

시스톨릭 어레이 설계 시스템의 개발

유 기 영[†] · 이 성 우^{††} · 박 동 기^{†††} · 김 윤 호^{††††}

요 약

본 논문에서는 정규 순환방정식으로 표현된 n 차원의 문제로부터 1차원이나 2차원의 최적 평면 시스톨릭 어레이를 체계적으로 유도하는 방법과 이를 구현한 시스톨릭 어레이 설계 시스템을 제시한다. 구현된 시스템은 주어진 정규 순환방정식을 구문분석하여 문제공간, 자료중속 벡터와 초기값의 위치를 구한다. 이들 정보로부터 공간-시간 변환을 하여 자동으로 시스톨릭 어레이를 유도한다. 이 시스템은 유도된 시스톨릭 어레이의 병렬수행이 올바르게 이루어지는지를 시각적으로 검증할 수 있게 해 준다.

Development of a Systolic Array Design System(SADS)

Kee-Young Yoo[†] · Sung-Woo Lee^{††} · Dong-Ki Park^{†††} · Yoon Ho Kim^{††††}

ABSTRACT

This paper presents a systolic array design method which derives 1 or 2 dimensional optimal planar systolic arrays from a given n dimensional problem represented as a regular recurrence equation and its implementation called a systolic array design system(SADS). The SADS parses a regular recurrence equation and gets the information such as problem space, data dependence vectors, and initial data positions. Systolic arrays are automatically derived by the space-time transformation from the information to be obtained in the parsing phase. The SADS allows us to verify the parallel execution of the derived systolic array through the graphical interface.

1. 서 론

신호 및 영상처리, 행렬계산 등과 같이 계산량이 많은(compute-bound) 응용분야의 문제에 적합한 하드웨어적인 해결책인 시스톨릭 어레이를 설계하는 방법에 있어서 초기에는 대부분 경험적인 방법으로 설계되었다. 그 후 단일 배정 코드(single assignment code)나 정규 순환방정식(uniform recurrence equation

:URE)으로 표현된 알고리즘에 대해서 공간-시간 사상(space-time mapping)을 도입하여 체계적으로 시스톨릭 어레이를 설계하는 방법에 대한 많은 연구가 있었다[2, 5, 7, 8, 9, 12]. 그러나 대부분 설계방법은 n 차원 문제를 $(n-1)$ 차원의 시스톨릭 어레이로 설계하는 방법이다. 본 연구에서는 평면 어레이의 특징과 unimodular 변환으로 링크행렬과 자료 중속 행렬로 이루어진 diophantine 방정식으로 공간행렬을 구하고, 공간행렬의 커널공간의 시간간격을 최소로 하는 시간함수로 만들어진 공간-시간 변환은 n 차원 문제로부터 체계적으로 1차원이나 2차원의 시스톨릭 어레이로 설계한다.

1980년대 후반과 90년대 초반부터 소프트웨어로 시스톨릭 어레이의 구조 설계를 자동적으로 만들고

※본 논문은 한국과학재단의 '95 핵심 과제 연구비 지원에 의한 것임.

† 정 회 원:경북대학교 컴퓨터공학과

†† 준 회 원:경북대학교 컴퓨터공학과

††† 정 회 원:태창기술연구소 연구원

†††† 정 회 원:계명전문대학교 사무자동학과

논문접수:1996년 1월 11일, 심사완료:1997년 4월 9일

자 하는 시도가 있었다[1, 3, 7, 10, 11]. 그러나 이들 시스템은 주어진 문제로부터 시스템릭 어레이를 설계하는 과정의 일부분만을 보여준다. Bradley[1], Hartenstein[3] 및 Kung[7]은 시간-공간사상의 결과를 이용하여 어레이를 유도하는 시스템 수준이거나 시스템릭 어레이의 전체 설계과정 중 일부분만 구현하였다. 또한 시스템릭 어레이의 수행이 병렬로 이루어지기 때문에 수행결과를 검증하기가 쉽지 않기 때문에 수행을 검증할 수 있는 방법이 요구되는데 Sedukhin[11] 등은 설계된 시스템릭 어레이의 정보를 이용하여 시스템릭 어레이의 수행과정을 검증하였지만 사용자가 쉽게 수행과정을 검증하는데는 미흡하다.

따라서, 주어진 문제의 알고리즘으로부터 공간-시간함수를 체계적으로 산출하여 시스템릭 어레이를 자동으로 설계하고, 입력 배치 함수를 구하며, 이를 이용해 자료값을 배치하고 시뮬레이션하여 주어진 알고리즘이 설계된 시스템릭 어레이에서 올바르게 수행되는지를 검증해주는 시스템이 요구되어진다.

본 논문에서는 주어진 n 차원의 정규 순환 방정식으로부터 평면 시스템릭 어레이를 설계하는 방법과 이를 기반으로 시스템릭 어레이를 자동으로 설계하고 도시하며, 대화식으로 수행을 검증할 수 있도록 시뮬레이션을 해주는 소프트웨어 시스템의 설계 및 구현에 대해 설명한다. 본 연구에서 개발한 SADS 시스템은 주어진 문제로부터 표현된 순환방정식을 구문분석하여 설계정보를 추출하고 이들 정보를 이용하여 시스템릭 어레이를 자동 설계하므로 기존의 다른 시스템보다 쉽게 시스템릭 어레이를 설계한다. 본 논문에서 제시한 공간-시간 변환에서는 1차원이나 2차원의 시스템릭 어레이를 설계하기 위해 연결링크로 구성되어질 수 있는 모든 링크행렬에서 기하학적으로 서로 다른 행렬을 이용하여 공간행렬을 구한다. 이렇게 함으로써 공간조건을 만족하는 공간행렬을 효율적으로 구할 수 있다. 그리고 구해진 공간행렬의 커널을 이용하여 시간조건과 공간-시간조건을 만족하는 시간벡터를 구한다. SADS 시스템에서는 이 방법을 기반으로 가능한 여러 공간행렬과 시간행렬을 계산하여 이들 중 최적인 것을 선택하여 시스템릭 어레이를 설계한다. SADS 시스템은 유도된 시스템릭 어레이의 PE 배치와 링크의 연결을 X window에 그려서 사용자에게 시각적으로 보여 준다. 시스템의 또

다른 특징으로 수행 도중 각 PE에 어떤 자료들이 입력되는지, 또 각 PE 마다 병렬적으로 수행되는 계산과정이나 계산결과를 대화식으로 검증할 수 있도록 pop up 윈도우를 통하여 보여준다. 주어진 문제가 설계된 시스템릭 어레이상에서 올바르게 수행되는지 쉽게 검증할 수 있도록 straightline 프로그램으로 수행과정을 윈도우에 나열하여준다. 본 연구에서 구현된 시스템을 이용하여 여러 가지 문제로부터 시스템릭 어레이를 설계할 수 있을 것이다.

본 논문의 구성은 다음과 같다. 제 2장에서는 소프트웨어 시스템의 구현에 기본이 되는 이론적 배경에 대해서 설명한다. 제 3장에서는 구현된 소프트웨어에 대해서 설명하며, 제 4장에서는 수행의 예를 보여준다. 마지막으로 5장에서 결론을 맺는다.

2. 이론적 배경

2.1 문제의 표현

주어진 문제는 n 개의 중첩 루프(loop)는 아래와 같은 계산문제로 구성되는 정규 순환방정식으로 표현된다.

$$a_i(p) = f(a_1(p-d_1), a_2(p-d_2), \dots, a_m(p-d_m)) \quad (1)$$

여기서, $p = [p_1, \dots, p_n]$, $p_j \in \mathbb{Z}$, $(1 \leq j \leq n)$ 는 정규 순환방정식이 계산되는 n 차원의 계산점(computation point)이다. 이와 같은 계산점들의 집합을 계산공간(computation space)이라 한다. 변수 a_j ($1 \leq j \leq m$)는 정규 순환방정식에서 나타나는 배열변수들이다. 벡터 d_k ($1 \leq k \leq m$)는 변수 a_k 에 대한 자료종속 벡터(data dependence vector)를 나타내며, 행렬 $D = [d_1, \dots, d_m]$ 를 자료종속 행렬(data dependence matrix)이라 부른다.

[예 1] 두 행렬 A, B 의 곱셈문제는 아래와 같이 표현된다.

$$1 \leq i \leq n, 1 \leq j \leq n, 1 \leq k \leq n$$

$$C[i, j, k] = C[i, j, k-1] + A[i, j-1, k] \times B[i-1, j, k] \quad (2)$$

문제의 계산공간은 $\{p = (i, j, k)^T | 1 \leq i \leq n, 1 \leq j \leq n,$

$1 \leq k \leq n$ 이며, 변수 A, B, C 에 대한 자료종속 벡터 $\vec{d}_1, \vec{d}_2, \vec{d}_3$ 라 하면 자료종속 행렬은 아래와 같다.

$$D = [\vec{d}_1, \vec{d}_2, \vec{d}_3] = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

2.2 공간-시간 변환(space-time transformation)

문제로부터 시스틀릭 어레이의 합성은 공간-시간 함수에 의한 문제공간에서 어레이 공간으로의 변환으로 이루어진다. 이들 함수는 문제의 각 계산점을 시스틀릭 어레이의 한 특정 위치의 PE와 수행될 시간에 사상한다[12]. 변환 행렬을 M 이라 하면, M 은 식 (4)와 같이 표현된다.

$$M = \begin{bmatrix} S \\ T \end{bmatrix} \begin{cases} S: \text{공간 행렬(공간 함수)} \\ T: \text{시간 벡터(시간 함수)} \end{cases} \quad (4)$$

p 를 문제의 한 계산점이라 할 때, 공간-시간함수에 의해 다음 식 (5)와 같이 변환된다.

$$Mp = \begin{bmatrix} S \\ T \end{bmatrix} p = \begin{bmatrix} Sp \\ Tp \end{bmatrix} = \begin{bmatrix} \text{space} \\ \text{time} \end{bmatrix} \quad (5)$$

즉, 계산점 p 는 시스틀릭 어레이에서 space 위치의 PE에서 time 시간에 수행됨을 의미한다. 주어진 문제의 각 계산점을 평면 시스틀릭 어레이로 사상시키기 위해서는 공간함수(행렬) S 와 시간함수(벡터) T 를 계산하여야 한다.

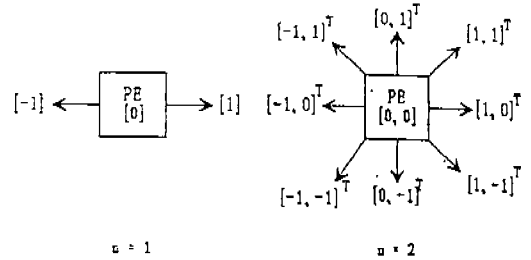
2.2.1 공간함수(space function)

공간함수 S 에 의해 문제의 각 계산점은 시스틀릭 어레이의 한 PE에 사상된다. 설계될 평면 시스틀릭 어레이의 네트워크는 그림 1에서 보여준 연결 링크로 이루어지므로 공간함수는 문제의 각 자료 종속 벡터를 네트워크를 이루는 각 연결 링크에 대응시킨다. 그러므로 공간사상 S 에 따라 시스틀릭 어레이의 PE 위치와 연결 링크의 모양이 달라진다.

그림 1에서 보여준 연결링크를 벡터로 표현하면 아래와 같다.

$$A_1 = \{[-1], [0], [1]\},$$

$$A_2 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \pm \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \pm \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \pm \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \pm \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$$



(그림 1) 평면 시스틀릭 어레이의 연결링크
(Fig. 1) The intercommunication link of planar systolic arrays

여기서, A_1 는 1차원 어레이의 연결링크 벡터이고, A_2 는 2차원 어레이의 연결링크 벡터이다.

m 개의 자료종속 벡터로 이루어진 $n \times m$ 자료종속 행렬을 D 라하고, 유도된 시스틀릭 어레이의 연결링크 벡터(A_1, A_2)중에서 m 개의 열벡터로 가지는 $k \times m$ 링크행렬을 L 이라 할 때, S 는 diophantine 방정식 $SD = L$ 를 만족하는 $k \times n$ 정수행렬이다.

본 연구에서는 설계방법에 있어서 먼저 $k(k=1, 2)$ 차 평면 시스틀릭 어레이상에 대해 그림 1과 같은 연결링크로만 이루어지는 기하학적으로 형태가 다른 어레이 네트워크의 구조를 미리 생성하고, 이를 사용하였다. 만들어질 수 있는 모든 링크행렬 L 중에서 unimodular 변환을 이용하여 기하학적으로 같은 것을 제거하고 기하학적으로 다른 네트워크 구조를 가지는 L 을 구하였다. 미리 생성된 기하학적으로 다른 네트워크는 모든 문제에 대한 시스틀릭 어레이를 설계하는데 이용되며, 그 수는 그리 많지 않은 고정된 개수이다[12].

주어진 문제로부터 $n \times m(n \leq m)$ 자료종속 행렬 D 를 구하고, 미리 생성된 기하학적으로 다른 어레이 네트워크를 나타내 주는 각 링크행렬 L 에 대해서 D 를 각 L 에 사상시키는 공간함수 S 를 계산하는 알고리즘은 다음과 같다.

단계 1: 자료종속 행렬 D 를 $D = [D' D'']$ 로 분리한다.

여기서, D' 은 $n \times n$ 정규행렬이며 부행렬이 가 한다.

단계 2: 부행렬 D'' 의 Smith normal form을 구한다.

즉, $RD' C = D''$ 를 만족하는 행렬 R, C 및 대각행렬

D_n 를 구한다.

단계 3: 미리 생성된 $k \times n$ 연결링크 행렬 L' 를 선택한다.

단계 4: 식 $P=L'C$ 를 계산한 후, $S=PD_n^{-1}R$ 에 의해 공간행렬 S 를 계산한다.

단계 5: $L'=SD'$ 를 계산한다. 행렬 L' 의 각 열벡터는 연결링크임을 확인한다.

그러므로 Diophantine 방정식 $SD=L$ 를 만족하는 공간행렬 S 를 구한다.

설계된 시스템릭 어레이의 각 연결링크는 그림 1에서 주어진 허용 가능한 연결링크 중에 하나이다.

2.2.2 시간함수(time function)

시간함수 T 는 문제의 각 계산점들이 시스템릭 어레이에서 수행되는 시각을 계산한다. 두 계산점을 \vec{p} 와 $\vec{p}+\vec{d}_i$ 라 하면(\vec{d}_i 는 자료 종속 벡터임), 이 두 계산점 사이의 링크에는 전송 시간이 필요하다. 따라서 다음 식 (6)의 조건이 성립되어야 한다(시간 조건).

$$T(\vec{p}+\vec{d}_i)-T_p=T \vec{d}_i \geq 1 \tag{6}$$

또 주어진 시간 t 에서 시스템릭 어레이상의 한 특정 PE 위치에서 계산되어지는 계산점 p 는 유일하다. 즉, 두 계산점 p 와 q 는 시스템릭 어레이의 한 특정 PE에서 같은 시각에 사상될 수 없다. 따라서 다음과 같은 식 (7)이 만족되어야 한다(공간-시간 조건).

$$\begin{bmatrix} S \\ T \end{bmatrix} (q-p) \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7}$$

공간행렬(함수) S 를 계산한 후, 시간조건과 공간-시간조건을 만족하는 시간함수 T 를 계산한다. 본 연구에서 공간함수 S 의 커널공간(kernel space)의 베이스 벡터로 표현되는 모든 계산점 p 를 선형 스케줄링 해주는 시간벡터 T 를 계산한다.

문제로부터 주어진 $n \times m$ 자료 종속 행렬 $D = \begin{bmatrix} \vec{d}_1 \\ \vdots \\ \vec{d}_m \end{bmatrix}$ 와 앞에서 구한 $k \times n$ 공간행렬 S 를 이용하여 시간벡터(함수) T 를 계산한다. 먼저 공간행렬 S 의 커널공간 $X(S)$ 의 베이스 벡터 $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_{(n-k)}\}$ 를 구한다. 베이스 벡터의 모든 순열의 각 경우에 대

해서 다음 식 (8)을 만족하는 시간벡터 T 를 계산한다.

$$\begin{aligned} \min T(p^* - p_*) + 1 &\geq |X(S)| \quad \text{subject to} \\ \vec{T}(\vec{d}_1, \dots, \vec{d}_m) &\geq (1, \dots, 1) \end{aligned} \tag{8}$$

여기서, 계산점 p^* 와 p_* 는 계산순서상 최종점과 최초점으로 공간행렬 S 의 커널공간 $X(S)$ 의 베이스 벡터 $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_{(n-k)}\}$ 들의 일차결합(linear combination)으로 표현되고, $|X(S)|$ 는 커널공간의 크기이다. 식 (8)을 만족하는 시간벡터 T 는 공간행렬의 커널공간에 있는 각 격자점(lattice)의 시간간격이 1보다 크며 최소가 되는 순차적인 시간을 사상시키는 벡터로 시간조건을 만족하는 벡터이다. 시스템릭 어레이 설계에 있어서 자료충돌 회피 조건은 아주 중요한데, 본 연구에서 계산된 S 와 T 는 자료충돌 회피조건을 자동으로 만족하기 때문에 추가로 검증할 필요가 없다.

2.2.3 최적 시스템릭 어레이

문제로부터 공간함수 S 에 의해 여러 종류의 시스템릭 어레이의 배치를 얻을 수 있고, 시간함수 T 에 의해 여러 종류의 실행 시간을 얻을 수 있다. 그러므로 시스템릭 어레이를 설계하는데 있어서 최적화의 기준을 설정하며 최적 시스템릭 어레이를 얻는 것이 중요하다. 본 연구에서는 생성될 수 있는 모든 평면 시스템릭 어레이에 대응하는 공간함수 S 중에서 PE의 개수를 최소화 하는 공간함수를 선택하고, 또한 PE의 효율을 극대화시키는 시간함수 T 를 선택한다.

2.3 입력자료 배치함수(I/O function)

이 절에서는 문제-크기 시스템릭 어레이 상에서 공간-시간 사상에 의하여 어레이 내부나 외부에 사상되는 입력 자료를 적절히 배치하는 방법을 대수적 표현으로 제시한다. 문제-크기 시스템릭 어레이 상에서 입력 자료의 위치는 일차적으로 입력 자료들에 공간-시간 사상 기법을 적용하여 얻어지는 정보들로부터 그 초기 위치가 구해진다. 이때 어레이 내부로 진입한 입력 자료들이 있으면 각각의 입력 자료에 상응하는 경계 PE를 기준으로 어레이의 외부로 후퇴시킴으로써 모든 외부 입력과 출력이 경계 PE에서만 일어나도록 한다.

2.3.1 입력 자료의 초기 위치

S와 T를 공간 함수와 시간 함수라 하면, 공간-시간 변환으로부터 변수 v의 초기 입력 자료 $\bar{i}_v \in I^m$ 가 $\Phi = T \bar{i}_v$ 시간에 $\Psi = S \bar{i}_v$ 처리요소에 입력된다는 정보가 얻어진다. 변수 $v \in V$ 의 자료들이 흐르는 링크 \hat{d}_v 와 이 링크가 가지는 지연 시간 $\hat{\pi}_v$ 에 대해, 한 입력 변수 \bar{i}_v 가 t_0 (시간 사상에 의해 주어지는 최초의 계산시간)에서의 위치 $\Omega_0(\bar{i}_v)$ 를 구하는 식은 다음 (9)와 같다.

$$\Omega_0(\bar{i}_v) = \Psi - \lfloor \frac{\Phi - t_0}{\hat{\pi}_v} \rfloor \hat{d}_v \tag{9}$$

2.3.2 입력 자료의 후퇴시간

초기 입력 자료의 어레이 내부 진입 여부 및 정도는 초기 입력 자료가 외부 입력되는 경계 PE B를 기준으로 파악된다. 또한, 외부 입력이 경계 PE에서 일어나도록 하기 위해서, B에 대해서 진입한 만큼 입력 자료를 후퇴시키는 방법을 적용한다. 우선 \bar{i}_v 의 경계 PE $B(\bar{i}_v)$ 를 구하고, $B(\bar{i}_v)$ 의 위치를 기준으로 하고, 링크 벡터 \hat{d}_v 를 고려하여 $\Omega(\bar{i}_v)$ 의 위치를 살펴보면 한 입력 자료가 어레이 내부로 진입한 정도 $\alpha(\bar{i}_v)$ 가 판별된다. 이것은 아래 식 (10)과 같이 표현된다.

$$\alpha(\bar{i}_v) = \hat{\pi}_v ((\Omega_0(\bar{i}_v) - B(\bar{i}_v)) \odot \hat{d}_v) \tag{10}$$

여기서, \odot 는 다음과 같은 연산자이다. 만약 $x \odot y = c$ 이면, x는 c와 y의 스칼라곱이다(c는 상수). 식 (10)으로부터 $\alpha(\bar{i}_v)$ 의 값이 $\alpha(\bar{i}_v) \geq 0$ 인 것이 하나라도 있으면, 어레이 내부로 진입한 입력 자료가 있는 것이다. 따라서, 모든 외부 입력이 경계 PE에서 일어나게 하기 위하여, 어레이 내부에 최대한로 진입한 정도에 대해서 후퇴를 시켜주어야 한다. 변수 v에 대해서 최대 진입한 값 $\alpha(\bar{i}_v)$ 를 α_v^m 로 할 때 시간의 관점에서 보면 α_v^m 의 값은 식 (9)의 floor 함수에 의해 다음과 같이 보정되어야 한다. 이렇게 보정된 값을 변수 v에 대한 최대 후퇴 시간 γ_v^m 이라 하며, 다음 식 (11)과 같이 표현 된다.

$$\gamma_v^m = \alpha_v^m - ((T \bar{i}_v^m - t_0) \bmod \hat{\pi}_v) \tag{11}$$

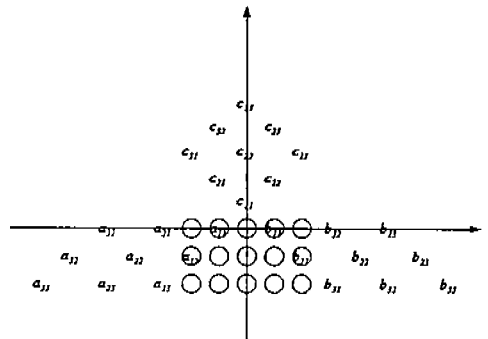
여기서 \bar{i}_v^m 는 최대한로 진입한 입력자료의 인덱스이다.

지금까지는 각 변수들을 개별적으로 고려하였으나, 최종적인 입력 스케줄링에서는 각 변수들을 총괄적으로 고려하여야 한다. 따라서, 각 변수에 대한 최대 후퇴 시간들로부터 최대의 후퇴 시간값 $\gamma^{\max} = \max \{\gamma_1^m, \dots, \gamma_k^m\}$ 을 구하고, 이 값을 최종적인 입력 배치 과정에 반영시켜 주어야 한다. 입력 자료들의 최종적인 초기 위치를 구하는 함수 $\Omega(\bar{i}_v)$ 는 다음 식 (12)와 같다.

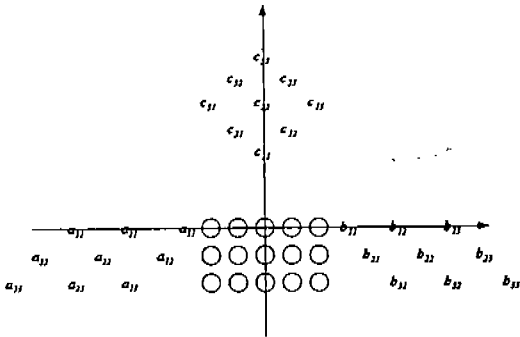
$$\Omega(\bar{i}_v) = B(\bar{i}_v) - (\gamma^{\max} + 1 - (\alpha(\bar{i}_v) - ((T \bar{i}_v^m - t_0) \bmod \hat{\pi}_v))) \hat{d}_v \tag{12}$$

[예 2] 예 1에서 행렬 곱의 정규 순환 방정식이 $N=3$ 일 때, 생성되는 시스톨릭 어레이와 입력 자료의 위치를 구하는 문제를 고려한다. $\begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$, $T = [1 \ 1 \ 1]$ 를 선택한다면 생성된 시스톨릭 어레이는 그림 2와 같다. 변수 A에 대응되는 연결링크는 $[1, 0]^T$, 변수 B는 $[-1, 0]^T$, 그리고 변수 C는 $[0, -1]^T$ 이다. 이들 링크의 지연시간은 모두 1이다.

우선 각 변수에 대한 입력자료들의 인덱스 공간상의 위치는 $A(i, 0, k)$, $B(0, j, k)$, $C(i, j, 0)$ 인데 이들을 식 (9)에 의해 최초 계산시간에 사상되는 위치 $\Omega_0(\bar{i}_v^m)$ 를 구하면 그림 2와 같다. 이들 변수들에 대해 각각의 최대 후퇴 시간 γ_v^m 을 구하면 $\gamma_A^m = 2(\bar{i}_A^m = A_{11})$, $\gamma_B^m = 2(\bar{i}_B^m = B_{11})$, $\gamma_C^m = 0(\bar{i}_C^m = C_{11})$ 이고 $\gamma^m = 2$ 이다. 최종적인 최대 후퇴 시간이 구해지면 모든 입력 자료



(그림 2) 최초 계산시간에 사상되는 초기 입력 배치 (Fig. 2) Initial input layout mapped at the first computation time



(그림 3) 최대 후퇴시간에 의해 재배치된 입력 배치
(Fig. 3) Input layout relocated by the maximum retreat time

를 식 (12)로 사상하여 어레이의 외부에 재배치한다 (그림 3).

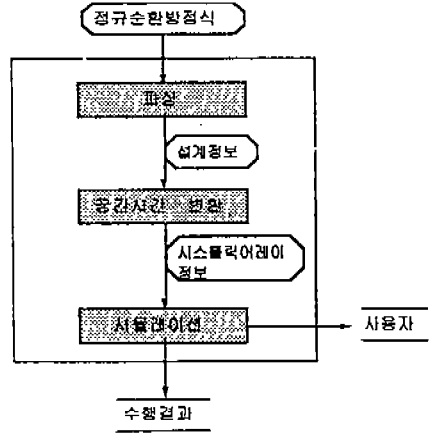
3. 설계 시스템 SADS의 구현

3.1 시스템의 전체 구성

시스템 SADS는 크게 세 단계로 구성된다. (1)파싱 (Parsing) 단계:주어진 정규 순환방정식으로부터 컴파일 기술을 이용하여 구문분석(parsing)을 하며, 시스템릭 어레이를 설계하는데 필요한 정보를 구하는 전처리 단계이다. (2)공간-시간 변환 단계:전처리 단계에서 구해진 여러 가지 정보를 가지고, 공간-시간 변환에 필요한 알고리즘으로 최적의 평면 시스템릭 어레이를 설계한다. (3)시뮬레이션 단계:설계된 문제-크기 혹은 고정-크기 시스템릭 어레이를 시각적으로 보기 쉽게 X window상에 도시하고, 시스템릭 어레이의 수행이 완전한가를 검증할 수 있게 시뮬레이션한다. 다음 그림 4는 SADS의 3단계를 보여준다. 각 단계는 개념적으로는 순차적 순서로 수행되지만 어느 단계든지 독립적인 순서로도 수행이 가능하다.

세 단계로 구성된 SADS 시스템은 X window상에서 크게 사용자 인터페이스 기능, 파싱 기능, 공간-시간 변환 단계, 그래픽 디스플레이 기능 및 검증 기능으로 이루어진다.

그림 5는 SADS 시스템의 초기 인터페이스를 보여준다. 크게 네 부분으로 나누어지는데, 시스템의 이름을 표시한 부분, 수행 제어 버튼 부분, 그려질 시스템을



(그림 4) 시스템의 기능별 구성
(Fig. 4) Functional Configuration of the system

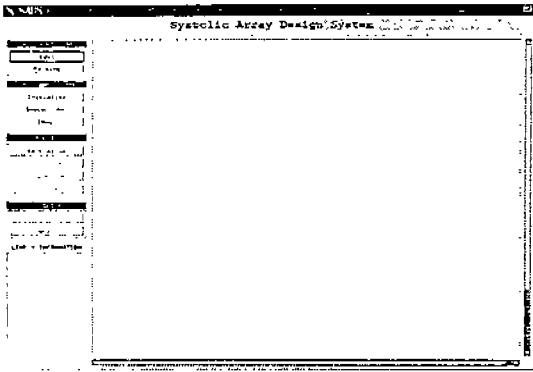
릭 어레이의 연결 링크를 도시하는 부분, 그리고 설계된 시스템릭 어레이를 그리는 윈도우 부분이 있다. 기능에 따라 수행 제어 버튼 부분은 크게 4가지로 구분된다. 먼저 전처리 기능, 공간-시간 변환, 시뮬레이션, 마지막으로 종료 및 자료의 출력으로 구분되어진다.

3.2 파싱

파싱단계는 그림 6과 같은 형식의 정규 순환방정식으로 표현된 문제를 입력받는다. 상수정의 부분에서는 방정식에서 사용하는 상수를 정의하고, 그 아래에 문제를 정규순환 방정식으로 표현한다. 그 다음에는 방정식에서 사용되는 변수들의 초기위치가 주어지고, 그 아래에 그것의 초기값이 주어진다. 이때 초기값이 주어지지 않을 수도 있다. 그러면 SADS는 시뮬레이션할 때 계산 결과를 인덱스를 가진 변수들의 문자열로써 표현한다. 그러나 초기값이 주어진다면 계산결과를 그 초기값을 이용해 숫자로 표현된다. 주어진 문제의 시스템릭 어레이를 설계하기 위해 시스템 SADS에 입력되는 입력화일의 형태는 다음 그림 6과 같다.

[예 3] 입력 형식에 맞게 만든 두개의 3×3 행렬 A와 B의 곱셈 C=AB의 정규 순환 방정식으로 표현된 원시 화일은 다음 그림 7과 같다.

정규 순환 방정식이 주어지면 SADS는 파싱(Parsing)을 수행한다. 파싱은 syntax directed translation 기법



(그림 5) 시스템 SADS의 초기 인터페이스
(Fig. 5) Initial interface of system SADS

```

상수 정의
%
루프 선언
순환방향식 몸체
%
원소의 초기위치
%
원소의 초기값(option)
    
```

(그림 6) 문제의 입력 형식
(Fig. 6) Input format of a problem

```

N= 3
%
1 ≤ i ≤ N, 1 ≤ j ≤ N, 1 ≤ k ≤ N;
  C[i,j,k] = C[i,j,k-1] + A[i,j-1,k] * B[i-1,j,k]
%
1 ≤ i ≤ N, 1 ≤ j ≤ N, k = 0;   C[i,j,k] = C(i,j)
1 ≤ i ≤ N, j = 0, 1 ≤ k ≤ N;   A[i,j,k] = A(i,k)
i = 0, 1 ≤ j ≤ N, 1 ≤ k ≤ N;   B[i,j,k] = B(j,k)
%
    
```

(그림 7) 행렬곱셈의 입력파일
(Fig. 7) Input file of matrix multiplication

을 사용하여 순환 방향식을 구문분석(parsing)하는 동시에 인덱스 제어 변수의 상계와 하계를 찾아서 계산 공간을 구하고, 방정식의 몸체로부터 자료종속 벡터

를 구하고, 방정식으로 표현된 연산을 수행하기 쉽게 후위표기법(postfix)으로 변환시키며, 배열값의 초기 위치를 추출한다. 마지막으로 배열 입력값의 초기 위치를 구하여 설계화일을 만든다.

3.3 공간-시간 변환

공간-시간함수를 구하기 위해 먼저 설계할 평면 시스틀릭 어레이의 차원(1차 혹은 2차)을 사용자로부터 입력받는다 차원이 결정되면 미리 생성되어 있는 어레이 네트워크(그림 1)로 자료흐름을 사상(mapping)시키는 공간함수 S 들을 계산한다. 얻어진 각 공간함수에 대하여 PE의 효율(utilization)을 극대화시킬 수 있는 시간함수 T 를 계산한다. 이렇게 구해진 공간-시간 함수들을 사용자에게 보여주면서 사용자가 PE의 개수가 최소이고, PE의 계산 효율이 최대한 함수를 선택하게 한다.

공간-시간 함수가 선택되면 식 (5)에 의해 시스틀릭 어레이가 생성된 후, 시스틀릭 어레이의 각 PE들이 올바른 계산을 하기 위해 연결링크를 따라 흐르는 자료가 올바른 시간에 올바른 PE에 도달할 수 있게 배치되어야 한다. 이를 위하여 전처리 단계에서 구한 초기값의 위치 정보와 공간 및 시간함수를 이용하여 초기값이 시스틀릭 어레이상에 위치할 좌표를 정한다. 입력자료의 위치를 정한 후, 설계된 평면 시스틀릭 어레이를 도시하고 시뮬레이션을 하는데 필요한 정보를 화일에 저장한다.

3.4 디스플레이 기능 및 시뮬레이션 기능

시스템 SADS는 X window상에 시스틀릭 어레이를 그리고, 시스틀릭 어레이의 수행을 시뮬레이션할 수 있고, 한 순간 PE의 계산 결과를 straight line 프로그램으로 보여주어 검증할 수 있다.

3.4.1 초기화

이 단계에서는 공간-시간 변환 단계의 결과 파일로부터 시뮬레이션에 필요한 모든 정보를 읽어 와서 시뮬레이션에 필요한 자료구조를 완성한다. 그 결과로 왼쪽 하단의 작은 윈도우에는 한 PE와 링크 방향 및 변수명과 지연시간을 보여주고, 주 윈도우에는 설계된 평면 시스틀릭 어레이의 전체 모습을 그려준다. 그리고 시스틀릭 어레이의 외부 연결링크를 가진 경

계 PE에는 해당 경계 PE를 통해 입력될 입력 자료가 그것의 입력시점과 함께 하나의 노드를 구성하여 연결 리스트(linked list)로 연결되어져 있다. 윈도우에 그려진 시스템릭 어레이에서 임의의 한 PE 위치를 마우스로 클릭하면 그 PE에서 각 변수에 흐르는 자료에 대한 정보를 pop up 윈도우에 나타내준다.

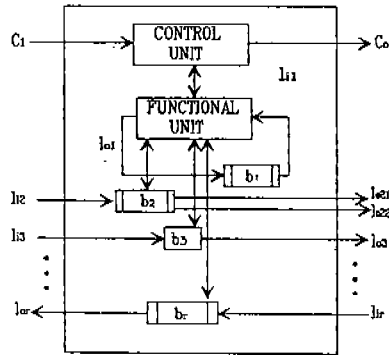
3.4.2 시뮬레이션의 수행

초기화가 끝나면 본격적인 시뮬레이션을 수행할 수 있는데 SADS는 시스템릭 어레이가 한 스텝씩 수행하면서 그 과정을 사용자에게 효과적으로 보여주며 검증할 수 있게 한다. 물론 시스템릭 어레이에 자료의 흐름이 하나도 없을 때까지 연속적으로 계속 수행할 수도 있다. 시스템릭 어레이의 모든 PE가 한 스텝을 수행한 후, 사용자가 임의의 PE를 누르면 각 링크를 따라 흐르는 자료를 pop up 윈도우에 보여준다. 병렬로 수행되는 모습을 눈에 띄게 하기 위해서 입력 자료에 의해 의미 있는 계산을 수행한 PE의 색깔을 빨간 색으로 변하게 하였고, 호스트는 검은 색으로 변하게 하는 동시에 각 PE에 계산되는 배열의 원소를 문자로 보여주었다.

SADS의 PE 내부 구조는 그림 8에서 보여준 것과 같다. 각 PE는 구조체로 표현되었으며 구성요소로 입력포트, 출력포트, 버퍼, 및 수행할 함수에 대한 포인터 등이 있다. PE의 수행을 위하여, 이웃한 PE에서 가져온 자료를 입력으로 받는 버퍼를 입력포트 버퍼라 한다. 각 PE에는 각 링크의 방향으로 입력포트 버퍼가 존재한다. 그리고 PE가 연산을 하고 난 뒤, 그 결과를 이웃한 PE에 전달해 주기 위한 버퍼가 있는데 이를 출력포트 버퍼라 하며 입력포트 버퍼처럼 각 링크에 존재한다. 또한, PE는 입력포트 버퍼의 값을 작업 버퍼(working buffer)로 읽어들이어서 연산을 수행한다. 그리고 연산 결과를 출력포트 버퍼에 전달한다. 다시 말해서 각 PE에서 수행하는 함수는 입력포트에 값이 도달된 후, 작업 버퍼에 값을 전송하여, 의미 있는 연산을 하며, 그 계산된 결과의 값을 출력포트로 전송한다. 그리고 출력포트의 값이 그 변수의 링크 방향을 따라 이웃한 PE의 입력포트 버퍼에 전달된다. 한 PE에서 이웃 PE로의 값 전달은 지연시간에 따라 연결 리스트로 구현하였다. 다음은 일반적인 PE의 연산이다.

```

Begin
/* 입력포트에서 임시 버퍼로 데이터를 전송한다. */
for ( pe=1; pe ≤ m; pe++ )
    work_reg[pe] = in_port[pe];
/* PE 의 실제 연산을 한다 (1 ≤ i ≤ m) */
    work_reg[i] = f(work_reg[i]);
/* 임시 버퍼에서 계산된 계산 결과물 출력 버퍼에 전송한다 */
for ( pe=1; pe ≤ m; pe++ )
    out_port[pe] = work_reg[pe]
End
    
```



(그림 8) 시스템릭 어레이의 PE 내부구조
(Fig. 8) The internal structure of PE of systolic arrays

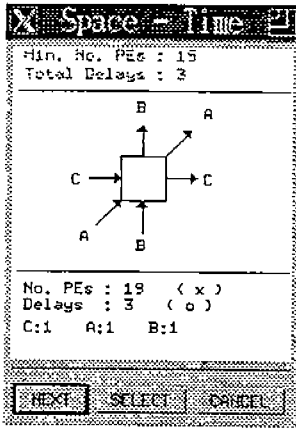
3.4.3 수행결과와 출력

SADS 시스템은 수행 후, 연산의 계산 과정을 사용자에게 보여주기 위해, 매 번 PE가 의미 있는 수행을 할 때마다 PE의 계산 과정을 연결 리스트로 저장해 둔다. 그래서 계산이 완료된 후 사용자가 원한다면, 그 계산 과정을 straight line 프로그램으로 디스플레이 되게 하여 시각적으로 알아볼 수 있도록 하였다. 따라서, 사용자는 주어진 정규 순환방정식이 공간-시간 함수에 의해 설계된 평면 시스템릭 어레이상에서 올바르게 병렬 수행됐는가를 검증할 수 있다.

수행이 모두 끝난 후 최종 결과의 검증용 위해 특정 배열의 한 원소를 선택하면 정해진 배열의 원소가 계산되어 온 과정을 straight line 프로그램으로 보여준다. 물론 계산 결과를 계산된 숫자로 볼 수도 있다. 이를 검사하므로써 평면 시스템릭 어레이가 바르게 설계되었는지를 쉽게 검증할 수 있다.

4. 수행 예(example)

여기서는 앞에서 언급된 예 2의 3×3 행렬곱셈이 수행되는 평면 시스틀릭 어레이를 시스템 SADS로 설계한 예를 보여준다. 예 3과 같이 주어진 행렬곱셈의 입력화일을 파싱하여 문제공간, 자료종속 행렬, 및 초기값의 위치정보를 추출하여 설계정보화일을 만든다. 다음 과정으로 공간-시간 변환을 수행한다. 계산된 가능한 공간 및 시간 함수 중에서 다음 그림 9와 같은 공간-시간함수를 선택하는 대화상자를 통해서 사용자가 공간 및 시간함수를 정한다.

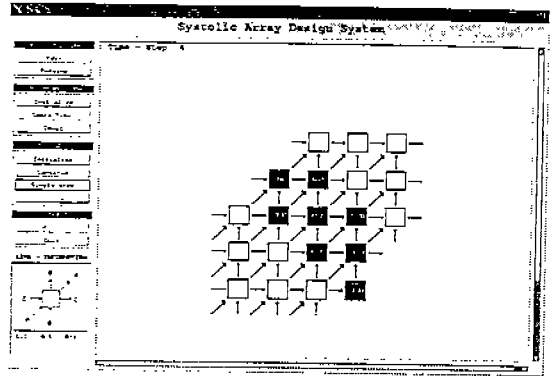


(그림 9) 공간 및 시간함수선택화면
(Fig. 9) Selection window for space and time functions

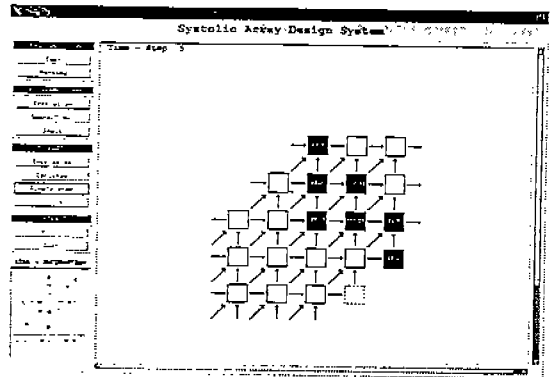
만약, 그림 9와 같이 사용자가 공간함수 $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ 와 시간함수 $T = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ 를 선택한다면 생성된 시스틀릭 어레이는 그림 10과 같다. 변수 A에 대응되는 연결링크는 $[1, 1]^T$ 이며, 변수 B는 $[0, 1]^T$, 변수 C는 $[1, 0]^T$ 이다.

그림 10과 11은 연속적인 두 시각의 상황을 보여준다. 검은 색으로 채워진 PE는 어떤 의미 있는 계산을 수행하는 것을 보여주고 있어서 자료의 시각별 움직임을 한 눈에 볼 수 있다. 매 스텝마다 어레이가 수행된 후 각 윈도우에 있는 각 PE를 클릭하면 그 PE가 가지는 정보를 보여주기도 한다.

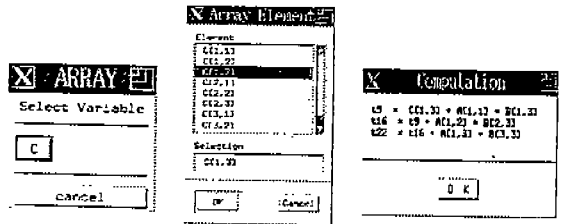
전체 어레이의 시뮬레이션이 끝나면 그림 12와 같이 배열변수 C와 원소의 인덱스를 정해주면 그 원소가 계산되어진 과정을 윈도우에 나열하여 준다. 심볼로 나열된 결과를 분석해봄으로써 쉽게 계산하여 정



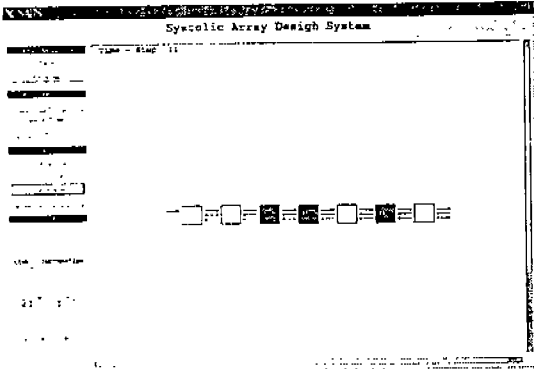
(그림 10) 시스틀릭 어레이의 수행과정(시간 스텝 4)
(Fig. 10) Execution step of the systolic array(time step 4)



(그림 11) 시스틀릭 어레이의 수행과정(시간 스텝 5)
(Fig. 11) Execution step of the systolic array(time step 5)



(그림 12) 수행 결과의 검증화면
(Fig. 12) Verification window of execution results



(그림 13) 행렬곱셈을 위한 1차 평면 시스톨릭 어레이
(Fig. 13) 1 Dimensional planar systolic array for matrix multiplication

해진 공간-시간 함수에 의해 설계된 평면 시스톨릭 어레이가 행렬곱셈 문제를 올바르게 수행시킴을 검증해 볼 수 있다.

한편, 사용자가 1차원 시스톨릭 어레이를 설계하고자 할 때 아래와 같은 공간 및 시간 함수 $S = [-1 \ -1 \ 1]$, $T = [2 \ 1 \ 2]$ 를 선택하면 그림 13과 같은 1차원 시스톨릭 어레이를 설계하고 그려주며, 2차원 시스톨릭 어레이의 경우와 마찬가지로 수행시키고 검증할 수 있다.

5. 결 론

본 논문에서는 정규 순환방정식으로 표현된 n 차원의 문제를 입력으로 받아 시스톨릭 어레이의 설계에 필요한 정보를 추출하고, 공간-시간함수 및 입력 배치함수를 계산하고 이를 기반으로 하여 1차원이나 2차원 어레이 레벨(array level)의 최적 평면 시스톨릭 어레이를 유도하였으며, 윈도우상에 그려주고, 또 입력자료를 스케줄링하여 정규 순환방정식이 시스톨릭 어레이상에서 수행되는 과정을 시각적으로 보여주는 소프트웨어 SADS를 설계하고 구현하였다. SADS 시스템은 Sun workstation의 X window에서 C 언어를 사용하여 구현하였다.

주어진 n 차원 문제의 정규 순환 방정식에서 얻어진 자료 종속 행렬과 평면 어레이의 연결링크로 구성 되어질 수 있는 모든 링크행렬에서 unimodular 변환으로 구한 기하학적으로 다른 링크행렬로 이루어진

diophantine 방정식으로 공간행렬을 구하였다. 이렇게 구한 공간행렬의 커널공간의 시간간격을 최소화 하는 시간함수로 만들어진 공간-시간 변환은 n 차원 문제로부터 체계적으로 1차원이나 2차원의 시스톨릭 어레이의 네트워크를 유도하였다. 입력되는 초기 자료의 위치를 유도된 어레이 네트워크 상에 배치하는 대수식을 만들어 SADS 시스템에 적용하였다. 공간-시간 변환과 입력 배치 함수를 기반으로 가능한 여러 공간행렬과 시간행렬을 계산하여 이들 중 최적인 것을 선택하여 입력 자료가 배치된 평면 시스톨릭 어레이를 설계하였다.

더욱이, 개발된 SADS 시스템은 유도된 시스톨릭 어레이의 PE 배치와 링크의 연결을 X window에 그려서 사용자에게 시각적으로 보여주며, 수행 도중 각 PE에 어떤 자료들이 입출력되는지, 또 각 PE 마다 병렬적으로 수행되는 계산과정이나 계산결과를 대화식으로 검증할 수 있도록 pop up 윈도우를 통하여 확인할 수 있게 하였다. 주어진 문제가 설계된 시스톨릭 어레이 상에서 수행되는 과정을 straight line 프로그램으로 확인할 수 있게 하였다. 개발된 시스템을 이용하여 행렬 곱셈이나 계산량이 많은 문제로부터 시스톨릭 어레이를 설계하고, 그것의 수행을 검증하였다.

주어진 문제의 크기가 크면 설계된 시스톨릭 어레이의 PE 개수가 증가하므로, 앞으로 분할방법을 SADS에 적용하여 문제의 크기가 아주 큰 문제로부터 작은 개수의 PE를 가지는 시스톨릭 어레이를 설계할 수 있는 시스템을 개발하는 것이 향후과제로 남아있다.

참 고 문 헌

- [1] Bradley, R. E. and Cappello, P. R., "The SDEF programming system," *Journal of parallel and distributed computing*, Vol. 7, pp. 201-231, 1989.
- [2] Fortes, J. A. B., and Wah, B. W., "Systolic Array-from Concept to Implementation," *Computer*, Vol. 20, No. 7, pp 12-17, 1987.
- [3] Hartenstein, R. W. and Lemmert, K. "A CHDL-Based CAD System for The Synthesis of Systolic Architecture," *International Conf. on Systolic Arrays*, pp. 105-115, 1989.

[4] Heller, D., "Motif Programming Manual," O'reilly & Associates, Inc, 1991.

[5] Kung, H. T. and Leiserson C. E., "Systolic Arrays(for VLSI)," *Sparse Matrix Proceedings SIAM*, pp. 256-282, 1978.

[6] Kung, S. Y., *VLSI Array Processors*, Prentice Hall, 1987.

[7] Kung, S. Y. and Jean, S. N., "A VLSI Array Compiler System(VACS) for Array Design," *VLSI Signal Processing III*, pp. 495-508, 1988.

[8] Lee, P. and Kedem, Z. M., "Synthesizing Linear Array Algorithms from Nested For Loop Algorithms," *IEEE Trans. on Computers*, Vol. 37, No. 12, pp. 1578-1598, 1988.

[9] Moldovan, D. I., "On the Analysis and Synthesis of VLSI algorithms," *Proc. of IEEE*, Vol. 71, pp. 113-120, 1983.

[10] Moldovan, D. I., "ADVIS: A Software package for the design of systolic arrays," *IEEE Trans. Computer-Aided Design*, CAD-6, pp. 33-40, 1987.

[11] Sedukhin, S. G. and Sedukhin, I. S., "Systematic Approach and Software Tool for Systolic Design," *Parallel Processing: CONPAR 94-VAPP VI*, pp. 172-183, 1994.

[12] 유기영, "최적의 평면 시스틀릭 어레이의 설계방법과 그 구현에 관한 연구," 한국과학재단 연구보고서, 1995.



유 기 영

1976년 경북대학교 수학교육과 이학사
 1978년 한국과학기술원 전산학과 공학석사
 1992년 Rensselaer Polytechnic Institute 이학박사
 1978년~현재 경북대학교 컴퓨터공학과 부교수

관심분야: 병렬 및 분산처리, 병렬 컴파일러, 어레이 프로세서 설계



이 성 우

1994년 경북대학교 컴퓨터공학과 졸업(학사)
 1996년 경북대학교 컴퓨터공학과 졸업(석사)
 1996년~현재 경북대학교 컴퓨터공학과 박사과정

관심분야: 어레이 프로세서 설계, 병렬 및 분산처리, 객체지향 프로그래밍



박 동 기

1994년 경북대학교 컴퓨터공학과 졸업(학사)
 1996년 경북대학교 컴퓨터공학과 졸업(석사)
 1996년~현재 태창기술연구소 연구원

관심분야: 병렬 및 분산처리, 소프트웨어 공학



김 윤 호

1983년 경북대학교 전자공학과 전산전공 공학사
 1993년 경북대학교 컴퓨터공학과 공학석사
 1995년 경북대학교 컴퓨터공학과 박사과정 수료
 1988년~1990년 삼성전자 연구원

1994년~현재 계명전문대학 사무자동화과 전임강사
 관심분야: 병렬처리, 배열 처리기 설계, 자연어 처리 등