The Journal of Natural Science,
Pai Chai University, Korea
Vol.8 No2: 69~75, 1996

다차원 디지탈 필터의 VLSI 구조

정재길, 김용호
배재대학교 전자공학과

A VLSI architecture for the multi-dimensional digital filter

Jae-Gil Jeong, Yong-Hoh Kim
Department of Electronics Engineering, Pai Chai University

다차원 디지탈 필터링 알고리즘을 실시간에 처리할 수 있는 효율적인 시스템 구현을 위한 VLSI 구조를 제안하였다. 다차원 디지탈 필터링 알고리즘에 내재된 병렬성을 상태공간식을 이용하여 추출하여, 이를 프로세서 설계에 사용함으로써 보다 효율적인 다차원 디지탈 필터의 실시간 구현을 가능하게 하였다.

This paper presents a VLSI architecture for the efficient implementation of the real time multi-dimensional digital filter. The computational primitive for the filter is obtained from the state space representation of the multi-dimensional general order filter. The computational primitive is used for the data path of the processor.

Key words : VLSI architecture, digital filter

I. 소 개

다차원 (multi-dimensional) 디지탈 필터 (digital filter)는 로보틱스 (robotics), 레이다, 소나 (sonar), 토모그라피 (tomography), 원격회의 (teleconferencing), 고선명 TV (high definition television), 원격감지 (remote sensing), 지구물리학 등 많은 분야에 응용될 수 있다. 대부분의 이들 응용분야에서는 매우 큰 다차원 데이타의 실시간 처리를 필요로 한다. 이를 위해서는 빠른 계산과 많은 데이타의 저장 및 전달이 필요하다. 또한, 향상된 성능 (performance), 정교함 (sophistication), 실시간 (real-time) 신호처리에 대한 요구는 계속 확대되고 있다.

종래의 von Neumann 컴퓨터 구조에 근거한 단일 프로세서 시스템에서 이들 알고리즘을 구현하기에는 프로세서, 메모리, 입출력 장치 사이에 필요로 되는 많은 데이타 전달로 인하여, 많은 제약과 비효율성이 따른다.[1] 따라서, 입출력 속도 또는 프로세서의 계산속도에 의해 그 시스템의 성능이 제한된다.

최근의 VLSI(Very Large Scale Integrated Circuit) 기술의 발전은 디지탈 필터링 시스템의 구현을 위한 방법에 큰 변화를 가져오고 있다. VLSI 기술이 낮은 가격에 거의 무한의 하드웨어 (hardware)를 제공함으로써, 여러개의 프로세서 (또는 functional unit)를 한개의 VLSI 소자에 구현할 수 있게 되었을 뿐만 아니라, 논리회로와 메모리회로를 한개의 소자에 함께 수용할 수 있게 되고, 집적도가 높아지면서 전체시스템을 한개의 VLSI 소자에 내장하는 것이 가능하게 되어 가고 있다. 이러한 발전은 디지탈 신호처리 시스템의 구현에 있어서 알고리즘에 내재된 병렬성을

이용하여 그 성능을 크게 향상시킬 수 있는 가능성을 제시하여 주고 있다.

이러한 VLSI 기술을 최대로 활용하기 위해서는 두분야에서 연구가 이루어져야한다. 하나는 효율적인 다중 프로세서 시스템의 구조에 관한 연구이고, 다른 하나는 다중 프로세서 시스템내의 프로세서들을 효율적으로 활용하기 위한 병렬처리 알고리즘에 관한 연구이다. 단일 프로세서 시스템에서 우수한 성능을 보이는 알고리즘이 다중 프로세서 시스템에서 그다지 높은 성능을 보여주지 못하는 경우는 아주 흔하다. 이는 단일 프로세서 시스템에서와는 달리 다중 프로세서 시스템에서는 프로세서간의 데이타 전달 등의 단일 프로세서 시스템에서는 필요없는 동작이 필요하기 때문이다. 따라서, 이미 널리 알려진 알고리즘의 경우도 다중 프로세서 시스템을 효율적으로 이용하여 그 성능을 향상 시키기 위해서는 기존 알고리즘의 분해방법을 연구하여 여러 프로세서간에 일이 균등하게 배당되고 또한 프로세서간의 데이타 전달을 최소화하여야 한다.

따라서, 발전된 VLSI기술을 이용하기 위한 알고리즘 연구는 기본적으로 일의 분배에 관한 것이다. 즉, 프로세서들 사이에 일을 적절히 배당함으로써 프로세서간의 데이타 전달량 및 속도를 최소화하고, 모든 프로세서에 같은량의 일을 배당하여 시스템의 성능을 최대화하는 것이다. 이에는 크게 두가지 방법, 즉 데이타 분할 (data partitioning) 방법과 알고리즘 분할 (algorithm partitioning) 방법이 사용된다. 데이타 분할 방법이란 데이타를 여러 조각으로 나누어 각각의 프로세서에 균등하게 배당하는 것이고, 알고리즘 분할이란 알고리즘을 여러 조각으로 나누어 각각의 프로세서에 적절히 배당하는 것이다.

본 논문에서는 효율적인 실시간 다차원 디지탈 필터의 구현을 위한 VLSI 구조를 제안함으로서, 실시간 다차원 디지탈 필터의 효율적인 VLSI 구현에 도움을 주고자 한다. VLSI 시스템은 종래의 시스템과는 그 설계조건이 상당히 다르므로, 그 특성을 최대한 살려 효율적인 시스템의 구현을 위해 알고리즘 개발 단계부터 효율적인 병렬 처리를 고려하여야 한다.

Ⅱ. 알고리즘 (Algorithm)

다차원 디지탈 필터링 알고리즘의 병렬성을 추출하기 위하여 상태공간함수를 도구로 사용하였는데, 이 방법은 주어진 알고리즘에 대해서 계산량의 증가없이 데이타 전달량을 최소화 시킬 수 있는 가능성을 가지고 있다. 상태공간함수를 이용하면 주어진 알고리즘에 대하여 여러가지 계산구조를 구할 수 있다 [3]. 이를 이용하면 간략화된 계산 및 데이타 전달 구조, 파라미터 변화에 대한 감소된 sensitivity, 향상된 finite word length arithmetic operations [4]을 얻을 수 있다. 일반적으로 계산량, 계산의 규칙성(regularity), 계산의 정확도사이에는 trade off가 존재하므로, 본 논문에서는 VLSI 시스템에서 성능(throughput)에 가장 영향을 미치는 계산의 규칙성에 중점을 두었다. 이는 VLSI의 설계시간 단축의 중요한 요소이다. 또한 프로세서간의 데이타 전달의 규칙화 및 국부화(localize)에 중점을 두었는데 이는 시스템의 확장성에 중요한 요소가 된다.

첫번째 섹션을 지원하는 (2차원 시스템의 first quater support와 동등) 다차원 DLSI(Discrete Linear Shift-Invariant) 필터에 대한 일반적인 multivariable difference equation은 다음과 같다.[21]

$$g(\boldsymbol{n}) = \sum_{j_1=0}^{L_1} \cdots \sum_{j_k=0}^{L_k} a(\boldsymbol{J})\, f(n_1-j_1,\ \cdots,\ n_k-j_k) - \sum_{\substack{j_1=0 \\ j_1+j_2+\cdots+j_k>0}}^{L_1} \cdots \sum_{j_k=0}^{L_k} b(\boldsymbol{J})\, g(n_1-j_1,\ \cdots,\ n_k-j_k) \tag{1}$$

이 식에서 $a(\boldsymbol{J})$와 $b(\boldsymbol{J})$는 필터의 특성을 결정하는 상수이고, $\boldsymbol{n}$은 $n_1, n_2, \cdots, n_k$를, $\boldsymbol{J}$는 $j_1, j_2, \cdots, j_k$를 의미한다. $f(\boldsymbol{n})$은 각각의 독립변수에 대하여 일정한 간격으로 샘플된 입력 데이타이고, $g(\boldsymbol{n})$은 각각의 입력 샘플에 해당하는 값을 갖는 출력이다. 필터의 상수는 적절한 임의의 값을 가질 수 있기 때문에, 위의 식은 모든 다차원 디지탈 필터를 나타내는 식이 된다. 식 (1)에 상응하는 다차원 필터의 전달함수 (transfer function)는 다음과 같이 주어진다.

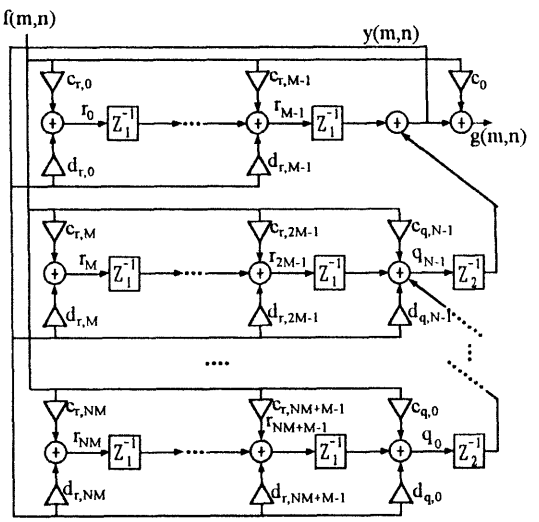
$$H(Z) = \frac{\sum_{j_1=0}^{L_1} \sum_{j_2=0}^{L_2} \cdots \sum_{j_k=0}^{L_k} a(\boldsymbol{J})\, z_1^{-j_1} z_2^{-j_2} \cdots z_k^{-j_k}}{\sum_{\substack{j_1=0 \\ j_1+j_2+\cdots+j_k>0}}^{L_1} \sum_{j_2=0}^{L_2} \cdots \sum_{j_k=0}^{L_k} b(\boldsymbol{J})\, z_1^{-j_1} z_2^{-j_2} \cdots z_k^{-j_k}} \tag{2}$$

$H(Z)$는 입력의 z-변환인 $F(Z)$와 출력의 z-변환인 $G(Z)$와의 관계를 나타낸다. $F(Z)$와 $G(Z)$의 관계는 다음과 같이 나타내어 질 수도 있다.

$$G(Z)=a(0)F(Z)+\sum_{j_1=0}^{L_1}\sum_{j_2=0}^{L_2}\cdots\sum_{j_k=0}^{L_k}{}_{j_1+j_2+\cdots+j_k>0}[a(J)F(Z)-b(J)G(Z)]Z^{-J} \tag{3}$$

이식에서 Z^{-J} 는 $z_1^{-j_1} z_2^{-j_2}\cdots z_k^{-j_k}$ 를 그리고 0는 0, 0, …, 0를 의미한다.

그림 1.은 식(1)의 2차원 식에 대한 블럭도이다. 이 그림에서 r 과 q는 상태변수(state variable)이다. 이 상태변수는 차후에 사용하기 위하여 저장되어야 하는 정보를 나타낸다. 따라서, 계산의 복잡성을 증가시키지 않고, 데이타의 전달양을 최소화할 수 있도록 상태를 선택하는 것은 매우 중요하다. 예를 들어, 영상처리에 응용하는 경우에 있어서, 수평상태 변수는 한개의 화소지연(pixel delay)을 나타내는 반면 수직상태변수는 화면의 한줄의 지연(line delay)을 나타낸다. 따라서, 하드웨어의 복잡성을 최소화 하기 위하여는 수직상태변수를 최소화하는 canonical form을 선택한다.



<그림 1> 2-D IIR 필터의 블록도

주어진 다차원 디지탈 필터의 상태공간식은 유일한 것이 아니다. 또한, 최소의 상태로 위식을 나타내는 방법도 현재까지 알려진 것이 없다.[22] 본 연구에서는 효율적인 다중 프로세서 시스템을 구현하기 위하여 사용될 수 있는 computational primitive를 유도하는 방법을 사용하였다. 이 방법은 충분히 범용이기(general) 때문에 블럭다이아그램(block diagram)이나 신호흐름그래프(signal flow graph) 형태로 표현된 알고리즘으로 부터 상태공간함수를 얻는데 사용될 수 있다. 따라서, 그림 1의 블럭다이아그램으로 부터 식 (1)에 해당하는 상태공간식을 구할 수 있다. 이러한 접근방법을 사용하여, 다차원 시스템에 대한 블럭다이아그램을 상태변수 index의 각 튜플에 한개 이하의 지연된 상태변수를 갖는 상태함수가 되도록 나눈다. 일반적인 N차원 디지탈 필터에 대한 식은 다음과 같이 주어진다.

$$\begin{aligned}q_{i,J}(\boldsymbol{n}) = {} & a(J)f(\boldsymbol{n})-b(J)g(\boldsymbol{n}) \\ & +q_{i,I-1}(\cdots,n_i-1,\cdots) \\ & +\sum_{\substack{j=1\\ j\neq i}}^{k} q_{j,M_j}(\cdots,n_j-1,\cdots)\end{aligned} \tag{4}$$

이때, 출력을 계산하는 식은 다음과 같다.

$$\begin{aligned}g(\boldsymbol{n}) = {} & a(0)f(\boldsymbol{n})+q_{1,L_1}(n_1-1,n_2,\cdots,n_k) \\ & +q_{2,L_2}(n_1,n_2-1,\cdots,n_k) \\ & +\cdots+q_{k,L_k}(n_1,n_2,\cdots,n_k-1)\end{aligned} \tag{5}$$

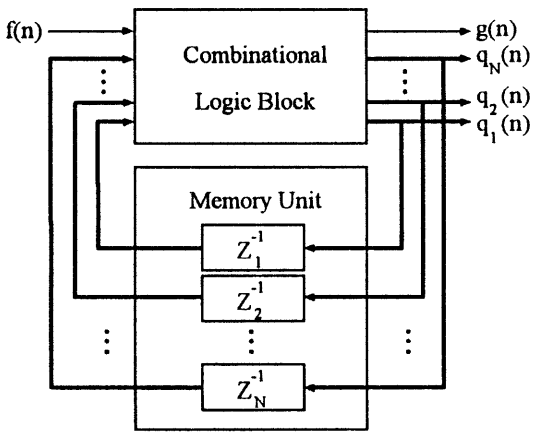
식을 간략화하기 위하여 새로운 변수 $y(\boldsymbol{n})$, c_{1J} 와 c_{2J} 를 다음과 같이 정의하였다.

$$\begin{aligned}y(\boldsymbol{n}) = {} & q_{1,L_1}(n_1-1,n_2,\cdots,n_k) \\ & +q_{2,L_2}(n_1,n_2-1,\cdots,n_k) \\ & +\cdots+q_{k,L_k}(n_1,n_2,\cdots,n_k-1) \\ c_{1J} = {} & a(J)-a(0)b(J) \\ c_{2J} = {} & -b(J)\end{aligned} \tag{6}$$

위에 정의한 값을 대입하면 대표적인 상태변수의 값은 다음과 같이 나타내어 진다.

$$\begin{aligned}q_{i,J}(\boldsymbol{n}) = {} & c_{1J}f(\boldsymbol{n})+c_{2J}y(\boldsymbol{n}) \\ & +q_{i,I-1}(\cdots,n_i-1,\cdots) \\ & +\sum_{\substack{j=1\\ j\neq i}}^{k} q_{j,M_j}(\cdots,n_j-1,\cdots)\end{aligned} \tag{7}$$

식(7)은 변수를 적절한 변수로 교환함으로써 N-D 디지탈 필터의 모든 상태변수나 출력계산에 이용될 수 있다. 예를 들면, 출력계산은 $c_{1J}=a(0)$, $c_{2J}=0$일때 식(7)로 부터 계산된다. 따라서, 식(7)을 N-D 디지탈 필터의 computational primitive로 정의할 수 있다. 또한 식(7)에서 $c_{2J}=0$이면 식(7)은 N-D FIR 디지탈 필터의 computational primitive가 된다. 따라서, 식(7)은 N-D 디지탈 필터의 범용 computational primitive이다. 식(7)에 주어진 N-D 상태공간함수에 대한 선형 finite state machine은 그림2와 같다.

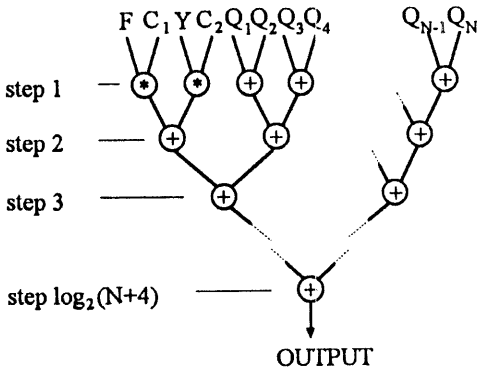


<그림 2> N-D 디지탈 필터의 finite-state 모델

State space model은 일반화된 finite state machine으로 나타낼 수 있으므로[2] 곱셈과 덧셈은 finite state machine의 조합논리회로 부분에 구현되고, 상태변수(state variable)는 memory units (delay elements)에 저장되어 진다. 이러한 개념을 이용하여, 한 알고리즘에 대한 state space model은 상응하는 finite state space machine으로 mapping 될 수 있다. State space model과 가능한 하드웨어 구조 사이에는 직접적인 상관 관계가 있다. N-D 디지탈 필터에 대한 일반화된 finite state machine model은 그림 2와 같다. 하드웨어 설계를 단순화 하기 위해서, finite state machine의 조합논리회로 부분에서 요구되는 계 computational primitive로 mapping한다.

이 computational primitive는 시스템에 사용될 특수 목적의 프로세서의 ALU (Arithmetic and Logic Unit) 설계에 사용된다. 다차원 디지탈 필터에 대한 computational primitive는 식(7)과 같다. 이 식을 이용하여 한 개의 상태변수나 출력을 계산하기 위해서는 두번의 곱셈과 최대 N+1번의 덧셈이 필요하다. Computational primitive를 구현하기 위하여 두개의 곱셈기와 N+1개의 덧셈기를 그림 3에 보여진 것과 같이 파이프라인 및 병렬처리할 수 있도록 트리구조를 이용하여 병렬처리할 수 있다.[5-9]

이렇게 구해진 coputational primitive는 바로 프로세서를 설계하는데 이용된다.



<그림 3> N-D 디지탈 필터의 병렬처리구조

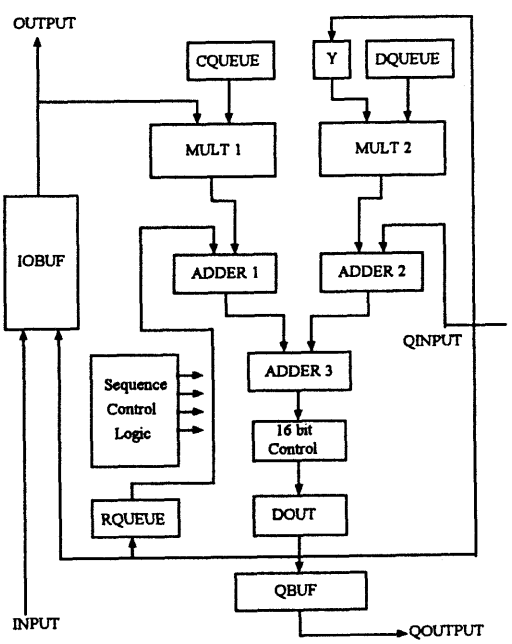
Ⅲ. 시스템 구현

디지탈 영상/비디오 처리를 위한 고성능 시스템의 설계에는 몇가지 방법이 사용될 수 있다. 가장 고전적인 방법은 한가지 알고리즘에 대하여 특수한 하드웨어를 개발하는 방법이다. 보다 최근의 방법으로는 systolic array,[10-15] wavefront array,[16-17] 프로그램어블 DSP, 또는 이 밖의 특수 목적용 프로세서를 이용한 Array processor와 같은 고성능 범용 구조를 이용하여 시스템을 구성하는 것이다.

디지탈 신호처리 알고리즘에서 데이타 전달은 일반적으로 규칙적이고 예측할 수 있다. Wavefront array 프로세서는 프로그램어블 프로세서로서 VLSI로 구현하기에 유리한 규칙성(regularity)과 지역성(locality)를 갖고 있을 뿐만 아니라, dataflow array[18,19] 프로세서의 data-driven특성을 갖고 있기 때문에, 데이타 전달이 규칙적이고 예측 기능한 알고리즘을 이용하는 영상/비디오 처리를 위한 다중 프로세서 구조의 좋은 개념적 모델이다. 본 논문에서 제안된 구조는 wavefront array 구조를 모델로 하여, 해당 알고

리즘의 computational primitive를 효율적이고 빠르게 계산할 수 있는 특수 목적용 (application-specific) 프로그램어블 프로세서로 구성된, 비동기식 array processor 시스템이다.

또한, 다중프로세서 시스템에서 실시간 다차원 디지탈 필터를 구현하기 위해서 시스템에서 사용되는 프로세서 설계시 다음과 같은 중요한 구조적 특성을 고려하였다. 프로세서는 식 (7)에서 주어진 computational primitive를 1 cycle에 수행하여야 한다. 이를 위해서는 병렬로 여러개의 계산을 수행하여야한다. 2-D 디지탈 필터의 경우, 2개의 곱셈과 3개의 덧셈, 각각의 funtional unit에 데이타의 공급, 상태변수의 저장, 데이타의 입출력, 주소의 계산 등을 병렬로 처리하여야 한다 (그림 2와 그림 3 참조). 단일 프로세서로는 실시간 처리 시스템의 구현이 불가능하므로 다중프로세서 시스템에서 효율적으로 사용되기 위하여 프로세서간의 통신에 있어서 고속 데이타 전달기능이 있어야 한다. 프로세서는 비동기적인 다중프로세서 시스템을 구현하기 위하여 적절한 handshaking 하드웨어가 필요하다. 여러가지 다른 application에 사용하기 위하여, 프로세서를 프로그램어블 하도록 하는 것이 바람직 하다. 예를 들면, 시스템의 차수 (order), 시스템 coefficients, 입력 데이타의 크기 또는 속도를 변형할 수 있어야 한다.



<그림 4> 2-D 디지탈 필터를 위한 프로세서 구조

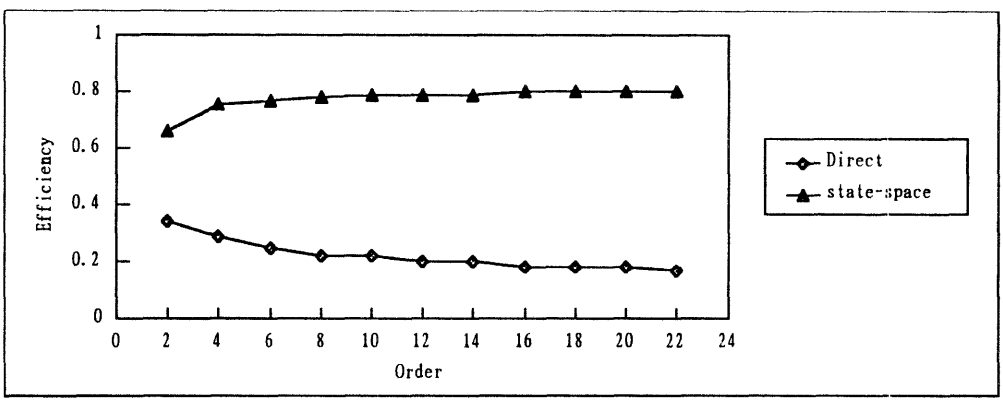
그림4. 는 2-D 디지탈 필터를 위한 프로세서의 block diagram이다. 상태변수 (state variable) 나 시스템 coefficient를 저장하는 queue의 주소는 자동적으로 발생되고, 데이타는 이들 queue로 부터 매 cycle마다 해당되는 레지스터로 보내진다. 이러한 특성은 주소계산과 데이타 전달에 따른 overhead가 전혀 없기 때문에 프로세서의 computational cycle time을 크게 줄일 수 있다. 프로세서의 arithmetic block은 2개의 곱셈기와 3개의 덧셈기를 갖고 있다. 입출력 데이타의 저장을 위한 static RAM, 2개의 가변장(variable length) coefficient queue, 상태변수 저장을 위한 register file과 여러개의 레지스터로 구성되어 있다. Sequence control block은 clock 발생기와 몇개의 synchronous counter 및 기타 조합회로로 구성된다.

Ⅳ. 결론 및 분석

T_1을 단일 프로세서를 가지고 식(1)을 계산하기 위한 사이클의 수라고하면, N차원 L차 IIR 디지탈 필터를 처리하기 위해서는 $2\cdot(N+1)^L-1$ 의 곱하기와 $2\cdot(N+1)^L-2$ 의 더하기가 필요하다. 따라서 $T_1=4\cdot(N+1)^L-3$ 이 된다. 또한, P를 프로세서의 수라 하고, T_P를 프로세서 P개를 사용하였을 경우 계산에 필요한 사이클의 수라 정의할 때, 가능한 최대의 speedup과 efficiency는 각각 $S_P=T_1/T_P$와 $E_P=S_P/P$로 정의된다. 표 1은 직접구현 (Direct implementation) 방식과 제안된 방식(State-space)의 성능 비교표이다. 이때 직접구현방식의 경우 1번의 곱하기와 1번의 누적(accumulation)을 1 사이클에 계산할 수 있는 범용의 DSP를 이용하고, 제안된 방식은 그림 4에 예시된 프로세서를 이용하였다고 가정하여 산출한 값이다. 또한 그림 5 는 2차원 IIR 필터를 제안된 방식과 직접구현 방식으로 구현하였을 때의 효율을 차수에 따라 도표로 나타낸 것이다. 도표에서 보는 바와 같이 직접구현 방식의 효율은 필터의 차수가 증가함에 따라 감소하지만, 제안된 방식의 효율은 증가한다. 따라서 제안된 방식이 필터의 차수가 증가함에 따라 직접구현방식에 비하여 적은 수의 프로세서를 가

<표 1> 1차원 및 2차원 디지탈 IIR 필터의 성능비교 (N : 필터의 차수)

	1차원의 경우		2차원의 경우	
	direct	state-space	direct	state-space
T_1	$4N+1$	$4N+1$	$4(N+1)^2-3$	$4(N+1)^2-3$
P	$2N+1$	4	$2(N+1)^2-2$	5
T_P	$\lceil \log_2(2N+1) \rceil +1$	$N+1$	$\lceil \log_2((2N+1)^2-1) \rceil +1$	$(N+1)^2+1$
S_P	$\frac{4N+1}{\lceil \log_2(2N+1) \rceil +1}$	4	$\frac{4(N+1)^2-3}{\lceil \log_2((2N+1)^2-1) \rceil +1}$	4
E_P	$\frac{2}{\lceil \log_2(2N+1) \rceil +1}$	1.0	$\frac{2}{\lceil \log_2((2N+1)^2-1) \rceil +1}$	0.8



<그림 5> 2차원 IIR 필터의 차수에 따른 효율

지고 실시간 구현이 가능하다.

다차원 데이타, 특히 디지탈 영상 (Image) 또는 비디오 (Video) 신호의 실시간 처리를 요구하는 Application의 증대에 따라, 이를 수행할 수 있는 시스템의 개발 경쟁은 매우 치열하다. 기존 의 상용 프로세서가 충분한 성능을 제공하지 못하고, 또한 이를 이용한 다중 프로세서 시스템의 비효율성으로 인하여, 개발기간 및 비용이 상대적으로 많이 소요되는 주문형 VLSI에 의존하고 있다.

본논문에서는 적용될 알고리즘으로 부터 내재된 병렬성을 알고리즘분해기법을 이용하여 추출하고, 분해된 알고리즘의 효율적인 병렬처리를 위한 다중프로세서 시스템구조 및 이를 위한 프로세서의 구조를 제안함으로써, 실시간 디지탈 신호처리 시스템의 효율성을 크게 증진 시킬 수 있었다.

제안된 프로세서 및 다중프로세서 구조는 여러가지 목적으로 사용될 수 있다. 따라서, 한 번의 설계로 여러 Application들에 적용될 수 있어 설계 및 개발 기간을 낮출 수 있고, 다량 공급에 따라 생산 원가를 낮출 수 있어, 다양한 용도의 디지탈 실시간 신호 처리 시스템의 저가 구현을 가능케 한다.

감사의 글

본 논문은 95년도 배재대학교 교내학술연구비 지원에 의하여 수행된 연구의 일부로 이에 감사를 드립니다.

참 고 문 헌

1. A. V. Kulkami and D. W. L. Yen, "Systolic processing and implementation for signal and image processing," *IEEE Trans. Comput.*, vol. C-31, no. 10, pp. 1000-1009, Oct. 1982.
2.. Z. Kohavi, *Switching and Finite Automata Theory*, New York, N. Y.: McGraw Hill Book Co.,1978.
3. M. Y. Dabbagh and W. E. Alexander, "Multiprocessor implementation of 2-D denominator-separable digital filters for real-time processing," *IEEE Trans. Circuits Syst.*, vol. CAS-37, no.6, pp.872- 881, 1989.
4. W. W. Edmonson and W. E. Alexander, "Error analysis of a high throughput state-space model for digital signal processing," in *Proc. of the IEEE pacific RIM conf. on Communications, Computers, and Signal Processing*, June, 1987.
5. J. H. Kim and W. E. Alexander, "A multiprocessor architecture for spatial domain digital filters," *IEEE Trans. Comput.*, vol. C-36, no. 7, pp. 876-884, 1987.
6. J. G. Jeong and W. E. Alexander, "The efficient real-time spatial domain 2-D IIR and FIR digital filter implementation," in *Proc. of IEEE 23rd SSST*, pp. 394-398, Mar. 1991.
7. S. M. Park, et al., "A novel VLSI architecture for the real-time implementation of 2-D dignal processing systems", in *Proc. of IEEE ICCD*, Oct. 1988.
8. J. G. Jeong, "A multiprocessor implementation of the real-time digital filter", in *Proc. of '92JTC-CSCC*, Jul. 1992.
9. J. G. Jeong, H. Y. Xu, and W. E. Alexander, "A block dataflow architectures for digital signal processing", *U.S. Patent Application* No. 071837314, Mar. 1992.
10. H. T. Kung and C. E. Leiserson, "Sysstolic arrays (for VLSI)," in *Proc. IEEE Sparce matrix*, pp.256-282, 1978.
11. D. I. Moldovan, "On the design algo- rithm for VLSI systolic arrays," *Proce- edings of IEEE*, pp. 113-120, Jan. 1983.
12. A. L. Fisher and H. T. Kung, "Synchronizing large VLSI processor array," *IEEE Trans. Comput.*, vol. C-34, no. 8, pp. 734-740, 1985.
13. T. Aboulnasr and W. Steenaart, "Realtime systolic array processor for 2-D spatial filtering," *IEEE Trans. Circuits Syst.*, vol. CAS-35, no. 4, pp. 451-455, Apr. 1988.
14. M. A. Sid-Ahmed, "A systolic realization for 2-D digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-37, no. 4, pp. 560-565, Apr. 1989.
15. H. K. Kwan and T. S. Okullo-Oballa, "2-D systolic arrays for realization of 2-D convolution," *IEEE Trans. Circuits Syst.*, vol. CAS-37, no. 2, pp. 267-763, Feb. 1990.
16. S. Y. Kung, "VLSI signal processing from transversal filtering to concurrent array processing," *VLSI and Modern Signal Processing*, pp.127-152, 1985
17. S. Y. Kung, K. S. Arun, R.J. Gal-Ezer, and D. V. Bhaskar Rao, "Wavefront array processor: language, architecture, and applications," *IEEE Trans. Comput.*, vol. C-31, no. 11, pp.1054-1066, 1982.
18. J. B. Dennis, "Dataflow supercomputer," *IEEE Computer*, pp. 48-56, November, 1980
19. K. P. Gostelow and R. E. Thomas, "Performance of a simulated data flow computer," *IEEE Trans. Comput.*, vol. C-29, no. 10, pp. 905-919, 1980.
20. K. T. Johnson, A.R. Hurson, and B. Shirazi, "General-purpose systolic arrays," *IEEE Computer*, vol.26, no. 11, pp. 20-31, November, 1993.
21. D. Dudgeon and R. Mersereau, *Multidimensional Digital Signal Processing*. New Jersey: Prentice-Hall Inc., 1984.
22. S. Y. Kung, B. C. Levy, M. Morf, and T. Kailath, "New results in 2-D systems theory, Part II: 2-D state space model," *Proc. IEEE*, vol. 65, pp.945-961, June 1977.