

Systolic Array의 설계

경북대학교 유기영*

● 목	● 차
1. 서 론	3.3 평면 시스톨릭 어레이 설계방법
2. 문제 및 시스톨릭 어레이의 모델	4. 자동 설계 시스템
2.1 문제의 모델	4.1 설계 시스템의 기능
2.2 시스톨릭 어레이의 모델	4.2 평면 시스톨릭 어레이 시스템 SADS
3. 시스톨릭 어레이 설계방법	5. 향후 연구과제
3.1 시스톨릭 어레이의 설계란?	6. 결 론
3.2 설계방법의 종류	

1. 서 론

VLSI 기술의 급속한 발전과 빠른 계산에 대한 요구는 병렬 계산(parallel computation) 시스템의 발전에 많이 기여하였고 계속해서 관심의 대상이 되어왔다. 병렬 시스템들 중에서 특별한 관심을 끄는 것 중의 하나가 시스톨릭 어레이(systolic array)이다. Kung과 Leiserson [12]에 의해 처음으로 제안된 시스톨릭 어레이는 간단한 연산능력을 가진 처리요소(processing element, PE)들이 규칙적이고 국부적으로 연결된 특수 목적의 시스템이다[12, 13, 14, 28]. 자료들은 매우 규칙적인 형태로 처리요소들을 통해 전달되며, 연산결과는 통신링크(communication link)를 통하여 이웃한 처리요소로 전달된다.

많은 시스톨릭 어레이들이 신호처리, 영상처리 그리고 행렬문제등의 계산이 대부분인 계산 중심(compute bound)의 문제들을 효율적으로 VLSI로 구현하기 위해 설계되었다. 70년대 말

과 80년대 초에는 대부분 경험적인 방법으로 시스톨릭 어레이를 설계하였다[11, 14]. 그후 80년대 중반부터 많은 연구자들이 주어진 문제 으로부터 여러가지 공간-시간 사상(space-time mapping)의 대수적인 방법으로 체계적인 시스톨릭 어레이의 설계방법을 제안하기 시작하였다[1, 3, 5, 10, 15, 20] [21, 22, 24, 25, 26]. 대부분의 방법들은 루프(loop)의 깊이가 n 인 순환 방정식(recurrence equation, RE)에서 $(n-1)$ 차원의 어레이 공간으로 사상시켜 $(n-1)$ 차원의 시스톨릭 어레이를 설계하였다. 90년대 들어와서 Lee와 Kedem 등은 n 차원의 루프 알고리즘을 q , $(1 \leq q \leq n-1)$ 차원의 시스톨릭 어레이로 변환 시키는 방법을 제시하였다[17, 18, 27]. 주어진 n 차원의 문제로 부터 VLSI로 구현이 용이한 1차나 2차원의 평면 시스톨릭 어레이를 설계할 수 있는 공간-시간 변환과 입력자료의 배치합수를 구하는 대수적이고 체계적인 방법의 연구가 진행되었다[29]. 이 글에서는 계산 위주의 문제로부터 시스톨릭 어레이를 설계하는 문제와 여러 연구자들에 의해서 제안된 설계방법를 구분하여 설명하며, 지금까지 연구실

*중심회원

에서 연구하여온 평면 시스톨릭 어레이의 설계 방법을 설명하고자 한다. 주어진 문제로부터 만들어질 수 있는 시스톨릭 어레이의 개수는 아주 많으며, 설계될 시스톨릭 어레이를 그려서 그것의 계산 형태를 알아보는 것은 아주 복잡하다. 설계된 시스톨릭 어레이가 제대로 동작하는지를 검증하는 문제도 대단히 복잡하고 번거롭다. 따라서, 자동적으로 시스톨릭 어레이를 생성하고 그려주는 소프트웨어에 많은 관심을 가지고 여러 소프트웨어 시스템이 개발되었다 [2, 7, 8, 16, 23]. 이들 대부분 시스템은 전체적으로 일괄성 있게 만들어진 시스템이 아니고 여러가지 제약이 많은 소프트웨어이다. 또한 이들 시스템은 문제로부터 시스톨릭 어레이를 설계하는 과정중에서 부분만을 보여준다. 따라서, 고급 언어로 주어진 문제로부터 자동으로 시스톨릭 어레이를 설계해 주며, 사용자에게 친근감을 주며, 또한 GUI가 실현되어 사용하기가 편리한 시스톨릭 어레이 설계 소프트웨어 개발의 필요성에 따라 본 연구실에서는 시스톨릭 어레이 자동 설계 시스템을 개발하였다. 또한, 개발된 자동 설계시스템에 대해서 설명한다.

본 고에서 설명하는 시스톨릭 어레이의 설계를 위한 방법을 여러 다양한 문제에 적용하여 시스톨릭 어레이를 쉽고 편리하게 설계할 수 있을 것이다. 개발된 시스톨릭 어레이 설계 소프트웨어를 이용하여 n 차원 문제로부터 최적의 평면 시스톨릭 어레이를 설계할 뿐만 아니라, 설계된 시스톨릭 어레이의 수행을 검증할 수 있어서 이미 만들어진 시스톨릭 어레이의 수행을 검증할 수 있을 것이다. 또한, 시스톨릭 어레이의 설계는 처리기 어레이의 구조, 시스톨릭 알고리즘(systolic algorithm) 및 병렬 처리에 관한 연구를 촉진 시킬 것으로 기대된다. 특히, 설계 소프트웨어를 이용하여 여러가지 DSP 문제 등에 하드웨어의 해답을 줄 수 있는 array-level의 시스톨릭 어레이를 설계하여 실제로 여러 응용분야에 활용할 수 있을 것이다.

이 글의 구성은 다음과 같다. 제 2장에서는 먼저 문제의 모델과 시스톨릭 어레이의 모델을 설명한다. 제 3장에서는 문제로부터 시스톨릭 어레이를 설계하는 문제와 만족하여야 할 조건을 설명하며, 여러가지 설계 방법들을 구분하여

소개한다. 그리고 연구되어온 평면 시스톨릭 어레이의 설계방법을 소개한다. 제 4장에서는 소개한 우리의 이론적 설계방법에 바탕으로 실제로 GUI 환경하에서 시스톨릭 어레이를 설계하고 수행을 검증하는 소프트웨어 시스템을 설명한다. 마지막으로 제 5장에서는 결론을 맺는다.

2. 문제 및 시스톨릭 어레이의 모델

이 장에서는 시스톨릭 어레이의 설계 방법을 설명하기전, 주어진 문제의 특성과 그 문제의 알고리즘 모델 뿐만아니라 주어진 문제로부터 설계 방법에 의해서 유도될 시스톨릭 어레이의 모델을 설명한다.

2.1 문제의 모델

디지털 신호처리와 영상처리에 사용되는 많은 알고리즘들은 특수 목적의 어레이 프로세서(array processor) 설계에 요구되는 국소적 자료흐름과 병렬연산 및 파이프라이닝(pipelining) 등과 같은 특징을 가지고 있다. 이런 특성들을 가지는 알고리즘들은 근본적으로 단일 배정(single assignment)의 개념을 내포하고 있는 순환방정식(recurrence equation)으로 편리하고 간결하게 표현될 수 있다. 이런 특성을 가지고 있으면 어레이 프로세서로 구현하기가 쉬우면 바로 시스톨릭 어레이로 설계가 가능한 알고리즘을 시스톨릭 알고리즘(systolic algorithm)이라 한다.

신호나 영상처리 분야에 응용되는 알고리즘들을 하드웨어적으로 해결해 주는 시스톨릭 어레이의 설계에 관한 많은 연구에서는 주어진 문제를 표현하는 수단으로 순환방정식을 이용하였다[9, 17, 18, 20, 22, 25]. 더욱이 시스톨릭 어레이의 설계를 쉽게 하기위해 자료의 흐름이 정규적인 순환방정식에 많은 관심과 연구가 이루어졌다. 정규 순환방정식의 각 연산에 필요한 자료들의 흐름을 나타내주는 자료종속 관계와 자료의 초기위치를 나타내주는 계산 그래프(computation graph, CG)와 순환방정식의 변수들 사이의 종속 관계로 만들어지는 축소 종속 그래프(reduced dependency graph, RDG)를 이용하여 문제의 성질을 분석한다.

Z 는 정수의 집합이고, Z^n 는 n -차원 유클리드 공간의 정수 격자점 (lattice point), 인덱스점들의 집합이다. 모든 격자점 $\vec{p} \in Z^n$ 는 원소들이 정수인 n 차원 열 벡터이다. 관심의 대상이 되는 유한개의 격자점들의 집합 C^n 은 Z^n 의 부분집합이다. 주어진 알고리즘은 각 정수 격자점 $\vec{p} \in C^n$ 에 대해서 배열 변수(array variable) $v_1(\vec{p}), v_2(\vec{p}), \dots, v_n(\vec{p})$ 의 값을 m 개의 어떤 순환방정식들에 의해 계산한다. 따라서, 주어진 알고리즘은 계산이 수행되는 각 격자점들의 집합 C^n 과 계산몸체 (computation body)로 나누어진다. 각 격자점들의 집합 C^n 를 계산공간 (computation space)이라 한다. 주어진 알고리즘의 계산공간 C^n 은 각 격자점들이 만족해야 하는 조건으로 어떤 인덱스(index)들의 부등식으로, 또한 알고리즘의 계산몸체는 정규 순환방정식으로 표현된다.

[알고리즘 모델] 다음과 같이 계산공간의 조건과 순환방정식으로 표현된 계산 몸체로 구성된 알고리즘을 정규 종속알고리즘(uniform dependence algorithm, UDA)이라 한다.

$$l_1 \leq i_1 \leq u_1, l_2 \leq i_2 \leq u_2, \dots, l_n \leq i_n \leq u_n$$

$$v_i(\vec{p}) = f_i(v_1(\vec{p} - \vec{d}_{i_1}), v_2(\vec{p} - \vec{d}_{i_2}), \dots,$$

$$v_{i_k}(\vec{p} - \vec{d}_{i_k}))$$

여기서,

- 열 벡터 $\vec{p} \in C^n \subset Z^n$ 는 격자점을 나타내며 계산셀 (computation cell)이라 한다. 이들 격자점들의 집합인 계산공간은 다음과 같이 표현된다.

$$C^n = \{ \vec{p} = [i_1, i_2, \dots, i_n]^T \in Z^n \mid l_1 \leq i_1 \leq u_1, \dots, l_n \leq i_n \leq u_n \}$$

n 은 알고리즘의 차원이며, 또한 계산공간 C^n 의 차원이기도 하다.

- l_k 와 u_k 는 k 번째의 인덱스 i_k 의 하계(lower bound)와 상계(upper bound)이다. l_k 와 u_k 는 인덱스 i_1, i_2, \dots, i_{k-1} 들의 선형함수이다.
- $f_i, (1 \leq i \leq m)$ 는 점 $\vec{p} \in C^n$ 에서 수행되는 k -항 함수를 나타낸다. 여러개의 함수들이 순환방정식 계를 이룬다.
- $v_i(\vec{p})$ 는 점 \vec{p} 에서 수행된 결과 값을 의

미한다.

- $\vec{d}_i \in Z^n, (j = 1, \dots, k)$ 은 정수 열 벡터로 자료종속 벡터(자료 흐름 벡터)이다. 이들 자료종속 벡터로 구성된 행렬 $D = [\vec{d}_{i_1}, \dots, \vec{d}_{i_k}]$ 를 자료종속 행렬이라 부른다. 자료종속 벡터들은 계산셀 \vec{p} 의 위치에 상관없이 항상 상수 벡터이기 때문에 이동-불변(shift-invariant) 벡터라 한다.

시스톨릭 어레이로 설계하고자 하는 문제의 자료흐름 벡터가 이동-불변인지, 시스톨릭 어레이로 구현이 가능한지, 순환방정식에서 배열 변수들의 자료종속 관계를 자료종속 그래프를 이용하여 분석한다. 문제의 자료흐름 벡터가 상수가 아닐 때, 자료흐름 벡터를 상수화하는 방법에 대한 연구도 많이 이루어지고 있다. 시스톨릭 어레이로 구현 가능한 문제로부터 시스톨릭 어레이를 설계하는데 필요한 정보는 격자점들의 집합으로 계산공간, 각 계산 셀에서 수행되는 순환방정식의 몸체, 자료종속 행렬, 그리고 입력 배열변수의 초기위치등이다.

디지털 신호 처리(DSP)에 기본적인 convolution 알고리즘을 예로 생각해 보자. Convolution 알고리즘은 아래와 같은 정규 종속알고리즘으로 표현된다.

$$0 \leq i \leq N, 0 \leq j \leq K$$

$$W(i, j) = W(i-1, j)$$

$$X(i, j) = X(i-1, j-1)$$

$$Y(i, j) = Y(i, j-1) + W(i-1, j) X(i-1, j-1)$$

위의 convolution 알고리즘으로부터 시스톨릭 어레이를 설계하는데 필요한 정보인 계산셀의 집합 $C^2 = \{ [i, j]^T \mid 0 \leq i \leq N, 0 \leq j \leq K \}$ 이고, 변수 W, X, Y 에 대한 자료종속 벡터들은 각각 $\vec{d}_w = [1, 0]^T, \vec{d}_x = [1, 1]^T, \vec{d}_y = [0, 1]^T$ 이다.

2.2 시스톨릭 어레이의 모델

시스톨릭 어레이는 병렬 처리와 파이프라이닝을 이용해 계산의 효율성을 높이는 일종의 병렬 계산구조이다. 이것은 많은 수의 매우 간단하고 동일한 처리요소들이 규칙적이며, 지역적(local)(또는 가까이 인접하는)으로 연결된 것이다. 시스톨릭 어레이의 중요한 특징은 모듈

성(modularity), 규칙성, 지역적 연결, 고도의 파이프라이닝, 동기화된 고도의 다중처리 등이다. 특히, VLSI로 구현하기 쉽게 1차나 2차 시스틀릭 어레이의 설계에 대한 연구가 관심의 대상되었다. 이런 시스틀릭 어레이를 평면 시스틀릭 어레이라 한다. 본 고에서는 평면 시스틀릭 어레이의 모델에 대해서 설명한다. 시스틀릭 어레이의 정의가 많은데 [5,15,28], 여기서 평면 시스틀릭 어레이의 특징은 다음과 같이 요약하여 기술할 수 있다.

- (1) 동기성(synchrony) : 자료들의 계산은 주기적으로, 즉 global clock의 일정한 시간 간격동안에 수행되며 네트워크의 링크를 통해서 자료가 전달된다.
- (2) 규칙성(regularity) : 시스틀릭 어레이는 유사한 처리 구조를 가진 처리요소들로 구성되어 있다. 즉, 연결 형태나 처리요소의 계산은 경계 처리 요소들을 제외하고는 모두 같다. 시스틀릭 어레이는 무한히 확장 가능하다.
- (3) 공간적/시간적 국부성(spatial and temporal locality) : 네트워크의 연결은 근접한 연결들을 갖는다. 즉, 공간적 국부성(spatial locality)을 가진다. 네트워크의 연결을 통해 통신을 할 때에 적어도 하나의 단위 시간동안 지연(delay)이 있다. 이를 시간적 국부성(temporal locality)이라 한다.
- (4) 선형성(linearity) : 자료는 전체 수행이 이루어지는 동안 처리요소들의 네트워크에서 일정한 속도와 방향으로 흐른다.

- (5) 평면성(planarity) : 시스틀릭 어레이의 연결 네트워크는 좌표 평면상의 정수 집합인 격자 평면(lattice plane)에 내포되어야 한다.

시스틀릭 어레이의 처리요소들은 그들의 계산을 하나의 단위 시간동안에 수행한다고 가정한다. 각 처리요소는 입력링크들과 출력링크 그리고 레지스터들을 가진 기능부(functional unit)와 제어부(control unit)로 구성되어 있다. 기능부는 입력링크를 통해 들어오는 자료를 가지고 각 처리요소가 수행할 함수에 따라 출력을 계산한다. 제어부는 처리요소들의 계산을 동기화시키고 순서화함으로써 제어한다. 평면 시스틀릭 어레이의 처리요소의 상호연결 네트워크(interconnection network)은 Z^1 또는 Z^2 에 속하는 정수 격자점에 위치한 처리요소들이 서로 통신링크(communication link)로 연결된 네트워크이다. 그림 1과 같이 각 처리요소들은 링크에 의해 이웃한 처리요소와 연결되어 있다. 각 링크는 연결된 처리요소들 사이의 정수 차 벡터에 의해 표현될 수 있다. 각 링크에 대해서 모든 처리요소들은 같은 크기의 쉬프트 레지스터를 가지며 자료들은 그 링크를 통해서 전달된다. 쉬프트 레지스터들은 지연 버퍼로서 필요하며, 버퍼의 길이는 지연시간을 의미한다. 네트워크의 가장자리에 있는 경계 처리요소들은 외부 입력링크(external input link)와 외부 출력링크(external output link)를 가진다. 이차원의 평면 시스틀릭 어레이에서는 두 개의 연결링크가 교차되는 경우가 있는데 이를 배제한 시스틀릭 어레이의 설계가 좋다.

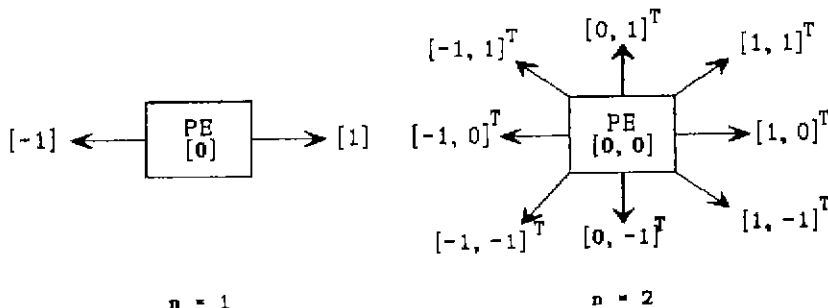


그림 1 처리요소의 가능한 연결링크

시스틀릭 어레이를 표현해주는 요소는 처리 요소가 위치하는 격자점의 집합인 프로세서 공간(processor domain) P , 처리요소가 수행할 함수 F , 연결 링크와 지연 시간(delay time), 그리고 외부 입/출력 링크(external input/output link)와 같은 네트워크에 대한 정보 N , 및 주어진 문제가 유도된 시스틀릭 어레이상에 정확하게 계산되기 위해 정확한 시간(right time)에 정확한 위치(right place)에 정확한 자료(right data)가 도달할수 있게 배치된 입력자료들의 위치 I 이다. 시스틀릭 어레이의 형식적인 모델은 4순서쌍(P, F, N, I)로 표현된다. 시스틀릭 어레이의 설계란 주어진 알고리즘 모델에서 어떻게 시스틀릭 어레이의 모델로 유도하는냐는 것이다.

3. 시스틀릭 어레이 설계방법

n 차원 문제로부터 일차나 이차의 시스틀릭 어레이를 설계하는 방법에 대해서 설명한다. 먼저, array-level의 시스틀릭 어레이를 설계하는 문제를 명시하며, 문제로부터 시스틀릭 어레이를 설계하는 방법들을 분류하여 설명하며, 방법 중에서 공간-시간함수에 의한 변환방법과 변환방법이 만족해야하는 조건을 설명한다.

3.1 시스틀릭 어레이의 설계란 ?

n 차원의 계산공간을 가진 문제로부터 시스틀릭 어레이를 합성할 수 있는 체계적인 방법이 시스틀릭 어레이의 설계 방법이다. 예를들어, 정규 순환방정식으로 표현된 행렬곱셈 문제로부터 그림 2와 같은 array-level의 시스틀릭 어레이를 유도하는 것을 시스틀릭 어레이의 설계라 한다.

시스틀릭 어레이의 설계방법은 문제의 각 계산점의 수행이 어느 PE에서 수행되는지를 계산해 주어야 하며, 또한 계산이 언제 어레이의 PE에서 수행되는지를 밝혀 주어야한다. 더욱이 알고리즘의 초기위치에 있는 입력자료의 값이 설계된 시스틀릭 어레이에서 어떻게 배치되어야 하는지를 명시해 주어야한다. 또한 설계방법은 유도될 시스틀릭 어레이의 네트워크 구조와 통신링크의 관한 정보도 명시해 주어야 한

다. 80년대 많은 연구자들에 의해서 주어진 문제로부터 어떻게 하면 그림 2와 같은 시스틀릭 어레이를 유도할 수 있는지에 대해서 여러 방법을 연구하였다.

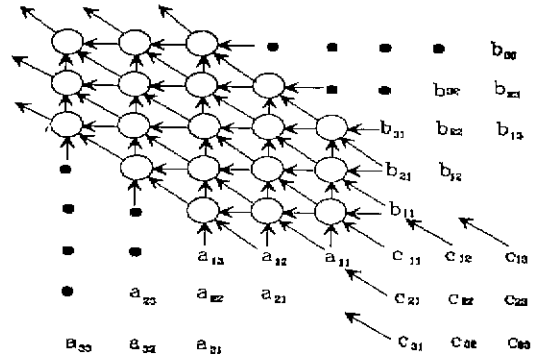


그림 2 3x3 행렬곱셈의 2차원 시스틀릭 어레이

3.2 설계방법의 종류

주어진 n 차원의 계산공간을 가진 문제로부터 시스틀릭 어레이를 합성할 수 있는 설계방법은 어떻게 시스틀릭 어레이로 유도하는가에 따라서 설계방법이 나누어진다. 70년대 말과 80년대 초에는 경험적인 방법 혹은 기하학적인 변환에 의한 방법으로 주어진 문제로부터 시스틀릭 어레이를 유도하였다. 그 후 많은 연구자들이 체계적이고 대수적으로 시스틀릭 어레이를 설계하는 방법을 제시하였다. 이들 방법들은 크게 매개변수(parameter)에 의한 방법과 공간-시간 변환에 의한 방법으로 나뉘어진다, 그리고 많은 설계방법중에서 n 차원의 문제에서 $(n-1)$ 차원의 시스틀릭 어레이를 설계하는 방법과 일 반적인 q , $(1 \leq q \leq n)$ 차원의 시스틀릭 어레이를 설계하는 방법으로 크게 분류할 수 있는데, 처음에는 전자의 방법에 대한 연구가 대부분이 었으나, 80년대 말부터 후자의 방법에 대한 연구가 많이 이루어졌다. Fortes 등은 여러가지 설계방법에 대해 요약하였다[6].

3.2.1 매개변수에 의한 방법

이 방법은 Li와 Wah[19]에 의해서 제시된 설계방법으로 시스틀릭 어레이는 자료흐름의

속도, 자료의 지역적 분포, 및 계산의 주기와 같은 3개의 매개변수로 성격 지어질 수 있으며, 이들 매개변수간에 관계는 제약방정식으로 표현되며, 이들 제약방정식의 최적해를 풀어서 이들 세 매개변수의 값이 결정되면 시스틀릭 어레이를 완전히 유도하게 된다.

첫번째 매개변수인 자료흐름의 속도(velocity)는 자료 x 가 한 시간주기 동안 지나가는 방향성 거리(directional distance)를 의미하는 벡터 \vec{x} 이다. 이 벡터 \vec{x} 의 크기는 i/j , ($i \leq j$, 정수)로 표현된다. 그것은 j 시간주기간 자료 x 가 i PE들을 지나 전파되었다는 것을 의미한다. 두번째 매개변수는 자료의 분포이다. 이것은 시스틀릭 어레이의 입력과 출력으로 사용되는 2차원 어레이상에서 자료 x 가 행과 열로 직선상에 어떻게 분포하는가를 의미하는 방향성 거리 벡터이다. 세번째는 주기(period)로 이는 2차원 어레이상에서 출력을 위해 반복적인 계산이 일어나는 시간의 차, 및 입력값이 접근되는 시간 차로 정의된다.

주어진 i 차원 순환방정식 $z_{i,j}^k = f[z_{i,j}^{k-1}, x(i, k), y(k, j)]$ 을 시스틀릭 어레이로 유도하려면 3개의 매개변수, 즉 자료흐름의 속도, 자료의 분포 및 계산주기로 만들어진 제약방정식을 주어진 순환방정식으로 부터 유도하며, 해를 구하여 시스틀릭 어레이를 만든다. 그 후, Dowling과 Taylor[4]에 의해서 매개변수 방법을 개선하여 주어진 루프(loop)로부터 행렬로 표현되는 시스틀릭 어레이 합성방정식을 유도하여 이 합성방정식을 대수적으로 풀어서 시스틀릭 어레이를 합성하였다. 매개변수에 의한 방법은 주어진 n 차원 문제를 $(n-1)$ 차원 시스틀릭 어레이로 설계하는데 사용하였다. 너무 제약이 많아서 일반적인 차원의 시스틀릭 어레이를 설계하기에는 어려운 방법이다.

3.2.2 공간-시간 변환에 의한 방법

n 차원 정규 종속알고리즘으로부터 k 차원 ($k < n$)의 시스틀릭 어레이를 공간-시간 변환에 의한 방법으로 합성한다는 것은 n 순서쌍으로 표시되는 계산셀의 정수 벡터 $\vec{p} = [p_1, \dots, p_n]^T$, (p_i 는 정수, $1 \leq i \leq n$)를 시스틀릭 어레이 공간상의 k 순서쌍으로 표시되는 처리요소의 좌표

인 정수 벡터 $\vec{s} = [s_1, \dots, s_k]^T$ ($k < n$)와 시간 스텝 t 로 사상시키는 것이다. 이것은 n 차원 문제 공간을 $(k+1)$ 차원 처리기-시간 공간으로 사상시키는 공간-시간 변환(space-time transformation)의 문제이며, 이는 선형 변환함수로 나타낼 수 있으며, 이것을 행렬로 표현할 수 있다. n 차원의 계산공간을 $(k+1)$ 차원의 처리기-시간 공간으로 변환시키는 선형 변환함수를 나타내는 행렬을 M 이라 한다면 M 은 다음처럼 표현될 수 있다.

$$M = \begin{bmatrix} S \\ \Pi \end{bmatrix}$$

S 는 $k \times n$ 정수 행렬, Π 는 $1 \times n$ 행 벡터이다. \vec{s}_i ($1 \leq i \leq n$)를 행렬 S 의 i 번째 열벡터, π_i 를 Π 의 i 번째 원소라 하자. 계산공간 C^n 안의 한 계산셀 $\vec{p} = [p_1, \dots, p_n]^T$ 가 주어지면,

$$M \vec{p} = \begin{bmatrix} S \\ \Pi \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} p_1 \vec{s}_1 + \dots + p_n \vec{s}_n \\ p_1 \pi_1 + \dots + p_n \pi_n \end{bmatrix} \\ = [s_1, \dots, s_k, l]^T$$

와 같다. 문제공간의 계산셀 $\vec{p} = [p_1, \dots, p_n]^T$ 의 계산은 공간행렬 S 에 의해서 시스틀릭 어레이 공간상의 k 순서쌍으로 표시되는 좌표 $\vec{s} = [s_1, \dots, s_k]^T$ ($k < n$)인 PE에서, 시간벡터 Π 에 의해서 시간 스텝 t 에서 수행된다. 그래서 S 를 공간 변환행렬이라 한다. 간단히 공간행렬(공간함수)이라 하고, Π 를 스케줄 벡터 혹은 시간벡터(시간함수)라 한다. 만약 $k = (n-1)$ 이면 공간-시간 변환방법으로 n 차원 문제이며 $(n-1)$ 차원 시스틀릭 어레이를 합성하는 것이다. 대부분의 설계방법은 이와같은 공간-시간 변환방법이다. 다음은 공간-시간 변환의 시간과 공간함수가 만족하여야할 조건을 설명한다.

자료 종속 벡터 \vec{d}_i 와 $\vec{q} - \vec{p} = \vec{d}_i$ 인 관계에 있는 계산셀 \vec{p} 와 \vec{q} 에 대해서, 시간 벡터 Π 는 다음의 부등식을 만족하여야 한다. 이 조건을 시간조건이라 한다.

$$\Pi \vec{q} - \Pi \vec{p} = \Pi(\vec{q} - \vec{p}) = \Pi \vec{d}_i > 0$$

또한, 공간행렬 S 와 시간벡터 Π 에 의해서 한 순간에 시스톨릭 어레이의 한 PE에서 서로 다른 계산셀 \vec{p} 와 \vec{q} 의 계산이 동시에 수행되게 변환할 수 없다. 이 조건을 공간-시간조건이라 한다. 이 조건은 다른 두 계산셀 \vec{p}, \vec{q} 에 대해서, $\Pi\vec{p} \neq \Pi\vec{q}$ 이거나 $S\vec{p} \neq S\vec{q}$ 이어야 한다. 시간조건과 공간-시간조건을 만족하는 공간행렬 S 와 시간벡터 Π 를 구하는 여러가지 방법에 따라 여러 설계방법이 연구되어 왔다.

대표적으로 Moldovan[21, 22], Miranker[20], Quinton[24], 및 Kung[15, 16]은 한 개의 투사벡터(projection vector)를 정하고 그 벡터를 따라 문제공간을 투사하여 시스톨릭 어레이의 프로세서 공간을 구할 수 있게 공간행렬을 정하고, 문제로부터 구해진 자료중속 행렬 D 에 대하여 정수 부등식 $\Pi D > 0$ 를 만족하며, 또한 공간-시간조건을 만족하는 시간벡터 Π 를 구하였다. 이들 방법에서는 정해진 투사벡터에 따라 공간행렬이 구해지므로 무한히 많은 공간행렬이 정해질 수 있고, 또한 1차원만 줄여서 $(n-1)$ 차원의 시스톨릭 어레이를 유도하였다. 시간조건과 공간-시간조건을 만족하는 시간벡터를 선형 계획법으로 풀어서 구하였다. 그 후 80년대 후반과 90년대 초에 Lee와 Kedem[17, 18], Rao[25, 26], Shang[27]에 의해서 n 차원 문제공간에서 q 차원 시스톨릭 어레이를 설계하는 공간-시간 변환방법을 제시하였다. 이들 방법에서도 역시 공간행렬은 임의로 선정하여 사용하였으며, 주로 시간벡터를 구하는 방법으로 시스톨릭 어레이의 설계 방법을 제시하였다. Lee와 Kedem은 Lamport의 초평면(hyperplane)과 보조 초평면으로 시간벡터를 구하였다. 그들은 시간벡터는 시간조건과 공간-시간조건 뿐만아니라 통신링크를 따라 흐르는 자료들이 충돌이 일어나지 않도록, 즉, 자료충돌 회피조건을 만족해야함을 보였다. 일반적으로 n 차원 문제에서 q 차원으로 변환할 때는 자료 충돌 문제가 발생함을 처음 언급하였다. Rao는 주어진 공간행렬과 적교가 되는 행렬을 구하고 그것을 이용하여 반복적인 방법으로 시간벡터를 구하였다. 그리고 Shang은 시간조건, 공간-시간조건, 및 자료충돌 회피조건을 만족하는 선형 계획 문제로 시간벡터를 구하였다. 이들 방법은

다중 투사로 일반 차원의 시스톨릭 어레이를 설계하는 방법을 제시하였다고 하지만 여전히 자동으로 공간행렬을 구하는 방법은 체계적으로 제시하지 못하였고, 또한 유도된 시스톨릭 어레이상에 입력 초기값의 배치를 구하는 대수적인 방법에 대해서 언급하지 않았다.

Zhong[30]은 주어진 문제가 n 차원일 때, 모든 연결링크(interconnection link)가 최근접 이웃 PE들과 연결되는 $(n-1)$ 차원 시스톨릭 어레이를 설계하는데 필요한 공간함수를 계산하는 대수적인 방법을 제시하였다. 이 방법은 행렬의 left-Hermit normal form을 이용하여 기하학적으로 다른 네트워크를 구하고 주어진 문제의 자료 중속 행렬이 기하학적으로 다른 네트워크로 사상시키는 공간행렬을 구하였다. 공간-시간 변환방법은 공간행렬과 시간벡터를 어떻게 구하였는가에 따라 여러 종류가 있으며, 많은 연구가들이 각자의 고유한 방법을 제시하였다.

본 연구실에서는 n 차원의 문제에서 VLSI 구현이 쉬운 1차원 혹은 2차원 평면 시스톨릭 어레이를 체계적으로 설계해주는 공간행렬을 자동으로 구하며, 구해진 공간행렬을 이용하여 시간벡터를 구하는 방법에 대해 연구하였다. 대부분의 설계방법에서는 입력값의 위치를 고려하지 않았다. n 차원 문제에서 $(n-1)$ 차원 시스톨릭 어레이를 합성할 때는 유도된 어레이상에서 입력값의 초기위치를 쉽게 결정할 수 있지만, 일반적인 q 차원 시스톨릭 어레이를 합성할 때는 그 결정이 어렵고 복잡하다. 문제의 입력자료의 초기위치로부터 공간행렬 및 시간벡터를 이용하여 입력자료가 위치하는 격자점을 명시해 주는 대수적인 입력자료 배치함수의 유도에 대해서 연구하였다. 다음 절은 이 방법에 대해서 간단히 설명하고자 한다.

3.3 평면 시스톨릭 어레이 설계방법

연구실에서는 n 차원의 인덱스 공간을 가진 문제로부터 1차원/2차원 평면 시스톨릭 어레이를 합성할 수 있는 체계적인 방법을 연구하였다. 설계될 시스톨릭 어레이는 VLSI 구현이 용이하게 그 차원을 1차 혹은 2차원으로 하며, 각 PE는 바로 이웃하는 PE와 상호연결되는 어레이

이를 설계하는 것이 연구의 목표이다. 일반적으로 시스틀릭 어레이의 설계는 세 함수 (1) 공간함수(행렬)(space function) (2) 시간함수(벡터)(time function) (3) 입력자료 배치함수(input layout function)로 이루어진다. 즉, 문제의 각 계산이 어레이의 어느 PE에서 수행되는지를 계산해 주는 공간함수(행렬), 문제의 각 계산이 언제 어레이의 PE에서 수행되는지를 계산해 주는 시간함수(벡터), 문제의 초기위치에 있는 입력자료가 시스틀릭 어레이에서 어떻게 배치되어야 하는지를 명시해주는 입력자료 배치함수이다. 공간함수는 행렬로 표현되므로 공간함수와 공간행렬은 같은 의미로 사용되며, 마찬가지로 시간함수는 벡터로 표현되기 때문에 시간함수와 시간벡터라는 용어를 같이 사용한다.

공간-시간 변환방법에서는 공간함수에 의해서 문제의 자료종속 벡터는 시스틀릭 어레이의 연결링크로 사상된다. 또한 계산셀들이 수행되는 어레이상에서의 PE 위치도 결정된다. 연결링크와 PE 위치는 어레이의 하드웨어를 결정하게 된다. 따라서 공간행렬의 계산은 하드웨어와 밀접하게 관련되어 있다. 또 공간행렬 S 가 구해지면 S 와 함께 시간조건이나 공간-시간조건을 만족하는 시간벡터 Π 는 반드시 존재하지만 시간벡터 S 가 먼저 주어졌을 경우에 공간-시간조건을 만족하는 공간행렬 S 가 반드시 존재하는 것은 아니다[18]. 따라서 공간-시간 변환방법에서 공간행렬을 먼저 구하는 것이 바람직하다.

3.3.1 공간함수(행렬)의 계산

S 를 공간함수를 나타내는 $k \times n$ 공간행렬이라 하자. 모든 자료종속 벡터 \vec{d}_i 와 $\vec{q} - \vec{p} = \vec{d}_i$ 인 계산셀 \vec{p} 와 \vec{q} 에 대해서 곱 $S \vec{d}_i$ 는 PE들 사이 즉, 구해진 시스틀릭 어레이에서 $S \vec{p}$ 와 $S \vec{q}$ 사이의 연결링크를 정의한다. 각 자료종속 벡터 \vec{d}_i 는 S 에 의해서 PE $S \vec{p}$ 로부터 PE $S \vec{q}$ 로 자료가 이동해 가는 연결링크 $c_i \vec{l}_i$, (c_i 는 양의 정수)로 사상된다. 여기서 각 PE는 최근점 PE와 상호접속망을 이룬다면 상수 c_i 는 1이고, 연결링크 \vec{l}_i 는 그림 1에서 보여준 평면 시스틀릭 어레이의 가능한 연결링크의 집합에 속해야 한다.

문제의 $n \times m$ ($m \geq n$) 자료종속 행렬을 $D = [\vec{d}_1, \dots, \vec{d}_m]$ 라고 하자. D 의 각 열은 자료종속 벡터이고, $S \vec{d}_i = \vec{l}_i$ 인 연결링크 \vec{l}_i 이 열인 행렬 $L = [\vec{l}_1, \dots, \vec{l}_m]$ 는 $k \times m$ 행렬로 연결링크 행렬(간단히 링크행렬)이라 한다. 행렬 L 은 시스틀릭 어레이의 상호접속망(interconnection network)를 나타낸다. 행렬 D 는 문제에서 주어지고, 설계될 1차 혹은 2차원의 시스틀릭 어레이의 상호접속망을 나타내주는 행렬 L 만 주어지면 공간행렬 S 는 diophantine 방정식 $SD = L$ 의 해이다. 구해질 공간행렬은 문제의 각 자료종속벡터를 그림 1에서 보여준 허용되는 연결링크의 집합중 한 요소로 대응시켜야 한다. 이런 조건을 공간조건이라 한다. 링크행렬 L 를 미리 정하는 것이 공간행렬 S 를 구하는데 아주 중요하다. 만약 1차원 혹은 2차원 시스틀릭 어레이상에서 만들어질 수 있는 기하학적으로 다른 모든 네트워크를 미리 알 수 있다면 공간행렬은 보다 쉽게 구할 수 있을 것이다. 이미 설명한대로 공간행렬 S 는 n 개의 미지수를 가지는 m , ($m \geq n$)개의 방정식 $L = SD$ 의 해이다.

자료종속 행렬 D 와 링크행렬 L 를 각각 나누어서 $D = [D' D'']$, $L = [L' L'']$ 라 하자. 여기서 D' 은 $n \times n$, D'' 는 $n \times (m-n)$, L' 는 $k \times n$, L'' 는 $k \times (m-n)$ 행렬이다. 그러면 $L' = SD'$ 이며, $L'' = SD''$ 이다. L' 은 L 의 $k \times n$ 부행렬이므로 그러한 행렬 L' 의 수는 $|\Delta_k|^n$ 이다. 여기서 $|\Delta_k|$ 는 k 차원 시스틀릭 어레이의 가능한 연결링크의 집합 Δ_k 의 연결링크의 수이다. 모든 L' 행렬을 나열할 수 있다면 각각의 L' 행렬에 대해서 방정식 $L' = SD'$ 이 존재한다. D' 이 nonsingular 정방행렬이기 때문에 방정식은 완전한 해를 구할 수 있다. 시스틀릭 어레이의 네트워크를 표현해주는 행렬 L' 중에서 기하학적으로 같은 것을 없애고 네트워크의 위상이 다른 것만 구하면 그 수가 많지 않다. L 은 순서화된 집합이므로 L 의 열 벡터들(즉, 연결벡터)이 자리바꿈을 하면 다른 시스틀릭 어레이를 나타내는 다른 네트워크의 기하학적 구조가 된다. 행렬 L_1 과 L_2 을 행렬 L' 중에 임의의 두 행렬이라 할 때, $L_2 = UL_1$ 되는 unimodular 행렬 U 가 존재하면 링크행렬 L_1 과 L_2 로 표현되는 두 시스틀릭 어레이는 기하학적으로 같다. 이는 네트워크

의 기하학적 구조 L_2 를 갖는 평면 시스톨릭 어레이는 네트워크의 기하학적 구조 L_1 의 평면 시스톨릭 어레이를 회전시킨 결과이므로 L_1 과 L_2 의 네트워크 구조를 갖는 두 시스톨릭 어레이는 기하학적으로 같다란 의미이다. Unimodular 관계는 동치관계이므로 링크행렬들의 집합의 동치류는 중요하다. 같은 동치류에 있는 링크행렬은 기하학적으로 구조가 같고 다른 동치류에 있는 링크행렬은 구조가 다르므로 공간행렬 S 을 찾기 위해선 각 동치류의 대표 행렬 L' 을 고려하는 것만으로 충분함을 알 수 있다. 링크행렬중에서 두 행렬이 unimodular 관계에 있는지를 알아야된다. 두 링크행렬의 Smith normal form을 이용하여 unimodular 관계가 있는지를 알 수 있다[29]. n 이 문제의 차원이고, k 가 설계될 시스톨릭 어레이의 차원일 때 g_n^k 는 각 동치류의 대표 링크행렬의 집합이라 하면 그 집합의 원소의 개수 즉, 생성될 수 있는 기하학적 구조로 다른 네트워크의 개수는 다음 표 1과 같다.

표 1 기하학적으로 다른 네트워크의 수

g_n^k	k	
	1	2
n	1	2
2	2	-
3	4	26
4	8	180
5	16	1160

주어진 자료중속 행렬을 시스톨릭 어레이의 미리 계산된 네트워크 구조로 사상시키는 공간행렬 S 는 두 방정식 $L' = SD'$ 와 $L'' = SD''$ 로 표현된다. 미리 생성된 연결 링크행렬 L' 에 대해서 우선 식 $L' = SD'$ 의 유일한 해 S 를 찾는다. 다음으로 식 $L'' = SD''$ 에 의해서 D'' 에 대응하는 나머지 링크행렬 L'' 을 계산한다. L'' 중의 연결링크들이 Δ_k 에 속하고 $L = [L' \ L'']$ 이 교차링크를 가지지 않으면 S 는 공간조건을 만족하는 공간행렬이다.

3.3.2 시간함수(벡터)의 계산

시간함수(벡터) Π 는 n 차원 문제의 각 계산셀에 대해서 시스톨릭 어레이상에서 동작할 특

정한 시간을 지정한다. 공간행렬 S 가 주어졌을 때, 시간조건과 공간-시간조건을 만족하는 시간함수(벡터)를 구하는 문제를 설명한다.

공간행렬에 의한 공간사상의 과정은 문제의 각 계산셀들을 투사공간(projection space)을 따라 처리기 공간으로 투사하는 것으로 생각할 수 있다. $\kappa(S)$ 를 공간행렬 S 의 커널공간(kernel space)이라 하자. 즉, $\kappa(S) = \{ \vec{x} \mid S \vec{x} = \vec{0}, \vec{x} \in \mathbb{Z}^n \}$. 여기서, $\vec{0}$ 은 k -열 null 벡터이다. 이 커널공간을 $n-k$ 차원의 투사공간이라 한다. 계산셀 \vec{p} 에 대해서 집합 $\kappa(S)_{\vec{p}} = \{ \vec{q} \in \mathbb{C}^n \mid \vec{q} - \vec{p} \in \kappa(S) \}$ 라 정의할 때, $\kappa(S)_{\vec{p}}$ 는 투사공간 $\kappa(S)$ 의 방향에서 기준점 \vec{p} 를 지나는 격자점들의 집합이다. 이 집합을 동일 공간집합(affine space set)이라 한다. $\kappa(S)_{\vec{p}}$ 에 속하는 격자점 \vec{q} 에 대해서 $S(\vec{q} - \vec{p}) = \vec{0}$ 이다. 이 사실은 $\kappa(S)_{\vec{p}}$ 에 속하는 모든 격자점들은 시스톨릭 어레이상에서 같은 처리요소에 사상됨을 나타낸다. 따라서, 동일 공간집합 $\kappa(S)_{\vec{p}}$ 의 각 격자점에서 서로 다른 유일한 순차시간을 정하는 벡터 Π 가 시간조건을 만족하면 시간벡터가 된다. 왜냐하면 동일 공간집합에 있는 두 계산셀 \vec{p} 과 \vec{q} 는 시스톨릭 어레이에서 동시에 같은 처리요소에 사상되지만 벡터 Π 에 의해서 $\Pi \vec{p} \neq \Pi \vec{q}$ 이므로 공간-시간조건을 만족하기 때문이다.

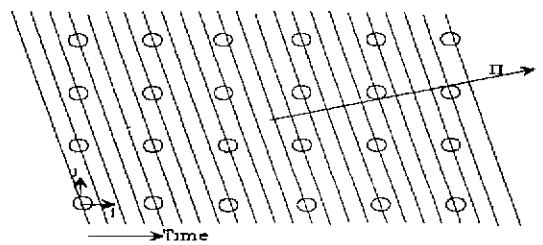


그림 3 시간벡터 Π 의 정의

그림 3에서 원의 임의의 공간집합의 격자점이라 할 때, 시간조건을 만족하며, 각 격자점의 시간을 유일하게 하는 벡터 Π 를 계산하면 시간벡터를 구하게 된다. 그런데 그런 시간벡터는 굉장히 많이 존재할 수 있다. 따라서, 각 격자점의 시간간격을 최적화하는 최적의 시간벡터 Π

를 계산하는 것이 의미가 있다.

공간행렬 S 가 주어졌을 때 $(n-k)$ 차원의 투사공간 $\kappa(S)$ 를 결정하는 것은 쉽다. 커널공간 $\kappa(S)$ 의 방향 베이스벡터의 수는 $(n-k)$ 이며, 베이스벡터를 $\vec{u}_1, \dots, \vec{u}_{(n-k)}$ 하자. 시스톨릭 어레이의 한 처리요소 PE_{s_p} 에 사상되는 동일 공간집합의 원소중에서 계산셀들의 집합을 $\kappa(S)_p$ 에 포함되는 $(n-k)$ 차원의 평행면체(parallelepiped)라 하며 $box_p \{\vec{u}_1, \dots, \vec{u}_{(n-k)}\}$ 로 표기한다. box_p 의 차원이 $n-k=2$ 일 때는 box_p 는 평행사변형이 되며, 차원이 k 일 때는 box_p 는 k 차원의 평행면체가 된다. 그림 3은 임의의 2개의 베이스벡터로 구성된 평행면체 $box_p \{\vec{u}_1, \vec{u}_2\}$ 를 나타낸다. 주어진 공간행렬 S 는 box_p 안에 있는 격자점들을 같은 처리요소에 사상한다. 그러므로 시간벡터를 구하는 문제는 box_p 안에 속하는 격자점들을 순차순서로 나열하며, 그 차가 최소인 벡터를 찾는 문제가 최적의 시간벡터를 구한다. box_p 안에 속하는 격자점들의 시간차를 최소화하는 문제는 다음과 같이 정의된 파이프라이닝 간격 α 를 최소화하는 Π 를 구하면 된다.

$$\alpha = \frac{\Pi(\vec{p}^* - \vec{p}^*) + 1}{|box_p \{\vec{u}_1, \dots, \vec{u}_{n-k}\}|}$$

여기서, \vec{p}^* 와 \vec{p}^* 는 box_p 안에 있는 계산점들의 계산 순서상에서 양극점을 의미한다. 위에 정의된 파이프라이닝 간격은 한 처리요소에서 연속된 두 계산 사이의 평균 시간 간격을 의미한다.

시간벡터 Π 를 계산하는 대략적인 방법은 먼저 $k \times n$ 의 공간행렬 S 의 투사공간인 커널공간 $\kappa(S)$ 을 구한다. 구해진 투사공간의 방향 베이스벡터를 $\vec{u}_1, \dots, \vec{u}_{(n-k)}$ 를 구한다. 그리고 계산한 방향 베이스벡터에 대해서 모든 순열(permutation)을 계산한다. 그 이유는 가능한 시간벡터를 모두 계산하기 위해서이다. 다음 단계는 하나의 순열을 선택한다. 그리고 선택된 순열을 구성하고 있는 방향 베이스벡터에 대해서 차례로 시간조건을 만족하며, 시간간격이 최소이고 연결링크의 지연시간이 최소인 시간벡터를 계산한다. 이 단계를 모든 순열의 경우에 대해서

반복 수행 후 그 중에서 α 를 최소로 하는 시간벡터를 구한다. 이와같이 구한 시간함수는 공간함수와 더불어 n 차원 문제를 $k, k \leq (n-2)$ 차원의 시스톨릭 어레이로 사상할 때, 자료 충돌 회피 조건이 만족됨을 알 수 있다[29].

3.3.3. 입력자료 배치함수의 계산

입력 자료들의 정확한 배치는 시스톨릭 어레이(systolic array)의 정확한 동작을 보장하기 위하여 반드시 기술되어야 할 사항이다. 왜냐하면 시스톨릭 어레이가 올바르게 설계되고 최적의 공간과 시간의 복잡도를 가지는 경우라 할지라도, 입력자료가 정확하게 주어지지 않으면 그 시스톨릭 어레이의 정확한 동작을 기대할 수 없기 때문이다. 시스톨릭 어레이의 처리요소들이 올바른 계산을 하기 위해서는, 각 처리요소들의 계산에 필요한 자료들이 모두 올바른 시간에 입력되어야 한다. 이러한 입력 자료에는 다른 처리요소들에서의 계산 결과인 자료들과 시스톨릭 어레이 외부에서 주어지는 자료들의 두 종류가 있다. 이 중에서 다른 처리요소들에서의 계산 결과인 자료들을 처리요소들과 이들로 구성되는 어레이가 정확하게 설계되었다면 이들 자료들은 제대로 흘러가게 된다. 따라서, 문제가 되는 것은 시스톨릭 어레이의 외부에서 주어지는 자료들이 어떻게 초기에 배치되어야 할 것인가 하는 것이다.

어떠한 문제로부터 시스톨릭 어레이를 유도하기 위해서 앞 절에서 설명한 공간-시간 변환 방법을 이용할 경우에는 입력 자료들이 어레이 외부뿐만이 아니라 내부에도 사상될 수가 있다. 이러한 경우에는 자료의 입력이 어레이의 경계(boundary)에 있는 처리요소들에서만 외부 입력이 일어나는 것이 아니라 어레이 내부에 있는 처리요소들에서도 외부 입력이 일어난다. 이러한 경우에는 시스톨릭 어레이의 내부에 있는 처리요소들도 외부 입/출력 기능을 가져야 하므로, VLSI 구현상 어레이의 설계 복잡도가 증가되고, 어레이 내부에 사상되는 자료들은 시스톨릭 어레이의 실행을 시작하기에 앞서선적재를 해주어야 한다. 이러한 문제들은, 외부 입력이 경계 처리요소들에서만 일어나는 것으로 전제하고, 어레이 내부에 사상되는 입력 자료들은

어레이의 외부로 후퇴(retreat)시킴으로써 해결하는 것이 가능하다[29].

시스톨릭 어레이상에서 입력 자료의 위치를 구하는 방법은 먼저 입력자료의 위치는 계산 그래프의 입력자료들의 초기위치에 공간과 시간함수를 이용하여 시스톨릭 어레이상에서의 그 초기위치를 구한다. 그리고 만일 프로세서 어레이 내부로 진입한 입력 자료가 있으면 경계 처리요소를 기준으로 어레이 외부로 후퇴시킴으로써 재배치(rescheduling)하여 구한다. 입력자료는 각 어레이 변수들에 대해서 별개로 다루어진다.

4. 자동 설계 시스템

시스톨릭 어레이 설계 방법에 있어서 초기에는 대부분 경험적인 방법으로 설계하였으며, 그 후 정규 순환방정식(uniform recurrence equation: URE)으로 표현된 알고리즘에 대해서 공간-시간 사상(space-time mapping)과 같은 대수적인 방법을 도입하여 보다 체계적으로 시스톨릭 어레이를 설계하였다. 1980년대 후반과 90년대 초반부터 컴퓨터 프로그래밍을 사용하여 시스톨릭 어레이의 설계를 간단하게 자동적으로 만드는 시스템이 개발되었다[2, 7, 16, 23]. 그 예로 Moldovan의 ADVIS는 공간-시간 함수를 이용하여 알고리즘을 시스톨릭 어레이로 사상하는 방법론을 제시하였고[23], S. Y. Kung의 VACS는 주어진 단일 배정코드의 종속 그래프(dependence graph)와 어떤 프로젝션 벡터에 의해 구해진 시스톨릭 어레이를 Sun Workstation상에 시각적으로 표현하였다[16]. 특히 Bradley의 SDEF 시스템은 공간-시간 함수를 통해 시스톨릭 어레이를 설계하며 추적하고, 컴퓨터 프로그래밍 차원에서 시스톨릭 프로그래밍의 제기를 주장하였다[2]. 그러나, 이들 대부분의 시스템은 일련의 시스톨릭 어레이의 설계 과정에 이용하기에 미흡하고, 더구나 설계된 어레이의 수행과정을 검증할 수 없거나, 검증하기에 불편한 시스템이다. 따라서, 고급언어로 표현된 주어진 문제에 대해 공간-시간 함수를 계산하여 시스톨릭 어레이를 자동적으로 설계하고, 시뮬레이션하며, 그 문제가 설계된 시스

톨릭 어레이에서 올바르게 수행되는지 검증을 해주는 시스템이 요구되어진다.

4.1 설계 시스템의 기능

일반적인 시스톨릭 어레이 설계 시스템이 가져야할 기능은 다음과 같이 생각할 수 있다. 먼저 주어진 문제를 어떻게 표현하여 시스템에 입력하며, 이를 시스톨릭 어레이를 설계하는데 필요한 정보, 즉 계산점의 범위, 계산 몸체, 자료 종속 벡터들 및 초기 입력값의 위치등에 관한 정보를 추출하는 전처리 단계를 고려하여야 한다. 대화식으로 또는 그래픽으로 시스톨릭 어레이를 직접 입력하는 시스템도 유용할 때가 있지만, 고급 프로그래밍 언어와 같은 문법에 따라 주어진 문제를 표현하면 사용자가 쉽게 주어진 문제를 서술하여 시스템에 입력할 수 있다. 입력된 문제의 표현을 구문분석하고 시스톨릭 어레이를 설계하는데 필요한 정보를 추출하기 위해 컴파일 기술을 이용하기가 쉽다.

다음으로 전처리 단계에서 추출된 정보를 가지고 앞절에서 설명한 공간-시간 변환함수를 자동으로, 더욱이 계산되어질 수 있는 여러 함수 중에서 주어진 최적조건을 만족하는 공간-시간 함수와 입력자료 배치함수를 계산하여 설계될 시스톨릭 어레이를 구하는 설계단계가 있어야 된다. 설계정보로부터 주어진 문제의 자료 종속 벡터를 설계될 시스톨릭 어레이의 연결링크에 사상시키는 공간함수를 구하고, 공간함수를 바탕으로 지연시간을 최소화, PE의 활용률을 극대화, 혹은 계산시간이 최소화와 같은 최적조건을 만족하며 동시에 시간조건, 공간-시간조건, 및 자료충돌 회피조건을 만족하는 시간함수를 구한다. 마지막으로 설계된 시스톨릭 어레이 네트워크상에 입력값의 위치를 결정해 주는 입력자료 배치함수를 구하여야 한다. 이들 함수를 구할 때, 사용자와 대화식으로 쉽게 구할 수 있게 시스템이 만들어져야 한다.

시스톨릭 어레이 설계 시스템은 공간-시간 변환함수와 입력자료 배치함수를 구하고 단지 시스톨릭 어레이를 그려주기만 할 수도 있다. 그러나 설계된 시스톨릭 어레이를 윈도우에 그려줄 뿐만 아니라 매 시각마다 시스톨릭 어레이의 수행을 시뮬레이션하여 시스톨릭 어레이

가 수행되는 과정을 추적할 수 있는 검증 기능을 가져야 한다. 특히, 각 PE가 동적으로 수행하는 형태를 쉽게 시각적으로 확인할 수 있도록 하여야 하며, 수행 중에 임의의 PE 내부에 어떤 자료들이 계산되는지를 윈도우를 통해 볼 수 있게 하고, 원한다면 그 계산의 계산과정 (computation history)을 사용자에게 보여 주어 수행을 검증하도록 GUI 기능을 살려서 구현되어야 한다.

4.2 평면 시스톨릭 어레이 시스템 SADS

본 연구실에서는 주어진 정규 순환방정식으로부터 공간-시간 함수를 이용하여, 시스톨릭 어레이의 PE 배치와 링크의 연결성과 임출력 타이밍을 계산하여, 평면 시스톨릭 어레이의 배치를 SDT-200에서 제공하는 X-Window에 그려서 사용자에게 시각적으로 보여 줄 수 있는 소프트웨어, SADS(Systolic Array Design System)를 개발하였다. 이 시스템은 설계된 시스톨릭 어레이의 각 PE들이 병렬적으로 수행되는 과정을 사용자가 볼 수 있게 하였으며, 그 수행 도중에 각 PE에 어떤 자료들이 임출력, 계산되는지를 팝업 윈도우를 통하여 보여준다. 또 출력값이 어떤 경로를 통해 계산되었는지 계산과정 (computation history)을 팝업 윈도우를 통해 시각적으로 보여줌으로써, 사용자는 설계된 시스톨릭 어레이상에서 주어진 문제가 올바르게 수행되는지 검증할 수 있게 해준다. 개발된 시스템 SADS는 정규 순환방정식으로 표현되는 DSP나 여러 행렬문제로부터 평면 시스톨릭 어레이를 설계할 수 있다.

첫단계는 정규 순환방정식으로 표현된 문제를 구문분석 (parsing)을 하여 평면 시스톨릭 어레이를 설계하는데 필요한 정보를 구하는 입력 전처리이다. 두번째 단계는 전단계에서 구해진 여러가지 정보를 가지고, 공간-시간 변환 및 입력자료 배치함수를 계산하여 알고리즘으로부터 최적의 평면 시스톨릭 어레이를 설계하는 단계이다. 마지막으로 세번째 단계는 설계된 시스톨릭 어레이를 시각적으로 보기 쉽게 X-Window상에 그리고, 시스톨릭 어레이의 수행이 올바른가를 검증하기 위해 시뮬레이션한다.

개발된 SADS 시스템의 수행을 보여주기 위

한 예로 행렬의 곱셈연산으로부터 2차원 평면 시스톨릭 어레이를 설계하고 시뮬레이션을 하는 과정중 한 시간 스텝의 수행결과를 보여주고자 한다.

4.2.1. 행렬곱셈의 전처리

SADS 시스템의 전처리 기능에서는 편집기 ("vi")를 이용하여 행렬곱셈 문제를 다음과 같이 정규 순환방정식으로 표현한 입력 원시화일을 만든다.

```

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

다음으로 입력화일을 선택하고, 구문분석하여 시스톨릭 어레이를 설계하는데 필요한 설계정보, 즉 계산점의 집합과 계산문제, 자료흐름 벡터 및 입력자료의 초기위치를 구하여 설계정보화일을 만든다.

4.2.2 2차원 시스톨릭 어레이 설계의 예

전단계에서 만들어진 설계정보가 들어있는 화일을 선택하고, 공간-시간 변환방법으로 시스톨릭 어레이를 설계하려면 윈도우의 명령버튼을 누른다. 설계하고자하는 시스톨릭 어레이의 차원을 2차로 하기위해 팝업 윈도우의 선택버튼을 누르면 자동으로 2차원 시스톨릭 어레이를 유도해주는 공간 및 시간함수가 계산된다. 모든 공간, 시간함수가 계산된 후는 사용자는 윈도우에 표시되는 공간-시간함수중에서 주어진 최적조건을 만족하는 함수를 정할 수 있다. 예로써 아래와 같은 공간함수(S)와 시간함수(Π)를 정하였다고 하자.

$$S = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad \Pi = [1, 1, 1]$$

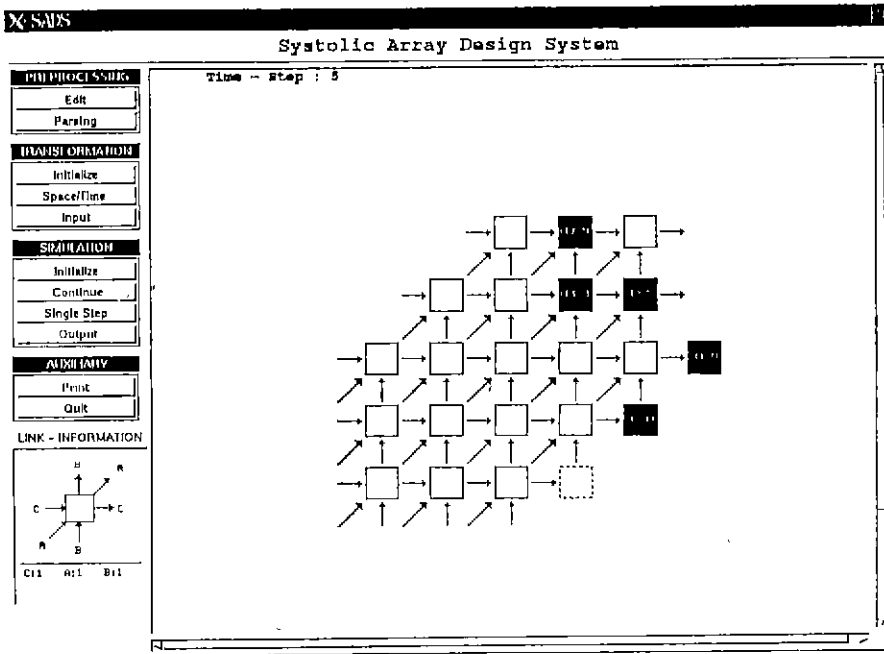


그림 4 곱셈문제의 시스톨릭어레이의 수행 예

마지막으로 설계된 시스톨릭 어레이의 수행을 시뮬레이션하기 위해 시뮬레이션을 수행하는 명령버튼을 눌러서 시스톨릭 어레이에 대한 정보가 저장되어 있는 화일을 선택하면 시스톨릭 어레이의 배치와 링크 모양이 주 윈도우상에 그려지며, 또 왼쪽 하단부의 작은 윈도우에는 각 변수에 대한 PE의 링크 모양이 그려진다. 왼쪽 하단부의 작은 윈도우안의 PE의 링크 방향을 보면 알 수 있듯이, 변수 A의 연결링크는 $[1, 1]^T$ 이며, 변수 B의 값은 아래에서 위로 흐르며 $[0, 1]^T$, 변수 C의 값은 왼쪽에서 오른쪽으로 흐른다($[1, 1]^T$). 변수 A, B는 단지 사용되는(수정되지 않는) 값으로 출력 값이 필요 없기 때문에, 주 윈도우에 그려진 전체 시스톨릭 어레이의 각 경계 PE에서 출력값을 위한 링크는 표시하지 않았다. 그러나 변수 C는 매 스텝마다 수정되며, 출력되는 값이므로 경계 PE에도 링크를 표시하였다.

다음에 매 시각마다 이산적으로 설계된 시스톨릭 어레이를 수행시키기 위해 수행 명령버튼을 누르면 시간 스텝이 0일 때부터 계산이 모두 끝날 때까지 수행을 한다. 따라서 시스톨릭 어

레이가 바르게 수행되는지를 추적할 수 있다. 그림 4는 시간 스텝이 5일 때 시스톨릭 어레이의 수행결과를 보여준다. 배열변수의 어느 원소가 어느 PE에서 계산되었는지를 볼 수 있다.

SADS 시스템은 자동으로 1차원 시스톨릭 어레이도 설계하고 검증할 수 있다.

5. 향후 연구과제

여기서는 시스톨릭 어레이의 설계분야에 앞으로 연구해야할 과제를 간단히 소개하고자 한다. 먼저 이제까지 설명한 공간-시간 변환방법 중 최적의 시간함수를 구하는 알고리즘을 개선하는 문제를 연구할 가치가 있다. 그 다음은 주어진 문제의 자료흐름 분석에 관한 연구이다. 대부분의 시스톨릭 어레이의 설계방법은 자료흐름이 아주 규칙적이고 정규적인 문제로부터 시스톨릭 어레이를 설계하였다. 그러나 많은 문제들은 자료의 흐름이 정규적이지만 불규칙적인 것으로 보이는 문제가 있다. 이런 경우 정규화(uniformization) 기법으로 자료의 흐름을 규칙적인 형태로 변환시킨 후 시스톨릭 어레이

를 설계하여야 한다. 이와같이 문제의 자료흐름 분석 및 정규화 기법에 대한 연구가 필요하다.

또 다른 한 관심분야는 결함허용(fault-tolerant) 시스틀릭 어레이 시스템의 설계이다. 하드웨어적이나 소프트웨어적으로 결함을 감지할 수 있고, 결함이 있어도 정확하게 수행할 수 있는 프로세서 어레이를 설계하는 방법이다.

마지막으로 관심분야는 분할(partition)이다. 주어진 문제로부터 설계된 시스틀릭 어레이의 PE의 개수는 주어진 문제의 크기에 의존하기 때문에 이런 시스틀릭 어레이를 문제-크기 시스틀릭 어레이라 한다. 어레이를 설계하고자 하는 문제의 크기가 아주 클 때 시스틀릭 어레이의 PE 개수가 많은 큰 시스틀릭 어레이가 유도되어서 VLSI로 구현하는데 어려움이 있다. 따라서 문제의 크기에 의존하지 않고 주어진 고정-크기의 시스틀릭 어레이상에서 크기가 아주 큰 문제를 수행되도록 분할하여 시스틀릭 어레이를 설계하는 문제가 아주 중요하다. 여러 가지 DSP 문제는 문제의 크기가 아주 클 때 때문에 이런 DSP 문제에 대한 시스틀릭 어레이를 설계하고자 할 때는 분할 기법이 꼭 필요하다. 분할 기법은 구현한 시스틀릭 어레이 설계 소프트웨어의 개발이 또한 연구과제이다. 본 연구실에서는 분할 기법에 대한 이론적 연구와 SADS에 구현 시키는 연구를 진행하고 있다. 앞으로 분할 기법을 적용하여 고정-크기 시스틀릭 어레이의 설계가 가능하게 되면 문제의 크기가 아주 큰 실제 응용문제에도 활용될 것이다.

6. 결 론

주어진 문제로부터 시스틀릭 어레이를 설계하는 방법에 대하여 설명하였다. 먼저 문제의 모델과 시스틀릭 어레이의 정의 및 모델에 대하여 소개하였고, 문제로부터 시스틀릭 어레이를 합성하는 방법 중에서 매개변수에 의한 설계 방법과 공간-시간 변환에 의한 설계 방법을 소개하였다. 본 연구실에서 연구하여 온 평면 시스틀릭 어레이의 설계 방법에서는 문제의 차원과 설계하고자 하는 시스틀릭 어레이의 차원이 정해지면 그 차원에서 만들어질 수 있는 모든 어레이

의 네트워크들 중에서 unimodular 동치관계를 이용하여 기하학적으로 서로 다른 어레이 네트워크를 미리 생성한다. 그리고 생성된 네트워크에 자료중속 행렬을 사상시키는 공간함수를 구한다. 구해진 공간함수에 의해서 동일 공간집합을 구하고, 이에 속하는 모든 계산점의 수행시간을 최소로 하여 PE의 활용률을 극대화하는 시간함수를 구한다. 마지막으로 시스틀릭 어레이상에서 입력값의 위치를 정해주는 입력자료 배치함수를 유도한다.

연구된 방법에 근거하여 사용자에게 친근감을 주는 MENU-based 시스템을 실현시키기 위해 X 윈도우상에서 n 차원의 문제로부터 최적조건을 만족하는 완전한 array-level의 평면 시스틀릭 어레이를 설계하고, 설계된 어레이를 시뮬레이션으로 검증할 수 있는 시스틀릭 어레이 설계 소프트웨어 시스템 SADS를 소개하였다. 시스템 SADS를 이용하여 여러 가지 DSP 문제 등에 하드웨어의 해답을 줄 수 있는 array-level의 시스틀릭 어레이를 설계하는데 활용할 수 있을 것으로 기대된다.

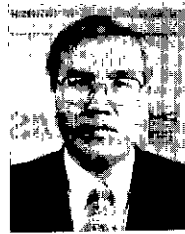
참고문헌

- [1] Augustine, F. and Varadarajan, R. "Efficient Mapping for Multidimensional systolic Arrays using Flexible buffer Structures," *International Conf. on Parallel Processing*, pp. 224-227, 1992.
- [2] Bradley, R. E. and Cappello, P. R., "The SDEF programming system," *Journal of parallel and distributed computing*, Vol. 7, pp. 201-231, 1989.
- [3] Chen, M. C., "Synthesizing Systolic Designs," *Proc. of 2nd International Symp. on VLSI Technology, Systems, and Applications*, pp 209-215, May, 1985.
- [4] Dowling, E. M, and Taylor, F. J., "Matrix Methods for the Design and Analysis of Recurrent Algorithms for Muti-Purpose Systolic Arrays." *IEEE international Conf. on ASSP*, pp. 2409-2412, May, 1989.
- [5] Fortes, J. A. B., and Wah, B. W., "Systolic Array-from Concept to Implementation,"

- Computer*, Vol. 20, No. 7, pp 12-17, 1987.
- [6] Fortes, J. A. B., and Fu, K. S., and Wah, B. W., "Systemetic Approaches to the Design of Algorithmically Specified Systolic Arrays," *Proc. of International Conf. on Parallel Processing*, pp. 300-303, 1985.
- [7] 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.
- [8] Heller, D., "Motif Programming Manual," *O'reilly & Associates, Inc*, 1991.
- [9] Karp, R. M., Miller, R. E., and Winograd S., "The organization of computations for uniform recurrence equations," *Journal of ACM*, Vol. 14, No. 3, pp. 563-590, 1967.
- [10] Kothari, S. C., Gannett, E. and Oh, H., "Optimal Designs of Linear Flow Systolic Architectures," *Proc. of International conf. on Parallel Processing*, pp. I-247-I-256, 1989.
- [11] Kuhn, R. H., "Transforming algorithms for single-stage and VLSI architectures," *Workshop on Interconnection Networks for Parallel and Distributed processing*, pp. 11-19, 1980.
- [12] Kung, H. T. and Leiserson C. E., "Systolic Arrays(for VLSI)," *Sparse Matrix Proceedings SIAM*, pp. 256-282, 1978.
- [13] Kung, H. T., "Why systolic architectures?," *IEEE Computer*, Vol. 15, No. 1, pp. 37-48, 1982.
- [14] Kung, H. T., "Let's Design Algorithms for VLSI Systems," *CALTECH Conf. on VLSI*, pp. 65-90, 1979.
- [15] Kung, S. Y., *VLSI Array Processors*, Prentice-Hall, 1987.
- [16] Kung, S. Y. and Jean, S. N., "A VLSI Array Compiler System(VACS) for Array Design," *VLSI Signal Processing III*, pp. 495-508, 1988.
- [17] 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.
- [18] Lee, P. and Kedem, Z. M., "Mapping Nested Loop algorithms into MultiDimensional Systolic Arrays," *IEEE Trans. on parallel and distributed systems*, Vol. 1, No. 1, pp. 64-76, 1990.
- [19] Li, G. J. and Wah, B. W., "The Design of Optimal Systolic Arrays," *IEEE Trans. on Computers*, Vol. c-34, No. 1, pp. 66-77, January, 1985.
- [20] Miranker, W. L. and Winkler, A., "Spacetime representations of computational structures," *Computing*, Vol. 32, pp. 93-114, 1984.
- [21] Moldovan, D. I., "On the Analysis and Synthesis of VLSI algorithms," *IEEE Trans. on Computers*, C-31, pp. 1121-1126, 1982.
- [22] Moldovan, D. I., "On the design of algorithms for VLSI systolic arrays," *Proc. of The IEEE*, Vol. 71, pp. 113-120, 1983.
- [23] Moldovan, D. I., "ADVIS: A Software package for the design of systolic arrays," *IEEE Trans. Computer-Aided Design, CAD-6*, pp. 33-40, 1987.
- [24] Quinton, P., "Automatic synthesis of systolic arrays from Uniform Recurrent Equations," *Proc. of 11th Annual Symp. on Computer Architecture*, pp. 208-214, 1984.
- [25] Rao, S. K., "Regular Iterative Algorithm and Their Implementations on Processor Arrays," *Ph. D. thesis*, Stanford University, 1985.
- [26] Rao, S. K., and Kaiath, T. "What is a Systolic Algorithm?," *SPIE Vol. 614 Highly Parallel Signal Processing Architectures*, pp. 34-48, 1986.
- [27] Shang, W. and Fortes, J. A. B., "On Time Mapping of Uniform Dependence Algorithms into Lower Dimensional Processor Arrays" *IEEE Trans. on Parallel and Distributed Systems*, Vol. 3, No. 3, pp. 350-363, 1992.
- [28] Ulman, J. D., *Computational Aspects of VLSI*, Computer Science Press, 1984.
- [29] Yoo, K. Y. "A Systolic Array Design Methodology for Sequential Loop Algorithms," *PH. D. thesis*, Rensselaer Poly-

technic Institute, 1992.

- [30] Zhong, X., Wong, I. and Rajopadhye, S. V., "Bounds on the number of linear allocation function," *VLSI Signal Processing IV, IEEE Press*, pp. 85-94, 1990.



유 기 영

1976년 경북대학교 수학교육학과 졸업(이학사)
 1978년 한국과학기술원 전산학과 졸업(공학석사)
 1992년 미국 Rensselaer Polytechnic Institute 졸업(이학박사)
 1978년~현재 경북대학교 컴퓨터공학과에 재직.
 관심분야: 병렬처리, DSP array processor 설계, 병렬 컴파일러 등임

● 제 6회 정보산업인 친선 골프대회 ●

- 주 최: 정록회 (情緣會)
- 주 관: 월간 경영과컴퓨터
- 일 시: 1995년 8월 26일(토) 오전 06:04 TEE OFF
- 장 소: 아시아나C.C. (36홀)
- 참가비: 12만원(그린피, 캐디피 및 카트료 등 포함)
단, 아시아나C.C. 회원권 소지자는 7만원
- 참가자격: 정보산업 분야에 종사하는 학자 및 고급경영관리자로서 초청자에 한함 (만 40세 이상)
- 참가인원: 96명
- 참가문의: 월간 경영과컴퓨터
골프대회 담당자 김주은 Tel: 333-4110
- 신청마감: 1995년 8월 14일(월)