of the execution time is $\Omega(N^{n-1})$. □

Now, we want to prove that the lower bound of the number of PEs required is $\Omega(N^{n-1})$ for executing an n-D FC-SURE on a linear array. Ramakrishnan and Varma [2] have shown a special case for matrix multiplication by formulating the computation of matrix multiplication as a game played with tokens on an undirected graph. We prove this lower bound by showing that every bisection of an n-D FC-SURE contains $\Omega(N^{n-1})$ edges. This implies that the memory storage required for executing an n-D FC-SURE is at least $\Omega(N^{n-1})$. Since the linear array is memory conflict-free, the lower bound of the number of PEs required is $\Omega(N^{n-1})$ for executing the n-D FC-SURE on the linear array.

THEOREM 7. *Every bisection of an n-D FC-SURE* $\mathcal{A}_M = (J^n, V, F, D)$ *considered as an undirected graph contains* $\Omega(N^{n-1})$ *edges.*

PROOF. The basic concept of this proof comes from Leighton's textbook [16] (Theorem 1.21: Every bisection of an n-D N-sided array contains at least N^{n-1} edges). Let $\mathcal{A}_I = (\bar{J}^n, V, F, I)$ be the equivalent MM-SURE of the FC-SURE $\mathcal{A}_M = (J^n, V, F, D)$ and be obtained from Theorem 1 by the unimodular matrix $T_u = [u_{ij}] = D^{-1}$. Define $\mu = \max_{i=1}^{n} \sum_{k=1}^{n} |u_{ik}|$ and $\mathcal{N} = \mu N$. Let undirected graph $\mathcal{D}$ be the graphical representation of the MM-SURE $\mathcal{A}_I$, where each node and link in $\mathcal{D}$ correspond to an index vector and a dependence vector in $\mathcal{A}_I$, respectively. A complete directed graph $\mathcal{G}$ is embedded to $\mathcal{D}$ by embedding the edge from node $v = [v_1, v_2, \cdots, v_n]^T$ to node $w = [w_1, w_2, \cdots, w_n]^T$ of $\mathcal{G}$ through the path $[v_1, v_2, \cdots, v_n]^T \to [w_1, v_2, \cdots, v_n]^T \to [w_1, w_2, \cdots, v_n]^T \to \cdots \to [w_1, w_2, \cdots, w_n]^T$. For any edge e of $\mathcal{D}$ connecting nodes $[j_1, j_2, \cdots, j_k, \cdots, j_n]^T$ and $[j_1, j_2, \cdots, j_k + 1, \cdots, j_n]^T$, there are, at most, $\frac{\mathcal{N}^{n+1}}{2}$ edges of $\mathcal{G}$ passing through e. The reason is that the edges of $\mathcal{G}$ from v to w passing through e if $v = [v_1, v_2, \cdots, v_k, j_{k+1}, j_{k+2}, \cdots, j_n]^T$ and $w = [j_1, j_2, \cdots, j_{k-1}, w_k, w_{k+1}, \cdots, w_n]^T$. Thus, the maximal number of feasible choices of v and w is $2\mathcal{N}^{n-1} j_k (\mathcal{N} - j_k) \le \frac{\mathcal{N}^{n+1}}{2}$. Then, for any edge e of $\mathcal{D}$, there are also, at most, $\frac{\mathcal{N}^{n+1}}{2} = \frac{\mu^{n+1} N^{n+1}}{2}$ edges of $\mathcal{G}$ passing through e.

Now, by contradiction, we assume that there is a bisection size $B < \frac{N^{n-1}}{\mu^{n+1}}$ in $\mathcal{D}$. Thus, at most, $B \frac{\mu^{n+1} N^{n+1}}{2} < \frac{N^{2n}}{2}$ edges of $\mathcal{G}$ pass that bisection. It is contradictory to that for any bisection of $\mathcal{G}$ there are at least $2 \frac{N^n}{2} \frac{N^n}{2} = \frac{N^{2n}}{2}$ edges of $\mathcal{G}$ passing through that bisection, since $\mathcal{G}$ is also a complete directed graph embedded in the n-D $N \times N \times \cdots \times N$ graph of $\mathcal{A}_M$. Therefore, we have every bisection of $\mathcal{D}$ contains at least $\frac{N^{n-1}}{\mu^{n+1}} = \Omega(N^{n-1})$ edges. □

THEOREM 8. *Executing an FC-SURE* $\mathcal{A}_M = (J^n, V, F, D)$ *on a modular extensible linear array, the lower bound of the number of PEs required is* $\Omega(N^{n-1})$.

PROOF. From Theorem 7, we have that every bisection of an n-D FC-SURE $\mathcal{A}_M = (J^n, V, F, D)$ contains $\Omega(N^{n-1})$ edges. Thus, for executing an n-D FC-SURE, only two cases will occur: Case 1: When time = t, half index vectors of the FC-SURE have been executed. Obviously, the memory storage required for executing the FC-SURE is at least $\Omega(N^{n-1})$. Case 2: When time = t, less than half index vectors of the FC-SURE

have been executed; but, when time = $t + 1$, more than half index vectors are executed. Then, when either time at t or at $t + 1$, the memory storage required for executing the FC-SURE is at least $\Omega(N^{n-1})$. Therefore, the memory storage required for executing an n-D FC-SURE is at least $\Omega(N^{n-1})$. It follows that, executing an FC-SURE on a modular extensible linear array, the lower bound of the number of PEs required is $\Omega(N^{n-1})$, since each PE in the linear array has constant memory storage. □

THEOREM 9. *An asymptotically optimal linear array can be designed in polynomial time for any FC-SURE by Theorems 1 and 4.*

PROOF. Optimality: It can be proven directly from Theorems 5, 6, and 8. Polynomial time: From Theorem 1, we know that the transformation of an FC-SURE to an equivalent MM-SURE by unimodular matrix can be done in polynomial time, and it depends only on the number of dimensions of the FC-SURE. From Theorem 4, we know that the mapping MM-SURE to the linear array only takes constant time. Hence, the theorem is proved. □

4 Conclusion

In this paper, the design of a unidirectional modular extensible linear array for an n-dimensional FC-SURE is studied. A polynomial time method is proposed that contains two major steps: In Step 1, the FC-SURE is transformed to an equivalent MM-SURE by a unimodular matrix. In Step 2, the MM-SURE is mapped to a unidirectional modular extensible linear array by a transformation matrix in a fixed form. Thus, the spacetime mapping transformation matrix for mapping the original FC-SURE to a linear array can be obtained by combining these two matrices together. Furthermore, the linear array designed by our method is asymptotically optimal in space and time.

Acknowledgments

This research was supported by the National Science Council of the Republic of China under contract NSC84-2213-E009-031.

References

[1] H.T. Kung and C.E. Leiserson, "Systolic Arrays (for VLSI)," *Proc. Sparse Matrix Symp.*, pp. 256-282. Soc. for Industrial and Applied Math, 1978.

[2] I.V. Ramakrishnan and P.J. Varman, "Modular Matrix Multiplication on a Linear Array," *IEEE Trans. Computers*, vol. 33, no. 11, pp. 952-958, Nov. 1984.

[3] P.J. Varman and I.V. Ramakrishnan, "Synthesis of an Optimal Family of Matrix Multiplication Algorithms on Linear Arrays," *IEEE Trans. Computers*, vol. 35, no. 11, pp. 989-996, Nov. 1986.

[4] V.K. Prasanna Kumar and Y.C. Tsai, "On Synthesizing Optimal Family of Linear Systolic Arrays for Matrix Multiplication," *IEEE Trans. Computers*, vol. 40, no. 6, pp. 770-774, June 1991.

[5] J.F. Myoupo, "A Way of Deriving Linear Systolic Arrays from a Mathematical Algorithm Description: Case of the Warshall-Floyd Algorithm," *Proc. Int'l Conf. Parallel Processing*, pp. I.575-I.579, 1991.

[6] S.K. Rao, "Regular Iterative Algorithms and Their Implementations on Processor Arrays," PhD thesis, Stanford Univ., 1985.

[7] P.Z. Lee and Z.M. Kedem, "Synthesizing Linear Array Algorithms from Nested for Loop Algorithms," *IEEE Trans. Computers*, vol. 37, no. 12, pp. 1,578-1,598, Dec. 1988.

[8] W. Shang and J.A.B. Fortes, "On Time Mapping of Uniform Dependence Algorithms into Lower Dimensional Processor Arrays," *IEEE Trans. Parallel and Distributed Systems*, vol. 3, no. 5, pp. 350-363, May 1992.

[9] K.N. Ganapathy and B.W. Wah, "Synthesizing Optimal Lower Dimensional Processor Arrays," *Proc. Int'l Conf. Parallel Processing*, pp. III.96-III.103, 1992.

[10] R. Varadarajan and B. Ravichandran, "Refinement Based Techniques for Mapping Nested Loop Algorithms onto Linear Systolic Arrays," *Integration, The VLSI J.*, vol. 14, pp. 249-277, Feb. 1993.

[11] P. Quinton, "Automatic Synthesis of Systolic Arrays from Uniform Recurrent Equations," *Proc. Int'l Symp. Computer Architecture*, pp. 208-214, 1984.

[12] M.E. Wolf and M.S. Lam, "A Loop Transformation Theory and an Algorithm to Maximize Parallelism," *IEEE Trans. Parallel and Distributed Systems*, vol. 2, pp. 452-471, Oct. 1991.

[13] W. Shang and J.A.B. Fortes, "Independent Partitioning of Algorithms with Uniform Dependencies," *IEEE Trans. Computers*, vol. 41, no. 2, pp. 190-206, Feb. 1992.

[14] J.C. Tsay and P.Y. Chang, "Designing Lower Dimensional Regular Arrays for Algorithms with Uniform Dependencies," *J. Parallel and Distributed Computing*, vol. 33, pp. 24-32, 1996.

[15] P.Y. Chang, "Design of Efficient Regular Arrays for Matrix Algorithms," PhD thesis, National Chiao Tung Univ., Hsinchu, Taiwan, R.O.C., 1995.

[16] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, 1992.