韓國 OA學會 論文誌
第3卷 第4號, 1998. 12.

98-3-4-2-4

Romberg 적분법을 위한 Systolic Array

박 덕 원*

Romberg's Integration Using a Systolic Array

Durk Won Park*

요 약

이 논문은 수치해석에서 적분값을 구하는데 이용되는 Romberg 적분법이 많은 계산량으로 인하여 소프트웨어적인 방법으로는 처리 속도가 떨어지므로 수치처리를 위한 툴 키트를 사용시 처리속도가 떨어진다. 그래서 이 논문에서는 시스토릭어레이를 이용하여 Romberg 적분법에 적분값을 구하는 새로운 하드웨어를 제안하였다. 이 새로운 하드웨어는 Romberg 적분법이 2단계로 나누어져있어서 2단계의 시스토릭어레이로 설계를 하였다. 첫 번째 단계는 사다리꼴 적분법에 의해서 근사치를 구하고, 두 번째는 단계는 구해진 적분값을 수렴속도도 빠르고 근사 값을 정확하게 하기 위해서 오차의 위수를 높여 가는 방법에 많이 사용하는 Richardson의 외삽법을 적용하여 적분값을 구하는 것이다.

Abstract

This paper proposed a systolic Arrays architecture for computing Romberg's integration method. It consists of systolic arrays of two stage, one for integration by Trapezoidal rule and the other for integration by using Richardson's extrapolation. the proposed its architecture is very high speed and regular. This paper illustrates how " mathematical hardware " package, as well as software library routines, may be part of the mathematical problem solver's tool kit in the future.

* 세명대학교 컴퓨터과학과 부교수
논문접수 : 98.11.10. 심사완료 : 98.12.12.

Ⅰ. 서 론

최근 들어 가격이 저렴하면서도 계산속도가 빠른 소형 반도체 소자들을 생산할 수 있는 VLSI 기술이 급격하게 발전하고 있으며 이런 반도체 기술에 힘입어 영상처리 등의 응용분야와 이들 분야의 기본처리 작업에 아주 중요하고 시간이 많이 걸리는 수치처리에서도 실시간 처리를 위한 요구가 많아지고 있으며, 또한 일부 분야에서는 실시간 처리가 가능하여 실생활에 도움을 주는 분야도 있다. 이런 요구에 부합하기 위해서 병렬처리에 대해서 많은 관심과 연구가 진행되어 왔고, 이 분야 또한 여러 가지 분야에서 괄목한 만한 성과를 가져왔다. 주로 영상처리, 음성인식, 로버트와 그 이외의 다양한 인공지능의 분야에서 많은 성과를 가져 왔다. 이렇게 병렬처리는 컴퓨터의 성능을 한 단계 높이고 실시간에 제약을 받던 작업들을 실시간 처리가 가능하도록 했다.

컴퓨터에 의한 수치처리는 아직도 하드웨어적인 기법보다는 소프트웨어적인 알고리즘을 의해서 처리되고 있다. 그러나 이러한 추세는 새로운 하드웨어의 기법과 많은 응용분야에서의 실시간 처리 요구에 의해서 하드웨어적인 방법으로 바뀌어 가고 있으며, 이에 따라서 국내외에서 많은 연구가 진행되고 있다. 그리고 이에 가장 적합한 하드웨어 기법 중에 하나가 시스토릭어레이(Systolic Array)라 할 수 있다. 어떠한 하나의 일을 처리하는데 있어서 전체 처리시간은 크게 입출력에서 걸리는 시간과 연산에 필요한 시간으로 볼 수 있는데 이들 중에서 입출력에서 걸리는 시간이 연산에 걸리는 시간보다 많은 그러한 일에 이 시스토릭어레이가 가장 적합하기 때문이다[1,2,3]. 이 하드웨어 기법은 한번 메모리로부터 가져온 데이터는 필요한 요소 요소에서 모두 사용된 후에 그 결과 또는 본래의 데이터를 다시 메모리에 저장하므로 보통의 대역폭을 가지고 높은 처리 효율을 가져올 수 있다. 결국 Compute-bound computation에 적합하다. 컴퓨터를 이용한 수치처리에는 각각의 처리에 따라서 많은 알고리즘이 개발되어 사용되고 있으나 이러한 알고리즘은 하드웨어보다는 소프트웨어에 중점을 두고 개발된 것이므로 실시간 처리를 위해서는 여러 가지 제약이 따른다. 특히 미적분을 컴퓨터에 의해서 처리하는 경우에는 많은 연산횟수로 인하여 입출력의 문제도 있지만 계산과정이 복잡하고 처리시간이 많이 걸려서 다른 응용분야에 적용을 하기 위해서는 하드웨어적인 기법을 도입하는 것이 필수적이라 하겠다. 그래서 이 논문에서는 적분을 수치처리하는 방법 중 결과가 가장 정확하여 많이 이용되는 Romberg의 적분법이 수식이 복잡하고 많은 계산 량으로 인해 소프트웨어적인 알고리즘에 의해서는 실시간 처리를 하는데 있어서 처리속도가 떨어지므로 시스토릭어레이로 구현을 하였다[4-9]

Ⅱ. Romberg의 적분법

적분법에는 사다리꼴 적분법, Simpson의 적분법, Romberg의 적분법, 미정계수법, Gauss의 적분법 등이 있으나 이들 중에서 비교적 정확하고 많이 사용되는 적분법이 Romberg의 적분법이나 이 방법은 계산 량이 많고 수렴하는 정도에 따라서 시간이 많이 소요된다.

Romberg의 적분법은 먼저 사다리꼴 적분법으로 근사 값을 구하고 다시 Richardson의 외삽법을 이용하여 좀더 정확한 값을 구하는 것이다. 이 과정은 2단계로 나누어져 있다. 1단계는 구간의 크기를 변화시키면서 그 구간의 크기에 대한 사다리꼴 공식으로 식(1)과 같이 $\int_a^b f(x)dx$ 를 구하는 것이다.

$$S_n^0 = \frac{h_n}{2}[f(a)+f(b)+2\sum_{i=1}^{2^n-1} f(a+ih_n)] \quad (1)$$

$$\left(\begin{array}{l} n=0,1,\cdots \\ h_n=\dfrac{(b-a)}{2^n} \end{array} \right)$$

S_n^0 는 $\int_a^b f(x)dx$ 를 $[a, b]$를 2^n개의 소구간으로 나누어서 적분 값을 구하는 사다리꼴 공식이다. 식(1)을 알고리즘에 적합한 형태로 변환한 것이 식(2), (3), (4) 이다.

$$S_0^0 = \frac{b-a}{2}[f(a)+f(b)] \qquad (2)$$

$$V_{n-1} = h_{n-1}\sum_{i=1}^{2^{n-1}} f(a+(i-\frac{1}{2})h_{n-1}) \qquad (3)$$

$$S_n^0 = \frac{1}{2}(S_{n-1}^0 + V_{n-1}) \qquad (4)$$

2단계는 위에서처럼 구한 값을 이용하여 수렴속도도 빠르고 근사 값을 정확하게 하기 위해서 수치해법의 한 방법으로 수치미분, 수치적분, 미분방정식 등에서 오차의 위수(Order)를 높여 가는 방법에 많이 사용하는 Richardson의 외삽법을 적용하여 식(5)로서 적분값을 구하는 것이다.

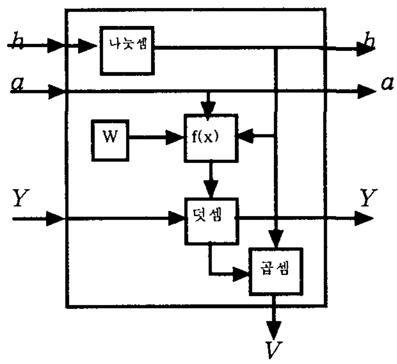
$$S_n^j = \frac{2^{2j}S_{n+1}^{j-1} - S_n^{j-1}}{2^{2j}-1} \qquad (5)$$

III. Systolic Array로 적용

1. 사다리꼴 적분법의 적용

Romberg의 적분법은 1단계, 2단계로 나누어서 값을 구하는데 먼저 1단계에서는 식(3)의 V 값을 구하는 것이 가장 시간이 많이 걸리는 식으로서 이것을 시스토릭어레이로 적용을 하여 값을 구하는 것이다. 이 V값을 구하는 것이 1단계로 구하는 과정을 시스토릭어레이로 적용한 아키텍쳐의 일반적인 형태는 그림1과 같이 나타낼 수 있다. 그림1 에서의 입력은 h, a, Y 로서 a는 적분을 위한 초기 값이고 h는 $(b-a)/2$가 되고, Y는 초기 값으로서 0 이 할당된다.

〔그림 1〕에서 셀은 $V_0, V_1, \ldots, V_n$ 의 결과가 나오는 셀과 그렇지 않은 일반적인 셀로 나누어지는데 결과가 나오는 셀의 구조는 그림2와 같고, 그 이외의 일반적인 셀의 구조는 그림3과 같다. 각 셀에서 W는 웨이트로서 각 셀이 항상 같은 웨이트 값을 갖는 것이 아니라 일정한 규칙으로 서로 다른 값을 갖는다. 셀이 갖는 웨이트를 보면 첫 번째 셀부터 1, 1, 3, 1, 3, 5, 7, 1, ... 로 웨이트 값이 홀수가 2^{n-1}개씩 부여된다.



$h \leftarrow h/2$
$a \leftarrow a$
$Y \leftarrow Y + f(a+h/2)$
$V \leftarrow Y \times h$

그림 2. 결과V가 나오는 셀의 구조
Fig 2. Design of a cell for computing

1단계에서의 시스토릭어레이는 사다리꼴 방식에 대해서 완전하게 적분 값을 구하는 것이 아니고 V값만을 구하고 나머지 값은 2단계의 시스토릭에서 구하게 된다. 그리고 여기서 구해진 값은 2단계의 입력으로 사용된다.

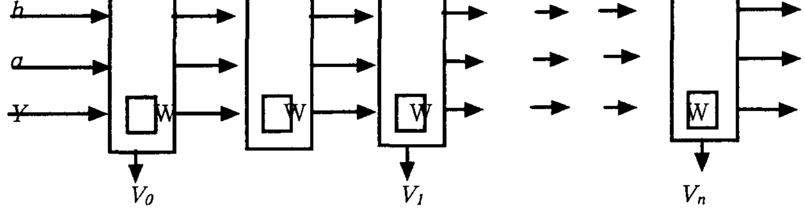


그림 1. V 값을 구하는 시스토릭어레이
Fig 1. Layout of a systolic arrays for computing a V

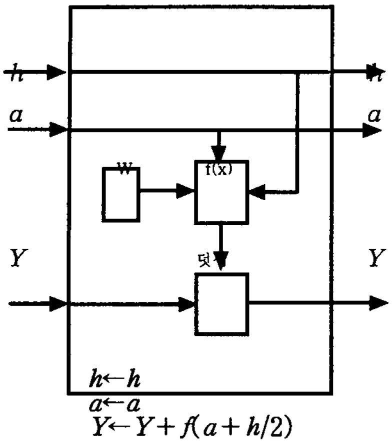


그림 3. 일반적인 셀의 구조
Fig 3. Design of a general cell

[그림 1]에서 입력 스트림 h, a, Y는 적분하려고 하는 함수가 n 개 인 경우

$$h_{n-1}, \ldots\ldots, h_2, h_1, h_0$$

$$a_{n-1}, \ldots\ldots, a_2, a_1, a_0$$

$$Y_{n-1}, \ldots\ldots, Y_2, Y_1, Y_0$$

와 같다. 그래서 여기서 제안하는 시스토릭어레이는 하나의 함수에 대해서 적분값을 구하는 것만이 아니라 연속되는 적분 값을 실시간으로 구하는 것이 가능하다. 그림3을 확장하여 1단계에서의 적분 값을 구하는 과정의 Snapshot을 보면 그림4와 같다. 그림4에서 모든 셀에서 V값이 계산되는 것이 아니고 1, 3, 7, 15, 31, ... 2^n-1셀에서 만 구해진다. 이렇게 구해진 값은 2단계의 입력으로 들어간다.

2. Richardson의 외삽법에 (Richardson's extrapolation) 대한 적용

위에서 1단계에서의 시스토릭어레이에 의해서 구해진 V값은 사다리꼴 방식에 의한 적분값으로 이것에 오차의 위수(Order)를 높여 가는 방법에 많이 사용하는

Richardson의 외삽법을 적용하는 것이다. 이렇게 적용한 것이 식(8)이다. 2단계에서는 이것을 시스토릭 아키테처로 설계하는 것이다. 2단계의 적분 값을 구하는 시스토릭어레이의 일반식은 그림7과 같다. 식(7)은 첫 번째 열에서 구해지며 식(8)은 첫 번째 열에서 구해진 값을 입력으로 해서 2번째부터 구해진다. 이렇게 구해지는 값은 일정한 범위 값으로 오차가 줄어들면 구하는 적분 값이 된다. 그림5 에서 셀의 구조는 두가지 형태가 있다. 이 셀을 구조는 그림6과 같다.

$$S_n^0 = \frac{1}{2}(S_{n-1}^0 + V_{n-1}) \qquad (7)$$

$$S_n^j = \frac{2^{2j} S_{n+1}^{j-1} - S_n^{j-1}}{2^{2j}-1} \qquad (8)$$

그림5 에서 초기치 S00 는 식(2)와 같으나 미리 값을 구해서 입력시키는 것이다. 그림5에서 셀은 첫 번째 행의 셀과 그 이외의 행에서의 셀로 두 가지 형식으로 나누어 지는데 첫 번째 행의 셀은 그림6의 a)에 그리고 나머지 셀은 그림6의 b)와 같다. 그리고 2단계 적분 값을 구하는 snapshot은 그림7에서 보여준다.

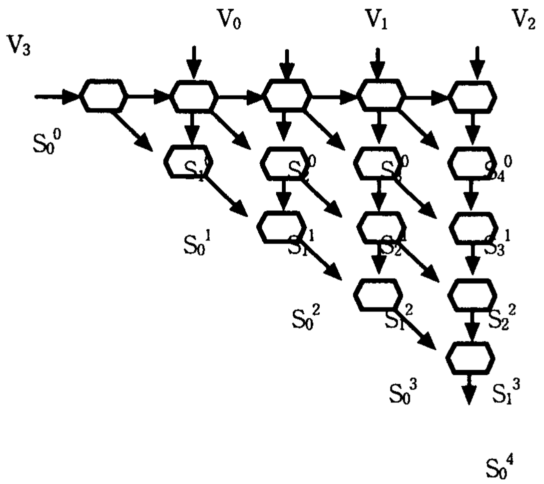


그림 5. 2단계의 적분값을 구하는 시스토릭어레이
Fig 5. Systolic arrays for Richardson's extrapolation

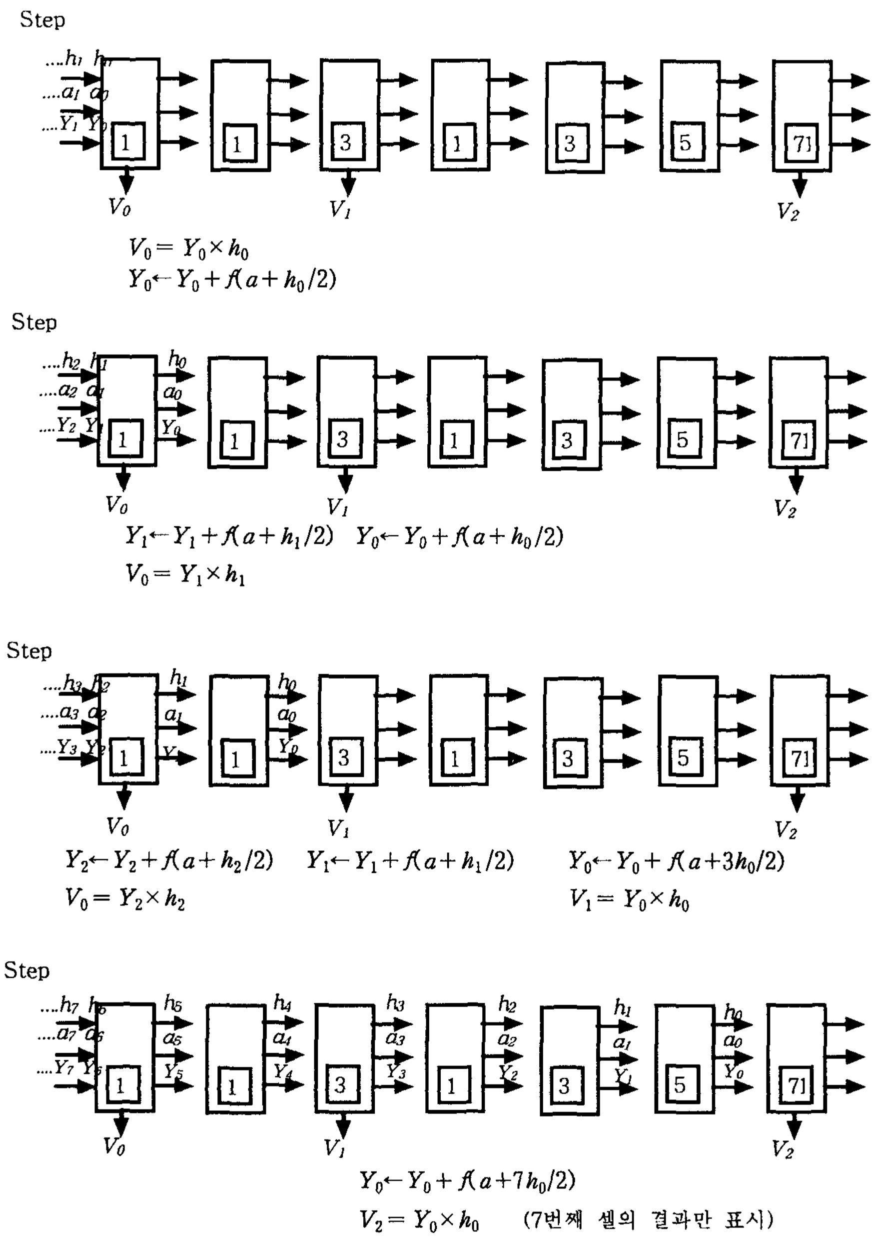


그림 4. 사다리꼴 적분값을 구하는 시스토릭어레이의 Snapshot
Fig 4. Snapshot of systolic arrays for Trapezoidal rule

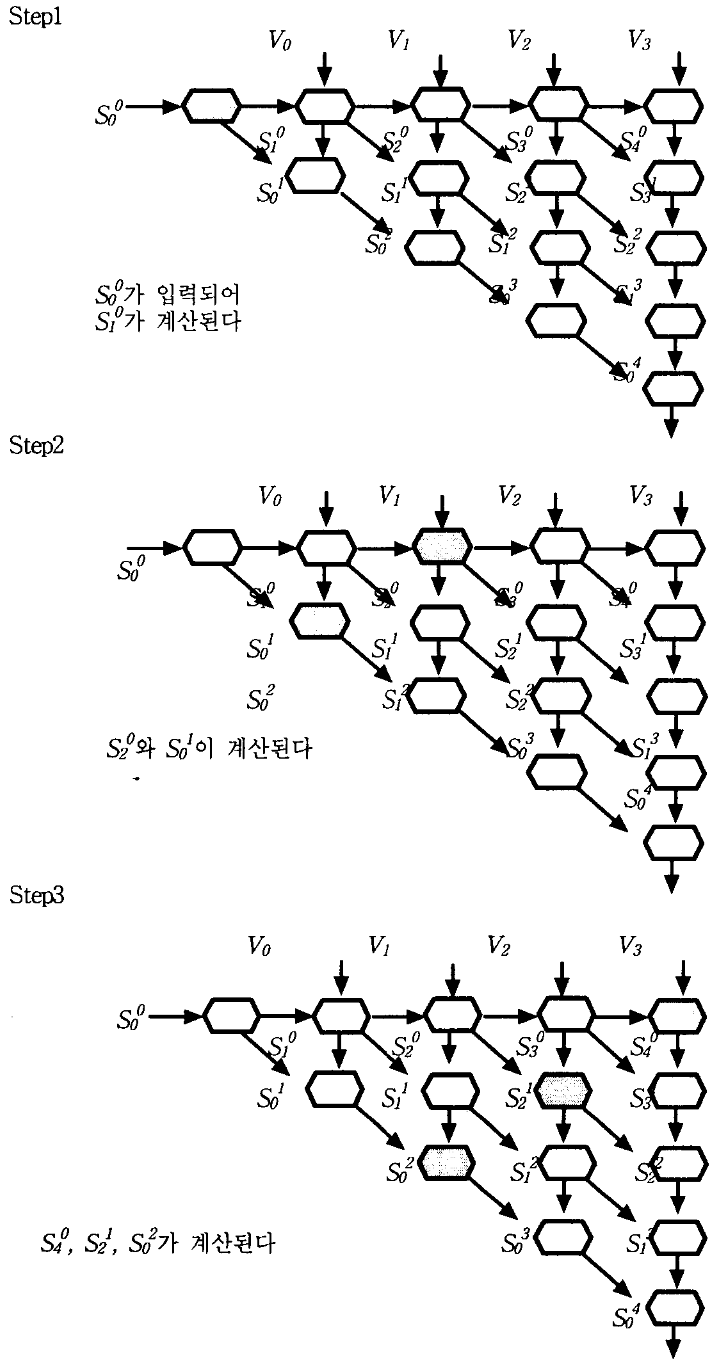


그림 7. 2단계 적분값을 구하는 시스토릭어레이의 Snapshot
Fig 7. Snapshot of systolic arrays for Richardson's extrapolation

IV. 결론

이 논문에서는 수치해석에서 적분 값을 구하는 알고리즘이 소프트웨어적인 방법으로는 처리시간이 많이 걸려서 실시간 처리에 어려움이 많고, 연속처리 하는데는 더욱 어려움이 많아서 이것을 시스토릭어레이로 제안을 했다. 적분법 중에서 Romberg 적분법이 사다리꼴 적분법과 Simpson 적분법에 비해서 수렴이 정확하고 정확한 적분 값의 근사치를 구할 수 있어서 많이 이용되므로 이것을 시스토릭어레이로 제안한 것이다. 제안한 시스토릭어레이는 Romberg의 적분법이 크게 두 단계로 나누어져 있어서 두 단계로 처리를 하였다. 1단계는 사다리꼴 적분법을 시스토릭어레이로 적용하여 적분의 근사치를 구하고 것이고, 2단계는 1단계에서 구해진 적분값을 수렴속도도 빠르고 근사 값을 정확하게 하기 위해서 수치해법의 한 방법으로 수치미분, 수치적분, 미분방정식 등에서 오차의 위수(Order)를 높여 가는 방법에 많이 사용하는 Richardson의 외삽법을 적용하여 적분값을 구하는 것이다. 이 방법은 시뮬레이션 한 결과 정확한 값을 얻을 수 있었다.

이 논문에서 제안한 하드웨어는 소프트웨어적으로 적분 값을 구하는데 걸리는 시간에 비해서 빠르므로 실시간 처리에 필요한 응용분야에 적용이 가능하며, 수치해석의 문제를 해결하는 툴 키트의 한 부분으로서 이용이 가능하다. 그리고 이 하드웨어는 연속해서 새로운 함수의 적분값을 구할 수 있다는 것이 기존의 소프트웨어적인 방법에 비해 좋은 잇점이라 하겠다.

이 논문에서의 보완해야 하는 부분으로서는 각 셀에서의 함수 값을 처리하는 부분이 속도가 떨어져 하나의 셀에서 걸리는 사이클시간이 길어질 수 있다는 것이다. 그리고 1단계에서의 V값이 매 셀에서 구해지는 것이 아니라 2n-1번째 셀에서 구해지므로 몇 개의 사이클 당 하나의 결과를 구할 수 있다는 것이 앞으로 더 연구되고 보완되어져야 한다.

참고문헌

〔1〕 박덕원, " Systolic Arrays를 이용한 경계선 검출에 대한 연구", 숭실대학교 대학원 석사학위 논문. 1987.

〔2〕 박덕원, "영상처리의 국부적 연산을 위한 Systolic Arrays", 세명대학교, 세명논총, 제6집, 175-193. 1997.

〔3〕 Mckeon, G. P., " Iterated Interpolation Using a Systolic Array ", ACM Trans. on Mathematical Software, vol.12, No.2, 162-170. 1986.

〔4〕 Kung, H. T., " Why Systolic Architecture ", Computer Magazine 15(1), 37-46. 1982.

〔5〕 Kai.Hwang and Faye A.Briggs, " Computer Architecture and Parallel Processing ", Mcgraw. Hill Series in Computer Organization and Architecture, 769-807. 1984.

〔6〕 Fisher, A. L. and Kung, H. T., " *Synchronizing Large VLSI Processor* Arrays ", IEEE Trans. on Computers, vol. c-34, no.8. 1985.

〔7〕 Kung, H. T and Picard, R. L., " One-dimensional Systolic Arrays for Multidimensional Convolution and Resampling ", VLSI for Pattern Recognition and Image Processing, Spring-verlag, 9-24. 1984.

〔8〕 Fisher, A. L., Kung, H. T. and Monier, L. M., " Architecture of the PSC : A Programmable Systolic Chip ", Proceeding of the 10th Annual Symposium on Computer Architecture. 1983.

〔9〕 Bridges, G. E., " Dual Systolic Architecture for VLSI Digital Signal Processing System ", IEEE Trans. on Computers, vol. c-35, no.10, 916-923. 1986.

저 자 소 개

박덕원

1986년:숭실대학교 전자계산학과 졸업(학사)

1988년:숭실대학교 대학원 전자계산학과 졸업(석사)

1997년 : 충남대학교 대학원 계산통계학과 졸업(박사)

1988년 - 1991 충남전문대학 조교수

1991년 - 현재 세명대학교 컴퓨터과학과 부교수

관심분야 : 컴퓨터아키텍처, 영상처리, 병렬처리.