

論文93-30B-7-1

VLSI 지향적인 APP용 2-D SYSTOLIC ARRAY PROCESSOR 설계에 관한 연구

(A Study on VLSI-Oriented 2-D Systolic Array Processor Design for APP (Algebraic Path Problem))

李顯洙*, 方晶姬*

(Lee Hyun Soo and Bang Jung Hee)

要約

본 논문에서는 기존의 전용 어레이 프로세서가 지니고 있는 문제점, 즉 문제에 대한 적용성 (flexibility) 이 부족함을 분석한다. 문제의 해를 구하는 접근방식이 유사한 APP(Algebraic Path Problem) 알고리즘중에서 대표적인 3 가지 알고리즘인 최단경로문제, 추이종속문제, 최소목 문제에 대해서 공통의 해를 구할 수 있는 새로운 APP 병렬 알고리즘을 제안하였다. 제안된 APP 병렬 알고리즘은 구조적인 개선이나, 기능상의 제약없이 다량의 데이터에 대해서 실시간 처리가 가능 할 수 있도록 하였다. 또한 2차원 Bit-Parallel 삼각 시스토크 어레이 프로세서로 구성하고 어레이 프로세서 내부의 1- PE 에 대한 세부설계를 수행하였다. 이에 대한 평가 방식으로는 Bit-연산처리 방식에 따른 복잡도 문제를 고찰 하여 전체적인 칩(chip)크기와 수행시간과의 관계를 기술하였다. 따라서 본 연구에서 설계한 프로세서는 실시간으로 입력되는 대량의 데이터에 대해서 $3N-4$ 의 수행시간 및 $O(N)$ 의 복잡도를 얻을 수 있었으며, 1- PE 내부의 총 게이트수 산출로 $O(N^2)$ 의 하드웨어 복잡도가 소요됨을 알 수 있었다.

Abstract

In this paper, the problems of the conventional special-purpose array processor such as the deficiency of flexibility have been investigated. Then, a new modified methodology has been suggested and applied to obtain the common solution of the three typical APP algorithms like SP(Shortest Path), TC(Transitive Closure), and MST(Minimum Spanning Tree) among the various APP algorithms using the similar method to obtain the solution. In the newly proposed APP parallel algorithm, real-time Processing is possible, without the structural enhancement and the functional restriction. In addition, we design 2-dimensional bit-parallel low-triangular systolic array processor and the 1-PE in detail. For its evaluation, we consider its computational complexity according to bit-processing method and describe relationship of total chip size and execution time. Therefore, the proposed processor obtains, on which a large data inputs in real-time, $3n-4$ execution time which is optimal $o(n)$ time complexity, $o(n^2)$ space complexity which is the number of total gate and pipeline period rate is one.

*正會員, 慶熙大學校 電子計算工學科
(Dept. of Comp. Eng., Kyunghee Univ.)

接受日字: 1993年 3月 3日

I. 서론

기존의 본 노이만에서의 순차처리방식에 의존한 시스템에서는 사용자들의 다양한 요구와 대량 데이터의 실시간 처리에 적합치 못한 제약사항을 가져옴에 따라 1980 년대를 전후로 순차처리 방식을 지양한 병렬 처리에 대한 연구가 활발해져 왔다.^[1] 또한 VLSI (Very Large Scale Integration) 기술에 따른 급격한 반도체 집적기술의 향상으로 고속화 처리의 시도와 임의의 문제에 대응하는 VLSI 시스템 구현에 대한 연구가 요구되고 있다. 따라서 이에 대응하는 병렬 처리 알고리즘을 기반으로 하는 VLSI 지향형 아키텍처에 대한 연구가 주목을 받게 되었다.^[2] 대표적으로 병렬 모델인 시스토크 어레이 프로세서^[3] [4]는 기존의 Uniprocessor 가 갖는 단점인 데이터의 연속적인 흐름과 결정성 이론에 의한 제어방식을 탈피하고, 처리기를 다수개의 Processing Element (PE) 로 분할하여 이들을 규칙적으로 배열함으로써 동기적인 제어에 의하여 병행적으로 데이터가 처리될 수 있는 어레이 프로세서로의 발전을 꾀하였다. 이러한 어레이 프로세서의 내부적 특성은 상호 인접한 내부 PE들간의 연결이 국부적이고, 규칙적인 인접한 배열간의 접속으로 데이터 상호간의 전달시간을 줄일 수 있으며, 또한 파이프라인 처리와 병렬 처리를 이용하여 성능을 극대화시킬 수 있는 고성능 아키텍처로써의 유효성을 지닌다. 일반적으로, 한가지 문제만에 국한시켜 고속으로 처리하는 전용 어레이 프로세서 설계에 대한 연구는 VLSI 컴퓨터 아키텍처의 주류를 차지하고 있으나, 이와같은 전용 어레이 프로세서의 성능을 더욱 극대화 하기 위해서 대상의 문제가 갖고있는 반복적이고, 규칙성을 지닐 수 있는 응용대상에 관한 연구가 계속되고 있으며, 이를 뒷받침 할 수 있는 병렬 처리 알고리즘에 관한 연구가 중요시된다.^[4] 현재 협의의 의미에서볼때, 전용 어레이 프로세서 (Special-purpose array processor)라 함은 하나의 문제만을 전담해서 수행할 수 있는 프로세서를 의미하는 것으로, 특수목적을 위하여 설계된 전용 어레이 프로세서로서의 특징을 갖는다. 그러나, 큰 단점으로서 구조적인 변경없이 다른문제에 적용될 수 없는 구조상의 제한성과 문제에 대한 적용범위가 한정 되어있다는 점이 존재한다. 다시말해, 응용 대상에 대해서 융통성 및, 적응성이 부족하여 데이터크기의 변화 내지는 다른 문제로의 접근시 재설계 하지 않으면 그 정확한 해를 구할 수 없는 제약사항을 갖는다.

따라서, 본 논문에서는 구조적인 개선 및 변형을 피하고 병렬알고리즘을 최대도 이용하여, 문제에 대한

융통성을 부여할 수 있도록 알고리즘의 병렬성을 고려하여 반복적인 특성을 갖고있는 APP 알고리즘 중 대표적인 3-문제 즉, SHORTEST PATH PROBLEM, TRANSITIVE CLOSURE PROBLEM, MINIMUM SPANNING TREE 에 대해서 공통의 해를 얻을 수 있는 새로운 APP 병렬알고리즘을 제안한다. 이것을 바탕으로 전용 어레이 프로세서의 단점인 융통성 결여, 기능의 확장성 제한에 따른 구조적 변화문제에 유연히 대처할 수 있는 적응성 (flexibility) 있는 다기능 1-PE를 세부 설계하고 고속화를 꾀하는 2차원 하삼각 시스토크 어레이 프로세서를 설계하였다. 이에, 본 논문에서는 특수목적을 위하여 설계된 전용 어레이 프로세서의 장점인 병렬처리를 최대한 이용하고, 제한사항을 최대한 극복할 수 있는 새로운 APP 병렬 알고리즘을 제안하여, 기존에 발표된 APP 알고리즘 중 하나의 기능만을 위한 전용 어레이구조와 각각 3 가지 알고리즘에 대해 비교한 결과, PE 수가 감소하고, 수행시간이 단축되는 결과를 얻을 수 있었다. 또한 응용대상으로한 APP 문제들에 대해 범용성을 부여하며, 구조적인 개선없이 APP 에 속하는 다른 문제에 대해서도 적용가능할 수 있는 PE를 설계함으로써, 제안한 APP 용 병렬 알고리즘의 병렬성을 토대로 3 가지 기능을 수행하는 프로세서를 설계하였다.

II. APP 알고리즘과 그 응용

1. APP 의 수학적인 배경

APP (Algebraic Path Problem)는 Shortest Path Problem, Transitive Closure Problem, Minimum Spanning Tree, Matrix-inversion Problem, Gauss-Jordan Elimination, Enumeration of Regular Expressions 등과 같이 그 문제의 특성을 행렬로 표현 할 수 있을 때, 인접한 행렬값을 기본으로 산술식 또는 논리식을 이용하여 원하는 해를 구할 수 있음을 의미한다.^[5] [6] 따라서 그래프 이론을 바탕으로 Edge 와 Vertex를 기본으로한 Path들간의 관계를 대수적 계산을 이용하여 해를 구하는 문제들의 집합을 의미한다.^[5] 위에서와 같이 문제의 특성이 그래프로 표현 되어질 수 있을 때, 이는 그래프 $G=(V, E, W)$ 로 정의할 수 있다. (V : 정점간의 집합, E : Edge간의 집합, W : 가중치 행렬 또는 인접행렬) 이때, 대상으로 하는 문제들의 특성과 알고리즘에 따라 다음에서 제시되는 $W(E, W(P), d_{ij})$ 의 값들을 이용하여 결과값을 계산한다.^[4]

(1) $W(E)$: 집합 $S = \{H, +, *, 0, 1\}$ 에서 H는 해당되는 문제를 의미하며 +, *는 해당되는 문제의 연산을 의미하며 0, 1은 각각의 연산에 대한 항등 원을 의미한다.

\Rightarrow THE EDGE WEIGHT (변의 가중치)

(2) $W(P)$: PATH P 에서 $W(E)$ 들의 곱으로써 정의된다.

$$W(p) = \prod_{E \in p} W(E)$$

\Rightarrow THE WEIGHT OF A DIRECTED PATH (유방향 그래프의 가중치)

여기서 APP는 모든정점의 D 값을 계산하는 것으로 다음과 같이 정의된다.

$$d_{ij} = +p \text{ is a path from } v_i \text{ to } v_j W(p)$$

따라서 APP의 연산은 각각 주어진 알고리즘의 기본연산과 문제의 특성에 따른 $W(E), W(P), d_{ij}$ 를 계산하는 것으로 각각의 알고리즘에 따라서 연산의 형태는 달라질 수 있다.^[6]

즉, 연산을 수행하기 위하여 N 개의 정점을 갖는 그래프에서 주어진 데이터 값을 N

N 행렬표현시 $A = [a_{ij}]$ 로 주어질 수 있다. 이때 처음에 주어진 행렬값을 A 라 하고, 연산결과후 출력되는 행렬값을 A+ 라 할때 $W(P)$ 들간의 행렬값을 $D = d_{ij}$ 로 정의시, $D = \{d_{ij}\}$ 를 반복계산하여 처음 반복연산을 $A=C_0$ 라 할때 $A=C_0$ 에서 출발하여, $C_n = d_{ij}$ 가 되는 $A = C_n$ 를 구하면 이때의 $A = [a_{ij}]$ 값이 원하는 해 A+ 가 된다.

2. 대표적인 APP 알고리즘 예

1) SHORTEST PATH PROBLEM

정의: 그래프 $G = (V, W, E)$ 가 주어질때 모든 정점간의 길이 중에서 가장 짧은 d_{ij} 의 길이를 찾는 것으로, Edge들의 Weight 값인 $S1, W(E), W(P), D, A = \{a_{ij}\}$ 는 다음과 같이 정의된다.^[6]

$$\begin{aligned} S1 &= \{H1, MIN, +, \infty\}, H1 = \{[0, \infty)\} \\ W(E) &= a_{ij} \\ W(P) &= W(e) \\ d_{ij} &= MIN p \text{ is a path from } V_i \text{ to } V_j W(p) \\ a_{ij} &= \begin{cases} W(V_i, V_j) & \text{if } (V_i, V_j) \in E \\ \infty & \text{otherwise} \end{cases} \end{aligned}$$

2) TRANSITIVE CLOSURE PROBLEM

정의: 그래프 $G = (V, E, W)$ 에서 추이종속성을 찾는 d_{ij} 를 찾는 것으로 Edge 들의 Weight 값인 $S2, W(E), W(P), D, A = \{a_{ij}\}$ 는 다음과 같이 정의 된다.^[6]

$$\begin{aligned} S2 &= \{H2, \dots, 1, 0\} \\ W(E) &= a_{ij} \\ W(P) &= AND W(e) \\ d_{ij} &= OR p \text{ is a path from } V_i \text{ to } V_j W(p) \\ a_{ij} &= \begin{cases} 1 & \text{if } (V_i, V_j) \in E \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

3) MINIMUM SPANNING TREE

정의 : 그래프 $G = (V, E, W)$ 에서 Edge간의 Weight 합이 최소인 SPANNING TREE d_{ij} 를 찾는 것으로 Edge들의 Weight 값인 $S3, W(E), W(P), D, A = \{a_{ij}\}$ 는 다음과 같이 정의된다.^[6]

$$\begin{aligned} S3 &= \{H3, MIN, +, 1, 0\} \\ W(E) &= a_{ij} \\ W(P) &= W(e) \\ d_{ij} &= MIN p \text{ is a path from } V_i \text{ to } V_j W(p) \\ a_{ij} &= \begin{cases} W(V_i, V_j) & \text{if } (V_i, V_j) \in E \\ \infty & \text{otherwise} \end{cases} \end{aligned}$$

III. 다기능을 위한 APP병렬알고리즘 제안

1. 설계사상

구조적이고 모듈성이 강한 VLSI 어레이 프로세서를 설계하기 위해서는 반드시 알고리즘의 INDEX SET 들의 변형과 데이터 의존 VECTOR들을 기본으로 한 알고리즘의 변형이 필요하다.^[6] 즉, 대상으로 삼는 문제의 병렬성을 최대한 이용할 수 있도록 병렬 알고리즘의 기본성격인 데이터의 내부연산이 반복적이고 순환적인 계산을 수행할 수 있도록 알고리즘의 변수들을 VECTOR화 해야 한다. 또, 알고리즘 수행동안 모든 변수는 오직 하나의 값만을 할당받을 수 있도록 Single assignment code 로의 변환이 중요하며, 알고리즘의 연산형태는 규칙적이며, 구조적인 성격을 갖는 반복식으로 표현될 수 있어야 국부적으로 반복적인 병렬계산을 할 수 있다.^[4] 이와같이 병렬성을 최대로 표현한 알고리즘일수록 어레이 프로세서의 병렬처리를 극대화할 수 있는 특성을 갖고있다. 따라서 알고리즘의 변형에 따라 어레이 프로세서의 설계 모형 또한 달라질 수 있으며, 그에 따른 성능 또한 달라질 수 있다.

이에 본 논문에서는 APP의 기본 알고리즘과

WARSHALL & FLOID^[4]가 제시한 알고리즘을 바탕으로 SHORTEST PATH PROBLEM, TRANSITIVE CLOSURE PROBLEM, MINIMUM SPANNING TREE를 대상으로 이 3개의 알고리즘이 갖는 공통적인 성격인 병렬성, 반복성, 규칙성을 추출하여 전용 어레이 프로세서에 기능적인 면에서의 융통성을 부여할 수 있도록 새로운 APP 병렬 알고리즘을 제안한다.

2. 제안한 APP병렬 알고리즘

```

FOR I=1 TO K DO(I=J)
  i   i-1  i-1  i-1      (1)
  Cii = @ Cii # (Aii , Bii)

FOR J=1 TO K DO (I≠J)
  i   i-1  i-1  i-1      (2)
  Cij = @ Cij # (Aii , Bij)
  i   i-1  i-1  i-1
  Cj = @ Cj # (Aj , Bii)

FOR H=1 TO K DO (H=G≠I)
FOR G=1 TO K DO      (3)
  i   i-1  i-1  i-1
  Chg = @ Chg # (Ahi , Big)

END

END FOR
    
```

제안된 알고리즘의 DATA흐름 및 DATA의존관계는 PE(i, j)로 표현될 수 있다.

$$\begin{matrix} PE(i, j) \\ PE(j, i) \end{matrix} = \begin{bmatrix} PE(i, j-(n-3)) \# PE(i, j) \\ PE(i, j-(n-2)) \# PE(i+1, j) \\ PE(i, j-(n-1)) \# PE(j+2, j) \\ PE(i, j-n) \# PE(j+3, j) \end{bmatrix}$$

제안된 알고리즘은 SHORTEST PATH PROBLEM, TRANSITIVE CLOSURE PROBLEM, MINIMUM SPANNING TREE의 알고리즘 특성과 APP 알고리즘 특성에 따라서 표1과 같이 연산자를 정의한다.

표 1. 연산자 기호

Table 1. Representation of operator.

문제 연산자	SHORTEST PATH PROBLEM	TRANSITIVE CLOSURE	MINIMUM SPANNING TREE
0	최소값(MIN)	OR (∨)	최소값(MIN)
#	가산 (+)	AND (∧)	최대값(MAX)

IV. 시스토크 어레이 프로세서 설계

1. 설계사상

제안된 알고리즘을 시스토크 어레이 프로세서로 설계하기 위하여 제안한 APP 병렬 알고리즘의 병렬성을 고찰하면 PE(i, j)는 PE(i-1, j), PE(i, j-1)의 결과값에 영향을 받으며 PE(i+1, j+1)의 데이터 값에 영향을 준다. K 방향으로 진행되는 사선상의 각 노드들은 동시에 처리된다. 병렬성 고찰을 기본으로 어레이 프로세서의 설계는 Canonical mapping 방법^[4]에 준한다.

따라서, 본 논문에서는 대상으로하는 문제에서 데이터 상호간의 의존성이 적고, 동시성이 높으며, 반복처리와, 병렬처리를 구사할 수 있는 알고리즘을 제안하여 1-PE 내 3가지 기능을 구사할 수 있는 시스토크 어레이 프로세서를 설계하였다. 또한 그래프 이론 중 3가지 문제가 갖는 무방향 그래프 특성 중 서로 대칭값이 같은 점을 이용, 하삼각 시스토크 어레이 프로세서를 설계함으로써 특수목적의 전용 어레이 프로세서가 갖는 구조상의 기능을 보완한다.

2. 시스토크 어레이 프로세서 구성

제안된 APP 병렬 알고리즘의 수행을 위한 2차원 Bit-parallel 시스토크 어레이 프로세서의 설계는 DG(Dependence Graph)^[4]에 기초를 두며, APP 중 대표적인 3-알고리즘을 1-PE 내에서 구조적인 변경 없이 문제에 대응할 수 있는 PE를 설계함으로써 이를 기초로 다양한 문제에 대응할 수 있는 하삼각 시스토크 어레이 프로세서를 설계한다.

시스토크 어레이 프로세서의 PE 갯수는 데이터 수에 의존하여 증가하고, 전체 처리시간은 데이터 수에 크게 의존된다는 사실을 볼때, 다소 PE 수를 줄이고 대량의 입력 데이터 N에 대해서 접근가능 할 수 있는 하삼각구조를 제시한다. 또한 시공간상 일정한 시간 T를 주기로 데이터의 연속적인 처리가 행해질 수 있도록 2차원 구조를 제시하여 Projection 방향 i, j에 따라서 데이터 A, B 값은 인접한 PE를 따라

파이프라인 처리를 수행하여 동일한 K 인덱스에 걸쳐있는 PE 들은 동시처리를 가능케 함으로써 1차원 구조의 처리속도를 향상시킬 수 있도록 한다. 이때, K 인덱스에 따른 데이터 의존 관계는 그림 1 과 같다.

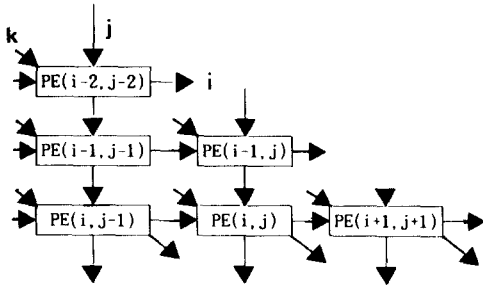


그림 1. 하삼각 구조의 데이터 의존관계
Fig. 1. Data dependence relation of low-triangular structure.

따라서 제한한 APP 병렬 알고리즘 수행을 위한 2 차원 Bit-parallel 시스톱릭 어레이 프로세서의 전체적인 흐름은 그림 2와 같으며, 기본 블럭도인 1-PE는 그림 3과 같이 세부설계 할 수 있다. 이때, 그림 2를 기초로 1-PE는 크게 데이터 연산부와 데이터 조절부로 나누어 연산을 수행할 수 있다. 그림 3.1은 데이터 연산부와 그림 3.2는 데이터 조절부의 세부 그림을 의미한다. 따라서 1-PE 내부의 데이터 흐름은 표 2의 4 단계를 따라 연산을 수행한다.

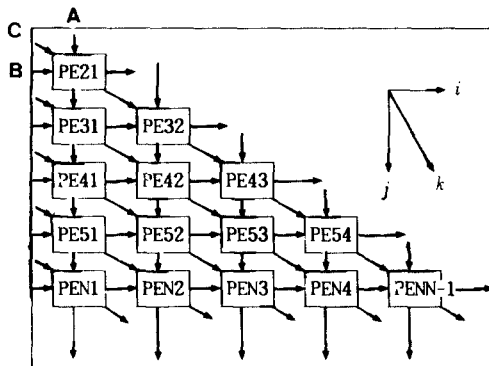


그림 2. 2차원 어레이 프로세서의 구성
Fig. 2. The configuration of 2-D array processor.

3. PE의 내부설계

그림 3.1 과 3.2 에서 구성된 기본 블럭도를 이용

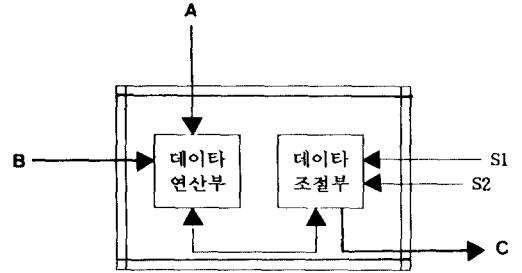


그림 3. 1-PE 구성도
Fig. 3. 1-PE Black Box.

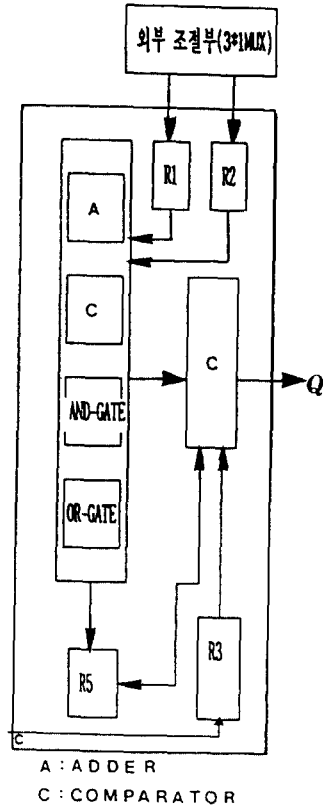


그림 3.1. 데이터 연산부
Fig. 3.1. Operation unit of data.

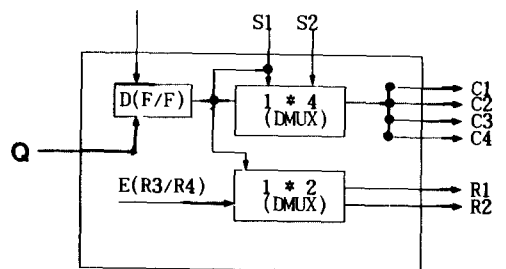


그림 3.2. 데이터 조절부
Fig. 3.2. Control unit of data.

표 2. 데이터 연산흐름

Table 2. Data operation flow

(STEP1)데이터입력과정: 외부 CONTROL 인 31 MUX를 통해 3 - 문제에 해당하는 데이터중 1개를 선택하여 레지스터 1, 2에 입력 한다.

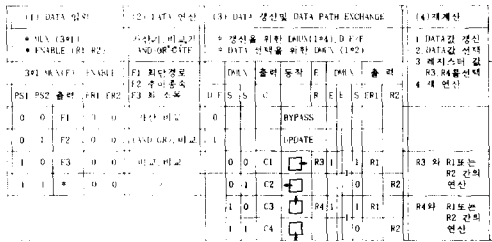
(STEP2)데이터연산과정: 각 레지스터1,2에 입력된 데이터값에 대해 문제에 해당하는 연산장치를 이용, 계산을 수행하고, 계산된 값을 레지스터 3에 입력 하고, 갱신처리를 수행한다.

(STEP3)데이터갱신과정: 연산과정후, 레지스터3 값이 갱신 도다면 14 DMUX를통하여 데이터 PATH를 변경하고 12 DMUX를 통해 변경된 데이터값의 재입력을 위해 레지스터1,2,3,4를선택한다.

(STEP4)데이터재입력과정: 갱신된 데이터값이 저장된 레지스터 3,4의선택과, 1,2 의선택에 의한 데이터를 재입력하여 (2)번 부터 (4)번까지의과정을 반복처리함으로써, 원하는 결과값을 얻을수있다.

표 3. 1- PE 내부의 데이터 흐름및 동작

Table 3. Data flow and operation in 1-PE.



하여 제안된 APP 병렬 알고리즘을 수행하기 위한 2 차원 Bit-parallel 시스톨릭 어레이의 1-PE 를 세부 설계 하였다.

전체적인 1-PE 의 세부설계는 그림 4 와 같으며

이때의 각 레지스터와 레지스터, 레지스터와 연산 기 사이의 데이터라인이 Bit-parallel 처리를 행함으로써 다소 H/W 복잡도는 증가하나 문제의 범용성 에 따른 고속화를 얻을 수 있었다. 이들의 세부동작 및 연산과정은 표 3 과 같다.

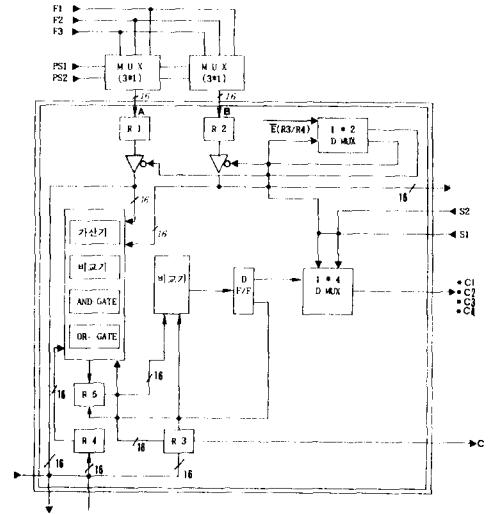


그림 4. 1-PE 세부설계

Fig. 4. 1-PE detail design.

4. 프로세서 상의 내부연산 과정

제안한 알고리즘에서, 기본 1-PE 의 세부설계에 따른 프로세서상의 전체적인 데이터 흐름은 그림 5와 같다. 구성된 1-PE 에서는 데이터 처리를 위하여 Projection i,j,k,l을 따라 데이터의 연산을 수행 할 때, 연산부를 통해 입력된 데이터 값의 연산결과와 이미 레지스터 3 에 입력된 데이터값의 연산 결과에 따라 데이터의 흐름은 달라진다. 즉 데이터 조절부의 D-MUX의 제어값에 따라서 데이터의 흐름은 여러

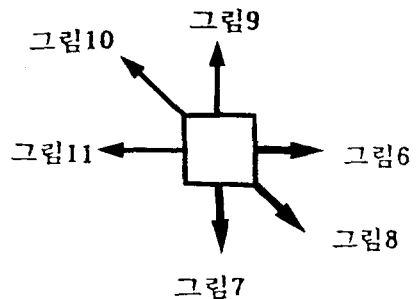


그림 5. 전체적인 데이터 흐름도

Fig. 5. General data flow.

방향으로 이동될 수 있다.그림 6 에서 그림 11은 데이터 이동경로의 방향을 의미하는것으로,그림 6,7, 8, 은 기본적인 데이터의 흐름을 의미한다. 즉, 연산 후 데이터값이 변경되지않는 경우를 의미 하고,그림 9,10,11은 연산후 데이터값이 변경된 경우를 의미한다. 그림으로 D-MUX의 값에 따라서 그림 9는 Up-pass 를 의미하고,그림 10은 Left - pass 를 의미하며, 그림 11은 Up-diagonal-pass를 의미한다.

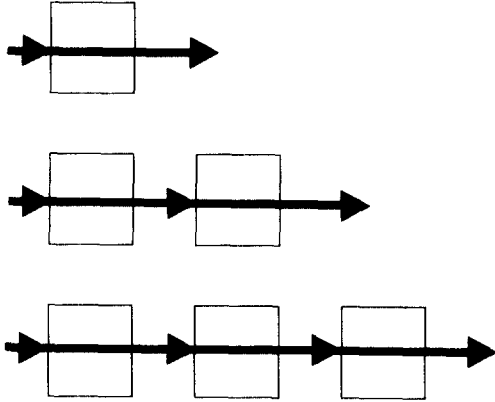


그림 6. PROJECTION I 에 따른 DATA 이동
Fig. 6. Data shift by projection i.

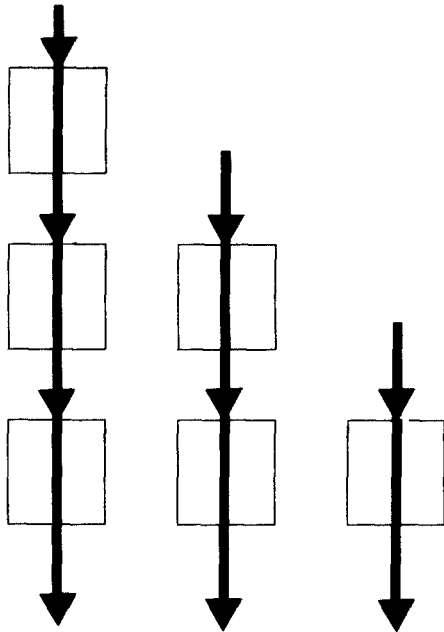


그림 7. PROJECTION J 에 따른 DATA 이동
Fig. 7. Data shift by projection j.

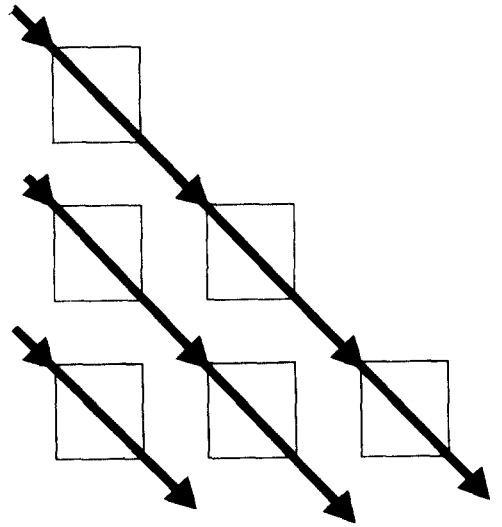


그림 8. PROJECTION K 에 따른 DATA 이동
Fig. 8. Data shift by projection k.

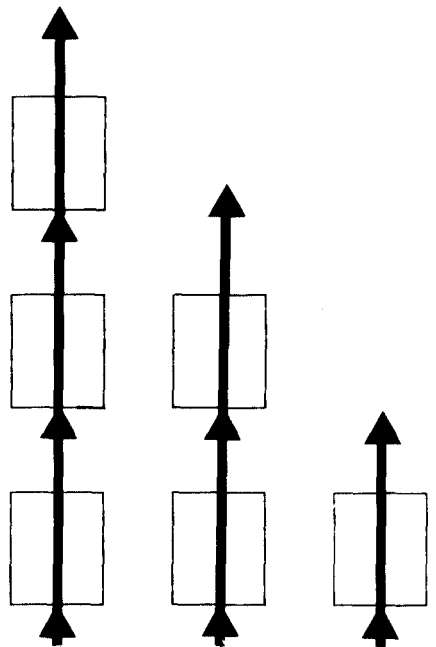


그림 9. UP-PASS 에 따른 DATA 이동
Fig. 9. Data shift by up-pass.

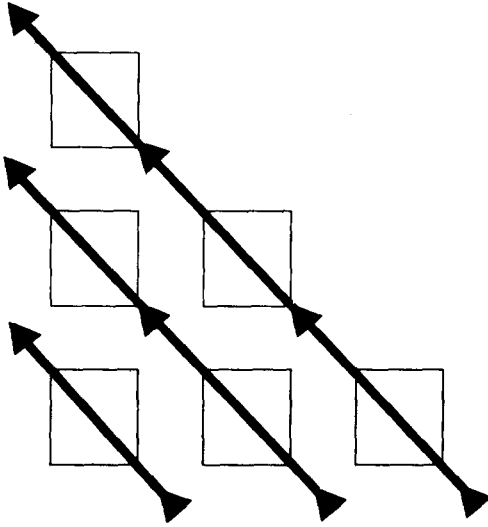


그림 10. UP-DIAGONAL-PASS 에 따른 DATA 이동
Fig. 10. Data shift by up-diagonal-pass.

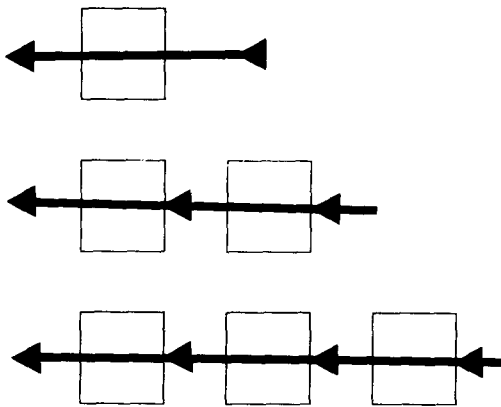


그림 11. LEFT-PASS 에 따른 DATA 이동
Fig. 11. Data shift by left-pass.

V. 성능 평가 및 분석

1. 아키텍처에 따른 성능비교 고찰

본 논문에서는 제안한 2차원 다기능 시스토크 어레이 프로세서의 성능을 아키텍처 측면에서 평가하기 위하여 제안한 APP 병렬 알고리즘을 기본으로 한 구조와 기존에 APP문제를 해결해왔던 WARSHALL-FLOID 알고리즘을 사용한 구조와의 복잡도(수행시간, PE수)에 따른 성능을 비교고찰하고, 최근 제안된 PARBS(Processor array reconfigurable bus) 구조^[7]와의 성능을 비교분석하여 그림 12, 13 과 같이

표현한다.

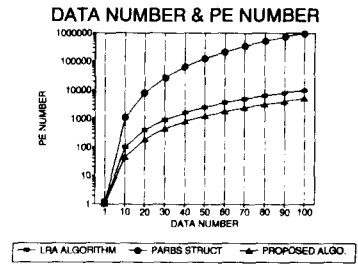


그림 12. DATA 수와 PE 수와의 관계
Fig. 12. Relations between data num. & pe num.

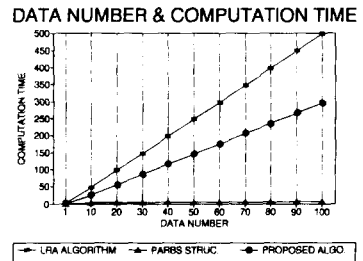


그림 13. DATA 수와 수행시간과의 관계
Fig. 13. Relations between data num & Computation time.

우선, S. Y. KUNG^[4]에 의해서 제안된 LRA 알고리즘을 기본으로 한 Orthogonal 시스토크 어레이 구조와 성능을 비교분석하면 표 4 와 같다. 이때, PE 수는 어레이 구조에 따른 PE들의 구성수를 의미하고 계산 시간의 산출은 전입력, 연산, 후출력 3 단계로 나누어 생각할 수 있다. 이때, 전입력 시간은 데이터들이 계산을 위해 미리 어레이로 이동되는 시간을 의미하고 연산은 어레이에서 실제 연산이 일어나는 시간을 의미하며, 후출력은 어레이에서 계산 되어진 결과를 인출하는데 걸리는 시간을 의미한다.

본 논문에서의 총 계산시간 산출은 전입력되는 시간은 고려하지 않으며 연산시간과 후출력시간의 합으로만 표현한다. 즉 1-PE에 대한 파이프라인 주기(데이터 입력열의 주기)를 1 이라 할때 제안된 어레이 구조에서의 전체 PE수는 $(N^2-N)/2$ 으로 구성되었기 때문에 총 클럭펄스에 따른 수행시간은 최초의 클럭펄스를 제외하고 매 클럭펄스마다 입력 데이터에 대해서 결과값이 파이프라인 주기 1만큼 있다가 출력됨

으로 데이터 N개에 대해서 3N-4 시간이 소요됨을 발견할 수 있다. 이때 연산에 필요한 클럭주기를 T 라 할때 T는 연산에 소요되는 시간의 합으로 표현될 수 있으며 클럭펄스의 주기는 내부 연산 알고리즘 각각에 따라서 달라질 수 있다. 이때, 각 알고리즘에 따른 내부 연산시간을 T-COMP 라 할때 다음과 같이 표현 할 수 있다.

SHORTEST PATH PROBLEM 의 T-COMP:

T-ADD+T-COMPARE 시간

TRANSITIVE CLOSURE PROBLEM 의 T-

COMP : T-AND + T-OR

+ T-COMPARE 시간

MINIMUM SPANNING TREE 의 T-COMP :

2 T-COMPARE 시간

따라서, 총 내부연산시간 T-COMP 는 다음과 같이 정의한다.

T-ADD + T-COMPARE + T-AND + T-OR
+ T-LATCH

이에 따른 총연산시간 T 는 T-COMP * Tcp(3N-4) 가 소요됨을 알 수 있다.

다음은 Feng^[7]에 의해서 제안된 PARBS 구조와 의 성능을 비교하면 표 4와 같다.이때 사용되는 알고리즘은 반복적인 행렬곱셈을 이용한것으로, APP알고리즘의 무방향그래프 성질을 이용하여 3-이동 단계, 즉 데이터 이동단계, 곱셈단계, 합산단계로 나누어 계산하는것으로, PARBS구조에서는 그중 곱셈단계 만을 취급한다.

표 4. 각 구조간의 비교

Table 4. The comparison of each structure.

	LRA 알고리즘 ^[6]	PARBS 구조 ^[7]	본 논문
계산시간	5N - 4	Log(N) * N	3N - 4
PE 수	N ²	N ²	(N ² -N)/2
구조적성격	정방향 S.A	PEs - Bus	하삼각 S.A
DATA 주기	1	1	1
GLOBAL-LINE	존재없음	존재함	존재없음

S.A : 데이터의 수에 따라 합산단계
N : 데이터의 수에 따라 합산단계

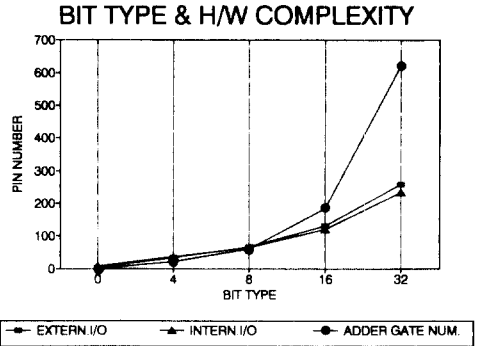


그림 14. 비트처리형태에 따른 H/W 복잡도

Fig. 14. H/W complexity by bit-processing type.

2. 2차원 하삼각 시스토릭 어레이 프로세서에 따른 성능평가

본 논문에서 제안한 APP 병렬 알고리즘을 이용한 2차원 Bit-parallel 시스토릭 어레이 프로세서의 성능을 평가하기 위하여 1-PE의 기능을 평가분석함으로써, 전체적인 아키텍처의 성능을 분석할 수 있다.

표 5. 1- PE 내부의 구성 요소

Table 5. Component in 1-PE.

1 - PE 내의 H/W 구성요소		
외부 데이터 라인수	입력	4개
	출력	4개
내부 데이터 라인 수 (입출력)		16개
연산기의 수	가산기	1개
	비교기	2개
	AND-GATE	1개
	OR-GATE	1개
	D-MUX	2개
	레지스터	5개
	D- F/F	1개
3-상태버퍼		2개
1 - PE 내 전체적인 H/W 구성요소		
일반식 (H-TOT)	$(N^2-N)/2H-FA + (N^2-N)H-COMP + (N^2-N)/2H-AND + (N^2-N)/2H-OR + (5N^2-5N)/2H-REG + (N^2-N)/2H-DF/F + (N^2-N)H-D MUX + (N^2-N)H-BUF$	
	일반식: 470N ² - 470N	

일반적으로 전용 프로세서 설계시 성능 평가는 크게 하드웨어 복잡도와 처리시간으로 구분지을 수 있다. 따라서 본 연구에서는 우선 1-PE내 집적되는 하드웨어 구성요소에 따른 게이트수의 산출로 하드웨어 복잡도를 평가 하여 그림 14와 같은 결론을 얻을 수 있었다. 또한 데이터 연산시간에 따른 성능을 분석할때, 중요한 평가요소인 Bit연산처리방식에 따른 Bit-parallel 방식과 Bit-serial 방식 으로 프로세서를 구성했을 경우를 각각 비교분석 함으로써 어레이 구조의 성능을 분석하여 그림 15,16 과 같은 결론을 얻을 수 있었다. 먼저 표 5 는 본 논문에서 제안된 1-

PE 내부설계에 따른 전체적인 하드웨어 구성요소로써, 주로 데이터 라인에 의한 입출력 핀(Pin)수와 연산기로 구성되어 있다.

이때 각 연산기는 모두 Bit-parallel 처리 방식으로 입.출력과 연산이 이루어지며, 본 절에서 구성된 모든 연산기는 게이트 단에서의 Logic 설계를 의미한다. 각 연산기의 기능은 표 6과 같다.

표 6. 1-PE 내부의 연산기의 기능
Table 6. Function of operator in 1-PE.

레지스터	기능
R1 / R2	A/B 입력 데이터를 저장
R 3	어레이 프로세서 내의 전 PE 에 초기 DATA 값 입력
R 4	DATA 값 갱신을 위한 레지스터
R 5	연산결과를 저장하는 레지스터
연산기	기능
가산기	F1 에 해당하는 DATA 값을 처리하는 연산기
비교기	F2 에 해당하는 DATA 값을 처리하는 연산기
AND-OR GATE	F3 에 해당하는 DATA 값을 처리하는 연산기
DMUX(1*2)	DATA 값 선택을 위한 연산기
DMUX(1*4)	DATA 결과값 선택을 위한 연산기
D-f/f	DATA 값의 변경 유무를 결정하는 연산기
제어선	기능
PS1, PS2	DATA 값을 선택하는 제어선
S1, S2	갱신된 DATA값의 PATH 를 제어하는선

DATA SIZE & EXTERN. I/O PIN

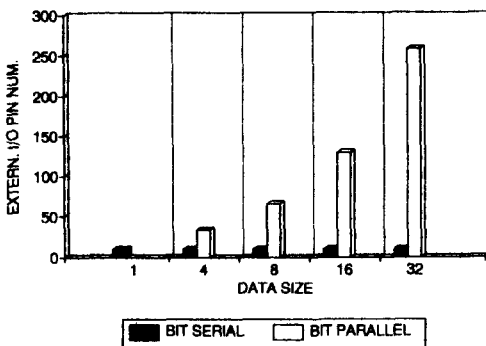


그림 15. 데이터 크기에 따른 외부 I/O 핀수
Fig. 15. External I/O pin number by data size.

DATA SIZE & INTERN. I/O PIN

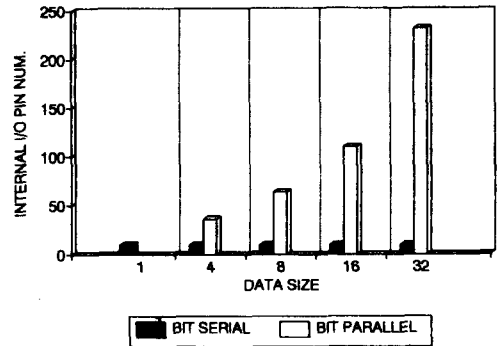


그림 16. 데이터 크기에 따른 내부 I/O 핀수
Fig. 16. Internal I/O pin number by data size.

표 7. 데이터 길이에 따른 비트연산처리 방식
Table 7. Bit-operation processing method by data length.

	BIT-SERIAL 처리		BIT-PARALLEL 처리	
	1 - PE & N - PE	1 - PE	N - PE	N - PE
외부 I/O PIN 수	10	34 개	50 개	50 개
내부 I/O PIN 수	15	36 개	50 개	50 개
연산기 중 ADDER - GATE 수	4	22 개	22 개	22 개

N = 4 Bit 인 경우

	BIT-SERIAL 처리		BIT-PARALLEL 처리	
	1 - PE & N - PE	1 - PE	N - PE	N - PE
외부 I/O PIN 수	10	66 개	226 개	226 개
내부 I/O PIN 수	15	84 개	204 개	204 개
연산기 중 ADDER - GATE 수	4	60 개	848 개	848 개

N = 8 Bit 인 경우

	BIT-SERIAL 처리		BIT-PARALLEL 처리	
	1 - PE & N - PE	1 - PE	N - PE	N - PE
외부 I/O PIN 수	10	130 개	962 개	962 개
내부 I/O PIN 수	15	120 개	848 개	848 개
연산기 중 ADDER - GATE 수	4	184 개	184 개	184 개

N = 16 Bit 인 경우

이때 연산시간의 성능을 높이기 위하여 Bit-parallel 처리방식을 이용하여 16 Bit 데이터에 대하여 연산을 수행함으로써, 표 5 에서 제시된 H/W 구성요소의 일반식을 기본으로 1-PE 내부에 집적가능한 게이트수는 총 940 개의 게이트가 소요됨을 분석할 수 있었으며, 이에 따른 하드웨어 복잡도는 $O(N^2)$ 임을 얻을 수 있었다. 따라서 표 7 은 Bit 처리방식에 따른 성능을 평가하기 위하여, 데이터 크기 (Word length) 변화에 따른 데이터 입출력 핀 수를 추정해 봄으로써, 실제 칩 설계시 VLSI 설계로의 유효성을 고찰할 수 있다.

즉, 1-칩내 집적가능한 PE수와 Bit 연산처리방식에 따른 입출력 핀 수와의 배선관계에 의해서 전체 칩 면적은 추정될 수 있다. 그럼으로 수행시간과 칩 면적 및 배선문제는 항상 이들 관계에 대한 절충점 (trade off)의 불가피성이 존재한다.

따라서 Thompson^[6]의 VLSI 복잡도 이론을 바탕으로, 본 논문에서 설계한 어레이 프로세서를 분석하면 표 8 과 같은 우수한 결론을 얻을 수 있었으며, 반도체 집적기술의 발전에 따라 1-칩내 집적가능한 게이트수를 약 40만개로 추정할 경우, 본 논문에서 제안된 구조의 성능평가 표 5 를 바탕으로 데이터 수 N 에 대해서 최대 15개까지 연산가능할 수 있음을 알고, 1-칩 내부에 PE수 105개까지 집적할 수 있는 구조임을 알 수 있다. 또한 Bit-parallel 연산처리방식에 의해서 데이터를 처리하기 때문에 공간상의 문제를 극복하며, 처리속도를 극대화할 수 있는 고성능 아키텍처의 유효성을 지닌다. 더 나아가 현 반도체 기술의 향상에 따라 Chip Level 에서의 PE 간의 접속으로 인한 대량의 데이터에 대한 실시간처리 문제를 향상시키고자 Wafer scale level 에서의 Chip간의 접속으로 인한 아키텍처를 실현할 경우, 동일한 구조와 동일한 알고리즘을 사용하여 구조상의 성능을 더욱 더 향상시킬 수 있을 것을 기대 확신한다.

표 8. 제안된 시스토크 프로세서의 성능평가
Table 8. The evaluation of Performance in proposed systolic processor.

	PE 수	계산시간	H/W 구성요소	AT ²	PT
제안된 구조의 복잡도	$(N^2 \cdot N)/2$ $O(N^3)$	$3N \cdot 4$ $O(N)$	$470N^2 - 470N$ $O(N^2)$	$O(N^4)$	$O(N^3)$

(단: N: 데이터 수, 데이터 크기: 16 Bit 정수, 데이터 처리방식: Bit-parallel)

VI. 결론

본 논문에서는 특수목적을 위한 전용 어레이 프로세서의 장점인 병렬처리를 최대한 이용하고, 그 기능상의 제한사항인, 융통성, 확장성의 문제를 극복할 수 있는 새로운 APP 병렬 알고리즘을 제안하여, 다기능 2차원 하삼각 시스토크 어레이 프로세서를 설계하였다. 따라서 기존 APP 알고리즘을 사용한 어레이 구조와 각각 비교 분석한 결과, PE 수가 감소하고, 수행시간이 단축되는 결과를 얻을 수 있었으며, 제안된 알고리즘의 병렬성을 토대로 프로세서를 설계하여 성능을 평가한 결과 수행시간 $O(N)$, PE 수 $O(N^2)$, 하드웨어 복잡도 $O(N^2)$ 의 수행결과를 얻을 수 있었다.

또한, 1-PE에 대한 세부설계를 행함에 있어, VLSI 설계시 입출력 핀 수가 일정하여 단순설계는 가능하나, 처리속도가 늦다는 단점을 갖고있는 Bit-serial 처리방식을 지양하고, 데이터 크기변화에 따른 다량의 데이터에 대해서 실시간처리를 할 수 있는 Bit-parallel 처리방식을 채택함으로써 구조적인 특성에 따라 고성능을 얻을 수 있는 아키텍처를 제안했다. 따라서 AT²^[6] 이론을 기본으로 VLSI 설계시 고려해야 하는 면적과 시간간의 절충점(Trade-off) 문제를 분석, 고찰해 봄으로써 VLSI 설계의 유효성을 확인할 수 있었다. 또한 수행시간 $O(N)$ 의 성능과 하드웨어 복잡도 $O(N^2)$ 의 결과는 VLSI설계의 용이함을 제시한다.

따라서, 본 논문은 기존의 array processor가 지닌 기능적 전용성의 문제를 극복할 수 있는, 다시말해 유사한 문제를 집적하여 기능적으로 확장된 개념의 array processor를 설계했다는 점에서 큰 특징 및 의의를 부각시킬 수 있다. 최근 들어서 계속되는 집적도의 발전 및 아키텍처 내부의 새로운 설계방법에 힘입어 대량의 데이터에 대해서 실시간으로 처리하기 위한 3차원 시 공간상에서의 설계방법에 관한 연구가 활발해져 오고있다. 더 나아가 Chip level 에서의 PE 간의 접속에 의한 프로세서 설계방법을 발전시켜 Wafer scale level 에서의 Chip과 Chip간의 접속으로 고성능 아키텍처를 꾀하려는 연구의 동향에 따라 본 연구는 동일한 구조와 동일한 알고리즘을 사용하여 구조상의 성능을 더욱더 향상시킬 수 있는 확신을 얻을 수 있었다.

본 논문의 응용분야로는 APP의 병렬 알고리즘특성과 확장성을 갖춘 어레이의 특징들을 고려하여 볼 때, 다량의 데이터처리를 요구하는 통신망 건설, 교통망건설에 유용하며, 단순소자의 신경망을 이용한 통신

상의 제반문제를 해결하는데 응용가능하다.

参 考 文 献

- [1] G.Rote, "A Systolic Array Algorithms for The Algebraic Path Problem", *Computing*, 34:191-219, 1985.
- [2] Y.robert, "Trends in Parallel Algorithms Design", *Parallel and Distributed Algorithms*, 1989.
- [3] M.J.Foster and H.T.Kung, "The Design of Special-Purpose VLSI Chips", *Computer Magazine*(to appear), CMU Computer Science Dept.technical report, Sept.1979.
- [4] S. Y.Kung, "VLSI ARRAY PROCES-SORS", Department of Electrical Engineering Princeton University, 1988.
- [5] S. Y.Kung, "Optimal Systolic Design for The Transitive Closure and Shortest P - ath Problems", *IEEE Computer*, c-36(5): 603-614, May, 1987.
- [6] S. Y.Kung and P.S.Lewis, "An Optimal Systolic Array for The Algebraic Path Pro - blem", *IEEE Trans.on Computer*, Vol.40, No.1, Jan., 1991.
- [7] Gen-Huey Chen, Biing-Feng Wang, and Chi-Jen Lu, "On the Parallel Computa-tion of The Algebraic Path Problems", *IEEE Trans.on Parallel & Distributed Syst.*, vol.3, no.2, March, 1992.
- [8] Ferng-Ching Lin and I-Chen Wu, "Broadcast normalization in Systolic Design", *IEEE Transaction on Comput-er*, vol.37, no.11, nov., 1988.
- [9] C.Thompson, "A Complexity Theory for VLSI" Ph.D thesis, Carnegie-Mellon Univ- ersity, Department of Computer Science, 1979.
- [10] Dan I.Moldovan and Jose A.B.Fortes, "Partitioning and Mapping Algorithms into Fixed Size Systolic Arrays", *IEEE Trans.Computer*, vol.35, no.1, Jan., 1986. vol.3, no.2, March, 1992.
- [11] V.Pan and J.Relif, "Fast and Efficient Solution of Path Algebraic Problem", *J. Computer. Syst. Sci.*, vol.38, pp.491-510, June, 1989.
- [12] Clauss Ph.Mongenet C., Perrin G.R., "Calculus of Space-Optimal Mapping of S- ystolic Algorithms on Processor Array", *IEEE Trans.on Parallel & Distributed system*, vol.3, May, 1990.
- [13] B.F.Wang and G.H.Chen, "Constant Time Algorithms for The Transitive Closure and Some Related Graph Problems on Processor Array with Reconfigurable Bus System", *IEEE Trans.on Parallel & Distributed Syst.*, vol.1, pp.500-507, Oct., 1990
- [14] Hsien-Fen Hsieh, "A Linear Systolic Algorithm for Finding Bridges on An Undirected Connected Graph", *Interna-tional Conference on Parallel Process-ing*, 1990.
- [15] Vipin Kumar and Vineet Singh, "Scalability of Parallel Algorithms for The All-Pairs Shortest Path Problem: A Summary of Results", *International Confer -ence on Parallel Processing*, 1990.
- [16] Cetin Kaya Koc and Rose Marie Piedra, "A Parallel Algorithm for Exact Solut-ion of Linear Equations", *International Conference on Parallel Processing*, 1991.
- [17] 방정희, 이현수, "APP용 다기능 ARRAY PROCESSOR 설계에 관한 연구", vol.15, no.2, *Proceedings of KITE Conference*, 1992. 11.

著者紹介



方晶姬(正會員)

1969年 2月 11日生. 1991年 2月 수원대학교 전자계산학과 졸업(이학사). 1993年 2月 경희대학교 전자계산공학과 졸업(공학석사). 1993年 경희대학교 전자계산공학과 박사과정 입학. 1993年 3月 ~ 현재 한국통신연구소 전임연구원근무. 주관심 분야는 Parallel algorithm, VLSI Architecture 설계, Intelligent Network 등임.



李顯洙(正會員)

1953年 10月 11日生. 1979年 2月 경희대학교 전자공학과 졸업(공학사). 1982年 2月 일본 경응의숙대학 전기공학과(공학석사). 1985年 2月 일본 경응의숙대학 전기공학과(공학박사). 1985年 ~ 현재 경희대학교 전자계산공학과 부교수 주관심분야는 Parallel algorithm, VLSI Architecture 설계 및 신경회로망의 응용 등임.