

부분행렬을 사용한 행렬 · 벡터 연산용 1차원 시스톨릭 어레이 프로세서 설계에 관한 연구

A Study On Improving the Performance of One Dimensional Systolic Array
Processor for Matrix · Vector Operation using Sub-Matrix

김 용 성 (Kim, Yong Sung)*

목 차

- I. 서론
 - II. 1차원 행렬 · 벡터 연산용 시스톨릭 어레이 프로세서 기본적인 설계
 - III. 분할된 부분행렬을 이용한 1차원 행렬 · 벡터 연산용 시스톨릭 어레이 프로세서
 - IV. 실험 및 고찰
 - V. 결론
-

Abstract

Systolic Array Processor is used for designing the special purpose processor in Digital Signal Processing, Computer Graphics, Neural Network Applications etc., since it has the characteristic of parallelism, pipeline processing and architecture of regularity. But, in case of using general design method, it has initial waiting period as large as No. of PE-1. And if the connected system needs parallel and simultaneous outputs, processor has some problems of the performance, since it generates only one output at each clock in output state.

So in this paper, one dimensional Systolic Array Processor that is designed according to the dependance of data and operations using the partitioned sub-matrix is proposed for the purpose of improving the performance. 1-D Systolic Array using 4 partitioned sub-matrix has

* 여주대학 게임엔터테인먼트과 교수

efficient method in case of considering those two problems.

Key words: Systolic Array Processor, Pipeline processing, sub-matrix.

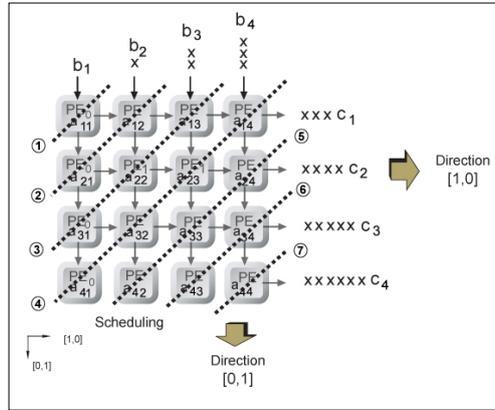
I. 서론

고속화 처리가 요구되는 디지털 신호처리, 그래픽, 신경망 등의 분야는 전용 프로세서 설계에 병렬처리 및 파이프라인 처리를 많이 사용하고 있다. 시스톨릭 어레이 프로세서는 병렬성과 파이프라인 처리의 특성을 가지고 있으며, 기본구조를 반복하여 사용할 수 있으므로 설계의 용이성을 가지고 있어서 이러한 고속의 전용 프로세서 설계에 적합하다^[1,2,3,4,5].

시스톨릭 어레이 프로세서는 입출력 요소에 의한 복잡성보다 연산량의 복잡도에 의존되는 연산처리의 고속화에 적합하므로 행렬 연산을 기본으로 하는 전용프로세서 설계에 주로 사용한다. 그래픽 분야에서 그래픽 기본요소 처리기, 전용 곡선 발생기, 투영 변환기, 벡터폰트 생성기 등에는 행렬·벡터 연산을 기본으로 사용하며^[6,7,8,9,10], 신경망의 경우에도 MAC(Multiplier with Accumulator)연산을 기본으로 하므로 연산부의 기본을 행렬·벡터 연산을 기본으로 사용할 수 있으므로^[11], 행렬·벡터 연산용 1차원 시스톨릭 어레이 프로세서의 속도 개선과 유용성의 향상은 연관분야의 전용 프로세서를 성능을 향상시키는 주요한 요소가 된다. 시스톨릭 어레이 프로세서의 체계적인 설계방법은 연산에 사용될 데이터와 연산장치와의 상호 의존성과 순환성에 따른 설계도에서 데이터 연산처리 순서를 고려하며, 도식적인 방법과 사상(mapping)을 사용하여 병렬성과 파이프라인 처리를 갖는 시스톨릭 어레이로 설계하는 방법이 대표적이다^[1]. 그러나 행렬·벡터 연산용 시스톨릭 어레이 프로세서의 경우 행렬연산의 기본방식을 변형 없이 적용함으로써 연산결과를 한 개씩 순차적으로만 산출하는 유용성에 문제점과 연산 지연시간 문제점을 갖는다. 그러므로 본 논문에서는 연산에 사용되는 행렬을 부분행렬로 분할하고 데이터의 종속성에 따른 설계를 수행하여 기본적인 연산지연 시간을 감소시키며, 연산 결과를 병렬성 있게 산출함으로써 유용성을 개선한 1차원 행렬·벡터 연산용 시스톨릭 어레이 프로세서를 설계하여 응용분야의 전용 프로세서에 대한 성능을 향상 시키는데 기여하고자 한다.

II. 1차원 행렬·벡터 연산용 시스톨릭 어레이 프로세서 기본적인 설계

일반적인 $[4 \times 4]$ 행렬 A와 $[4]$ 벡터 B의 경우는 식 (1)과 같이 표현된다. 행렬 A와 벡터B의 승산결과 벡터C는 식 (2)와 같은 연산을 수행하며, 이 연산은 곱의 누적된 합으로 표현되므로 승산과 가산으로 이루어진다^[1].



<그림 1> 행렬·벡터 연산용 어레이 프로세서의 SFG(n=4)

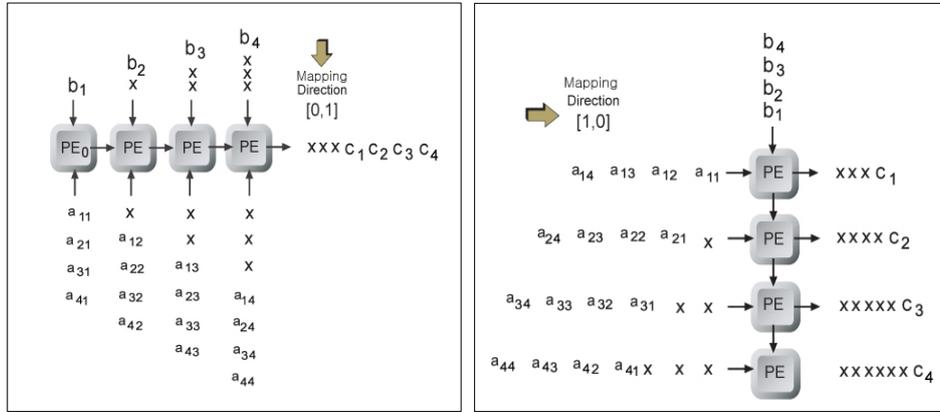
$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \tag{1}$$

$$c_i = \sum_{j=1}^n a_{ij} \times b_j \quad (i=1, \dots, n, n=4) \tag{2-1}$$

$$c_i^k = \sum_{j=1}^k a_{ij} \times b_j \quad (i=1, \dots, n, n=k=4) \tag{2-2}$$

$$c_i^k = c_i^{k-1} + a_{ik} \times b_k \quad (i=1, \dots, n, n=k=4) \tag{2-3}$$

식 (2-1)의 행렬 a_{ij} 와 벡터 b_j 의 곱의 합을 식(2-2)와 같이 표현하면 식 (2-3)과 같이 순환적 표현된다. 이 식에 대하여 연산의 종속관계에 대한 DG(Dependence Graph)에 대한 2차원 구조의 SFG(Signal Flow Graph)를 그림 1에 표시하였다, $[4 \times 4]$ 행렬 $a_{11} \sim a_{44}$ 는 각 기본처리요소(PE : Pprocessing Element)에 할당되고, 열 방향으로 각각 $b_1 \sim b_4$ 가 입력되어 매 클럭(clock) 마다 승산과 전단의 연산결과가 누적되어 식(2)의 연산이 수행되며, 3 클럭의 지연시간 후 4번째 클럭에서 첫 번째 연산이 출력되고, 이 후 매 클럭 마다 다음 행의 끝 PE에서 출력이 생성되므로, $[n \times n] \cdot [n]$ 의 경우 $n-1$ 클럭의 지연시간 후 n 클럭 후 최종 출력이 생성된다. 그림 1.에서 연산과정에 따라 동일 시간에 수행되는 PE를 점선으로 표시하였고, ①~⑦순서대로 수행되는 스케줄링(Scheduling)을 나타내었다. 이에 따라 $[1,0]$ 방향과 $[0,1]$ 방향으로 사상(Mapping)한 행렬·벡터 연산용 1차원 시스틀릭 어레이 프로세서를 그림 2에 표시하였다.



(a) [0,1]로 사상된 경우

(b) [1,0]으로 사상된 경우

<그림 2> 행렬 벡터 연산용 1차원 시스틀릭 어레이 프로세서의 기본구조

[0,1]로 사상된 경우는 k번째 PE에 $a_{ik}(i=1,..,4)$ 와 b_k 가 입력되어 연산되며 연산결과는 k+1번째 PE에 전달되어 누적되어 연산된다. 4클럭 이후 매 클럭마다 $c_i(i=1,..,4)$ 가 출력된다. [1,0]로 사상된 경우는 k번째 PE에 $a_{kj}(j=1,..,4)$ 와 $b_j(j=1,..,4)$ 가 입력되고 연산 누적되어 c_k 가 출력된다. 기본형의 문제점은 $[n \times n] \cdot [n]$ 의 경우 n-1클럭의 지연시간을 갖는 것과 그래픽 등에 응용 시에 두 개의 좌표에 대한 값이 동시에 병렬 출력되어야 하는 경우는 유용성이 감소하는 단점을 갖는다.

III. 분할된 부분행렬을 이용한 1차원 행렬 · 벡터 연산용 시스틀릭 어레이 프로세서

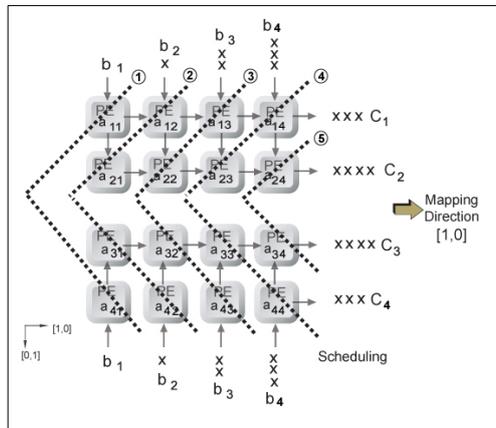
본 논문에서는 초기연산 지연시간의 감소와 두 개 이상의 연산결과를 동시에 병렬로 출력하기 위하여 행렬 연산 시 $[n \times n]$ 행렬을 $[m \times r]$ 행렬($p=n/p, q=n/r, p,q,n,m$:정수)인 부분행렬로 분할하여 다음과 같이 3가지 방식의 행렬 · 벡터 연산용 1차원 시스틀릭 어레이를 제안하고자 한다.

1. 분할 부분행렬을 사용한 행렬 · 벡터 연산용 시스틀릭 어레이 프로세서의 설계

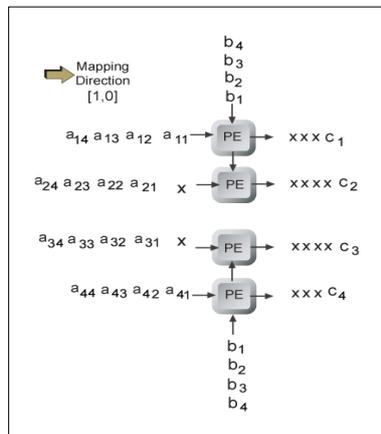
n=4인 경우, $[4 \times 4]$ 행렬 · $[4]$ 벡터의 연산에서 $[4 \times 4]$ 행렬을 2개의 $[2 \times 4]$ 부분행렬로 분할하면 식(3)과 같이 2개의 $[2 \times 4]$ 행렬 · $[4]$ 벡터의 연산으로 표현할 수 있다. 식(3)의 연산에 대한 DG를 구성하고 이에 대한 SFG를 그림 3에 나타내었다.

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (3-1)$$

$$\begin{bmatrix} c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (3-2)$$



<그림 3> 2분할 부분행렬을 이용한 행렬 · 벡터 연산용 어레이 프로세서의 SFG(n=4)



<그림 4> 2분할 부분행렬을 이용한 행렬 · 벡터 연산용 1차원 시스틀릭 어레이 프로세서(n=4)

식(3-1)의 연산은 상단부 2개의 행을 구성하는 PE에서 수행하며, 식(3-2)의 연산은 하단부 2개의 PE에서 수행한다. 연산의 순서는 ①~⑤의 순서로 수행되므로, 초기지연시간 4클럭 이후

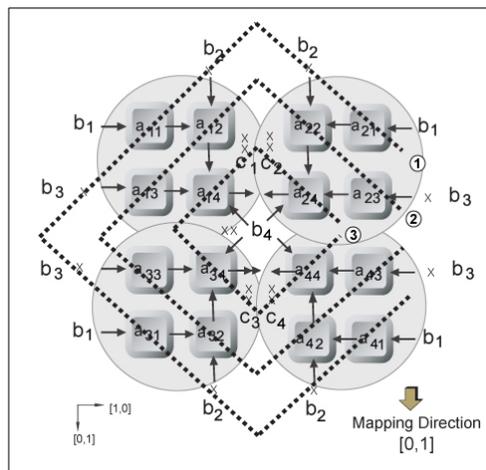
c_1 과 c_4 이 출력되고, 다음 클럭에 c_2 와 c_3 가 출력되므로 동시에 병렬로 출력할 수 있다. SFG에서 [1,0]로 사상한 시스틀릭 어레이 프로세서를 그림 4에 표시하였다. 그림 3의 스케줄링에 따라 그림 4 시스틀릭 어레이의 PE에 행렬 a_{1j} 와 $a_{4j}(j=1,\dots,4)$ 는 동시에 입력되고, 1클럭 후에 a_{2j} 와 $a_{3j}(j=1,\dots,4)$ 에 입력된다. $b_j(j=1,\dots,4)$ 는 PE의 상하 열 방향으로 매 클럭마다 순차적으로 입력되므로 4번째와 5번째 클럭에 2개씩의 연산결과를 동시 출력할 수 있다.

2. 4분할 부분행렬을 사용한 행렬 · 벡터 연산용 시스틀릭 어레이 프로세서의 설계

식 (1)에서 행렬을 1차 행벡터로 분할하면 식(4)와 같이 4개의 행벡터와 열벡터의 곱으로 표현이 되고, 그림 5에서 원형 내에 표시된 4개의 PE로 구성되며, 각각에서 $c_j(j=1,\dots,4)$ 의 연산 결과가 산출된다.

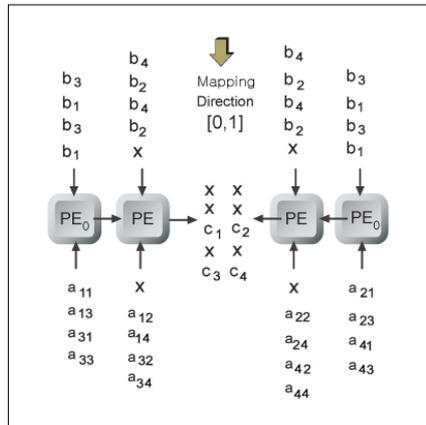
$$[c_i] = [a_{i1} a_{i2} a_{i3} a_{i4}] \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}, \quad (i=1,\dots,4) \tag{4}$$

그림 5에서 스케줄링은 점선으로 표시된 ①~③의 순서에 따라 처리되어 3번째 클럭에 4개의 처음 연산 결과가 산출된다. 이에 따라 [0,1]방향으로 사상한 시스틀릭 어레이 프로세서를 그림 6에 표시하였다. 그림 6에서 1, 2번째 PE에 행렬 값 a_{1j}, a_{3j} 과 $b_j(j=1,\dots,4)$ 이 입력되고, 4, 3번째 PE에 행렬 값 a_{2j}, a_{4j} 와 $b_j(j=1,\dots,4)$ 가 입력된다. 그림 5에서 ③번 연산 시 ①번에 해당되는 PE에서 연산이 동시에 이루어지므로 사상된 그림 6의 프로세서의 연산 결과는 3 번째



<그림 5> 4분할 부분행렬을 이용한 행렬 · 벡터 연산용 어레이 프로세서의 SFG(n=4)

클럭에서 c_1 과 c_2 의 출력이 발생되고, 이후 두 클럭마다 두개의 출력이 생성된다.

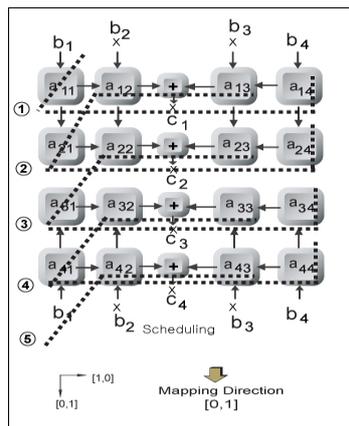


<그림 6> 4분할 부분행렬을 이용한 행렬·벡터 연산용 1차원 시스틀릭 어레이 프로세서(n=4)

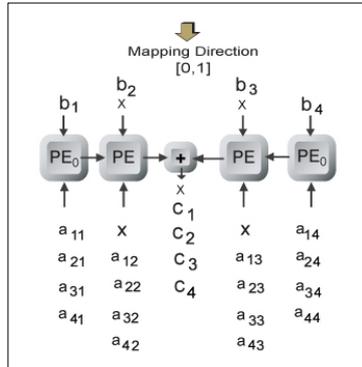
3. 8분할 부분행렬을 사용한 행렬·벡터 연산용 시스틀릭 어레이 프로세서의 설계

행벡터를 다시 두개로 분할하면 식(4)는 식(5)와 같이 2원소의 행과 열 벡터로 분리되므로, 8개의 행과 열벡터로 분할하여 연산을 수행하면 전체 연산결과 c_j ($j=1, \dots, 4$)을 산출할 수 있다.

$$[c_i] = [a_{i1} \ a_{i2}] \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + [a_{i3} \ a_{i4}] \cdot \begin{bmatrix} b_3 \\ b_4 \end{bmatrix}, \quad (i=1, \dots, 4) \quad (5)$$



<그림 7> 8분할 부분행렬을 이용한 행렬·벡터 연산용 어레이 프로세서의 SFG(n=4)

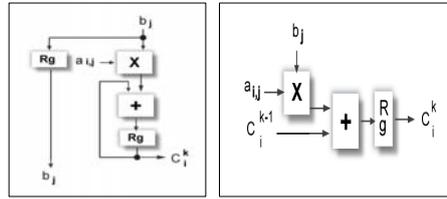


<그림 8> 8분할 부분행렬을 이용한 행렬·벡터 연산용 1차원 시스틀릭 어레이 프로세서(n=4)

8분할 부분행렬을 이용한 행렬·벡터 연산용 어레이프로세서의 SFG를 그림 7에 나타내었다. 식(5)에서 $[a_{i1} \ a_{i2}] \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ 의 연산은 상단 좌측 2개의 PE에서 수행되고, $[a_{i3} \ a_{i4}] \cdot \begin{bmatrix} b_3 \\ b_4 \end{bmatrix}$ 의 연산은 우측 2개의 PE에서 수행되며, 수행결과는 중간 가산기에서 연산 되어 출력 c_i ($i=1, \dots, 4$)가 생성된다. $a_{ij} \times b_j$ ($j=2,3$) 연산과 $a_{i+1j} \times b_j$ ($j=1,4$)의 연산은 동시에 수행되므로 스케줄링은 번호 ①~⑤순서에 따라 진행된다. 이에 따라 [0,1]방향으로 사상한 1차원 시스틀릭 어레이를 그림 8에 표시하였다. 그림 8에서 1, 2번째 PE에 행렬 값 a_{ij} ($i, j=1, \dots, 4$)와 b_j ($j=1, \dots, 4$)는 각 PE에 순서적으로 입력되고, a_{i2j} , a_{i3} , b_2 , b_3 ($i=1, \dots, 4$)만 한 클럭 지연되어 입력되므로 두 번째 클럭에서 처음 연산 결과 c_1 이 출력되고 이후 한 클럭마다 출력이 생성되므로 초기지연 시간을 감소시킬 수 있다. 구조상 가산기의 수가 증가할 것으로 예상되지만 좌측과 우측의 첫 번째 PE에서는 누적 연산이 필요 없으므로 가산기의 수는 증가하지 않고 감소하게 된다.

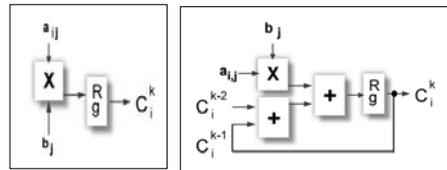
4. 제안된 시스틀릭 어레이 프로세서의 PE설계

본 논문에서 제안한 시스틀릭 어레이 프로세서의 기본처리요소(PE)를 내부 구성도를 그림 9에 나타내었다. 그림 9의 (a)는 그림 2의 (b)와 그림 4의 2분할 한 경우에 해당되며, 식(2-3)의 행렬 a_{ik-1} 와 벡터 b_{k-1j} 의 곱의 누적 c_i^{k-1} 은 승산기와 가산기에서 수행되어 출력 c_i^k 를 출력한다. 그림 9의 (b)는 그림 2의 (a)와 그림 8의 PE에 해당되며, 전단 PE의 c_i^{k-1} 와 현재 PE의 승산 값을 누적한 c_i^{k-1} 을 다음 단으로 전달한다. 그림 9의 (c)는 그림 2(a), 그림 6, 그림 8의 첫 번째 PE인 PE₀이 이므로 누적을 위한 가산기가 생략할 수 있다. 그림 9의 (d)는 그림 6의 종단 PE에 해당되며, 전단 PE의 c_i^{k-2} 와 현재 PE의 전단계 누적 값 c_i^{k-1} 및 현재 승산한 값이 모두 누적되어야 하므로 두 개의 가산기가 필요하다.



(a) 그림 4

(b) 그림 8

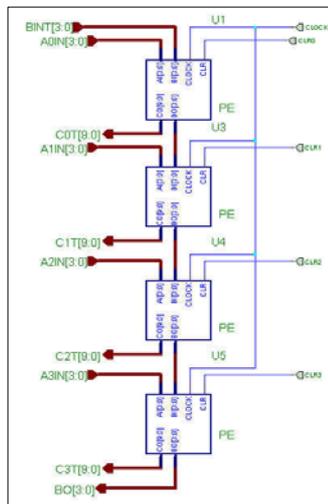


(c) PE₀

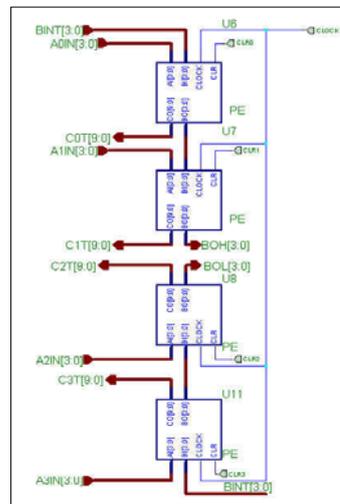
(d) 그림 6

<그림 9> 제안된 시스틀릭 어레이 프로세서의 PE

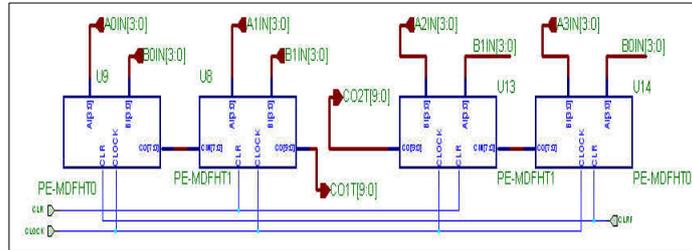
IV. 실험 및 고찰



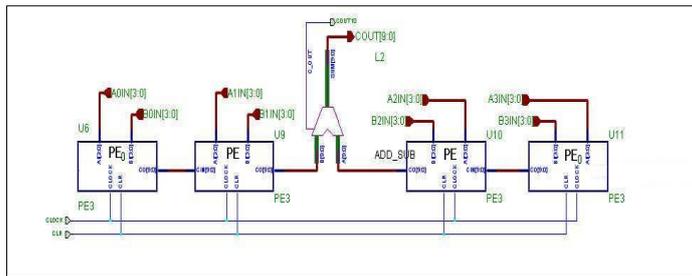
(a) 기본(그림 2.(b))의 설계



(b) 2분할 부분행렬(그림 4)의 설계

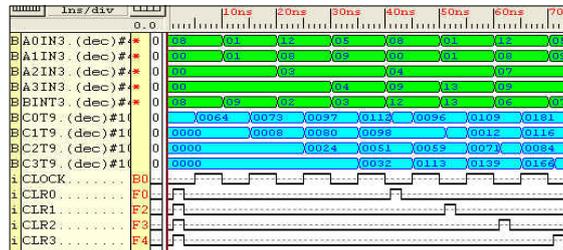


(c) 4분할 부분행렬(그림 6)의 설계

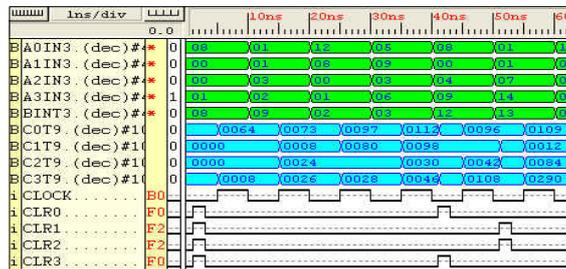


(d) 8분할 부분행렬(그림 8)의 설계

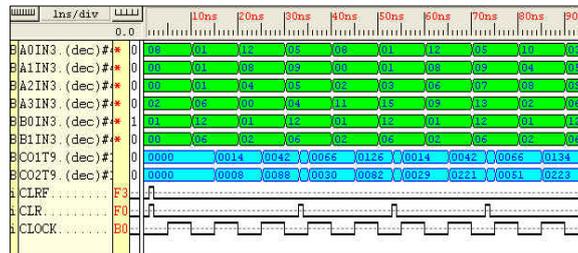
<그림 9> 시뮬레이션에 사용한 행렬·벡터 연산용 1차원 시스틀릭 어레이 프로세서의 구성도



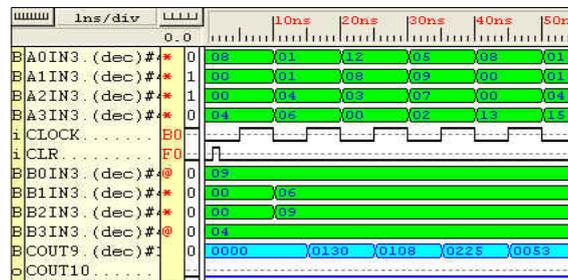
(a) 기본(그림 2.(b))의 설계



(b) 2분할 부분행렬(그림 4)의 설계



(c) 4분할 부분행렬(그림 6)의 설계



(d) 8분할 부분행렬(그림 8)의 설계

<그림 10> 제안된 행렬·벡터 연산용 1차원 시스틀릭 어레이 프로세서의 입출력 신호

본 논문에서 제안한 부분행렬 분할을 사용한 행렬·벡터 연산용 1차원 시스틀릭 어레이 프로세서에 대한 검증은 HLL과 Xilinx FPGA용 논리회로 시뮬레이터를 사용하였다. 그림 9에 시뮬레이션에 사용한 행렬·벡터 연산용 1차원 시스틀릭 어레이 프로세서의 구성도를 나타내었다. 그림 9의 (a)와 (b)에서 ArIN (r=0,...,3)신호는 행렬 a_{ij} ($i=r+1, j=1,..,4$)의 입력신호로 사용하고, BrIN(r=0,...,3)는 벡터 b_j ($j=r+1$)의 입력 신호로 사용하며, CrT(r=0,...,3)은 c_i ($i=r+1$) 출력 결과를 표시한다. 그림 9의 (c)에서 ArIN(r=0,...,3), BrIN(r=0,...,3)에는 그림 6의 데이터 순서에 따라 입력되고, CO1T와 CO2T는 연산 결과 신호이며 그림 6의 두 번째와 세 번째 PE의 출력신호이고, 그림 8에서는 COOUT이 결과 출력 신호를 표시한다.

그림 10의 (a)는 4클럭 이후 매 클럭마다 출력이 생성되며, 그림 10. (b)의 경우는 4번째 클럭에서 CO1T와 CO2T에서 동시에 출력이 생성되고 다음 클럭에서 CO1T와 CO2T에서 결과가 동시에 출력됨을 나타낸다, 그림 10. (c)에서는 CO0T CO1T에서 3번째 클럭에서 동시에 결과가 출력되고 이후 2클럭마다 동시에 결과가 출력되는 것을 알 수 있다. 그림 10. (d)에서는 2번째 클럭 이후 매 클럭마다 결과가 출력됨을 볼 수 있다.

승산기의 연산 시간을 t_{mul} , 가산기의 연산시간을 t_{add} 라고 하는 경우, 1개 연산기의 평균 연산 시간을 $t_{op} = \max \{ t_{add}, t_{mul} \}$ 라 하면 기본처리요소의 연산 시간은 PE의 최대 소요시간에 동기가 맞추어지므로 $t_{pe} = 2t_{op}$ 가 된다. n=4인 경우 행렬·벡터 연산결과는 4개가 발생되는데, 이러한 연산을 반복 수행하는 경우의 시간과 초기지연시간 및 연산기 크기의 비교를 표 1.에

나타내었다.

<표 1> 제안된 행렬·벡터 연산용 1차원 시스틀릭 어레이 프로세서의 연산시간 및 연산기 크기 비교(p : 반복연산 회수, n=4)

종류	비교항목	연산시간(tpe)		연산기 크기			동시 출력
		연속 연산	초기 지연	×	+	Rg	
그림 2. (a) 기본형		3+4p	3	4	3	4	-
그림 2. (b) 기본형		3+4p	3	4	4	8	-
그림 4. 2분할		1+4p	3	4	4	8	가능
그림 6. 4분할		1+4p	2	4	4	4	가능
그림 8. 8분할		1+4p	1	4	3	4	-

연속연산의 회수가 작은 경우초기 지연연산시간은 전체 연산시간에 큰 영향을 미치게 되는데 그림 8의 8분할 한 경우가 가장 적음을 알 수 있고, 그림 6과8의 경우 가산기 수가 증가할 것으로 예상되지만 III의 3절과 그림 9의 (c)에 나타낸 것과 같이 초기 사용되는 PE₀에는 가산기 요구되지 않으므로 그림 2의(a)와 그림 8의 8분할한 경우가 연산기의 크기가 가장 작음을 알 수 있다. 그러므로 초기지연 시간을 최소로 하는 경우는 8분할 부분행렬을 사용한 경우가 가장 좋으며, 동시 출력을 고려하는 경우는 초기연산 시간을 고려할 때 그림 6의 4분할한 부분행렬을 사용한 경우가 효율적임을 알 수 있다.

V. 결론

시스틀릭 어레이 프로세서는 병렬성, 파이프라인 처리, 규칙적 설계의 용이성등으로 디지털 신호처리, 그래픽, 신경망 등의 전용프로세서 설계에 자주 사용되며, 특히 1차원 행렬·벡터 연산용 시스틀릭 어레이 프로세서는 적용성 및 효율성이 높으므로 이용도가 높다. 본 논문에서는 행렬·벡터 연산용 시스틀릭 어레이 프로세서의 효율성을 높이기 위하여 연산 초기 지연시간의 감소와 동시 출력이 가능하도록 하였으며, 이를 수행하기 위하여 연산에 사용되는 행렬을 부분행렬로 분할하였으며, 2분할, 4분할 8분할 행렬을 적용하여 시스틀릭 어레이로 설계를 하였다. 기존의 시스틀릭 어레이 프로세서와 비교하였을 때 초기 연산지연 시간 및 연산기의 크기에 있어서 8분할 경우가 가장 우수하였으며, 동시 출력을 고려한 경우는 4분할 경우가 우수함을 알 수 있었다. 본 논문에서 제안한 시스틀릭 어레이 프로세서는 1차원 행렬·벡터 연산

에 한정되므로 행렬·행렬 연산 등 복잡도가 증가하는 경우에도 효율성을 높일 수 있도록 계속 연구가 되어야 할 것이다.

참고문헌

- [1] 이종석, 이승주, 김용성, 윤현식, 조원경 저, “Group이론을 이용한 영상처리용 어레이 프로세서 설계에 관한 연구”, 『대한전자공학회 학술대회 논문집』, 제12권 제2호, 대한전자공학회, 1989, pp.545-549.
- [2] 김운제, 김용성, 김은원, 조원경 저, “1차원 Bit-serial 시스틀릭 Wld 프로세서의 설계에 관한 연구”, 『대한전자공학회 학술대회 논문집』, 제13권 제2호, 대한전자공학회, 1990, pp.182-186.
- [3] J.A.K.S. Jayassinghe, F.Moelaert Ei-Hadidy, G. Karagiannis, Otto E. Herrmann, and J. Smit, Two-Level Pipelined Systolic Array Graphics Engine, *IEEE Journal of Solid-State Circuits*, vol.26, no.3, 1991, pp.229-236.
- [4] 김남섭, 김용성, 윤현식, 조원경 저, “잉여수계를 이용한 영상처리용 2차원 Fir 디지털 필터의 구현에 관한 연구”, 『대한전자공학회 학술대회 논문집』, 제14권 제1호, 대한전자공학회, 1991, pp.180-183.
- [5] S.Y. Kung, *VLSI Array Processors*, Prentice-Hall, 1998.
- [6] 김용성, 조원경, “고속 곡선 근사화를 위한 1차원 시스틀릭 어레이 프로세서 설계에 관한 연구”, 『경희대학교 전자공학과 전자통신공학 연구논문집』, 제10권, 제1호, 1991, pp. 25-28.
- [7] 김용성, 조원경, “행렬·벡터 연산용 1-차원 시스틀릭 어레이 프로세서를 이용한 그래픽 가속기의 설계”, 『전자공학회논문지』, 제30권, B편, 제1호, 대한전자공학회, 1993, pp.1-9.
- [8] 이권철, 김용성, 조원경 저, “행렬벡터 연산용 어레이 프로세서를 이용한 벡터폰트 생성기 설계에 관한 연구”, 『대한전자공학회 학술대회 논문집』, 제16권 제1호, 대한전자공학회, 1993, pp.182-185.
- [9] 김용성, 조원경, “1차원 시스틀릭 어레이 프로세서를 이용한 고속 곡선 발생기에 관한 연구”, 『전자공학회논문지』, 제31권, B편, 제5호, 대한전자공학회, 1994, pp.1-11.
- [10] 김용성, “재귀적 입력을 사용한 고속 베지어 곡선 발생기에 관한 연구”, 『여주대학 논문집 산업기술연구소편』, 제31권, B편, 제5호, 1998, pp.1-11.
- [11] 윤현식, 조원경 저, “잉여수계를 이용한 디지털 신경회로망의 실현”, 『전자공학회논문지』, 제30권 B편 제2호, 대한전자공학회, 1993, pp.142-148.