

영상처리 하드웨어

朴東宣*, 李俊煥**

* 全北大學校 情報通信工學科, ** 全北大學校 電子工學

1. 서론

다양한 정보를 제공하는 영상신호는 텔레비전과 같은 방송으로부터 패턴인식, 로봇트비전 등의 제 분야에서 응용되고 있다. 응용목적에 따라 영상신호는 간단한 영상처리(Low-level Image Processing)에서 고수준의 복잡한 컴퓨터 시각(Computer Vision)에 이르는 방법들로 처리된다. 초기 영상처리(Early Processing)는 필터링, 영역분할, 특징추출, 변환에 의한 매핑, 광류검출 등과 같은 기술로서 디지털영상에 대한 국부적인 계산이 주종을 이룬다. 여기에서 추출된 영상정보들을 이용하여 상위 수준(High-level Image Processing)의 영상처리는 부호화, 형상인식 등을 달성하게 된다.^[1]

이차원 구조의 디지털영상신호는 데이터량이 매우 많아 실시간 처리를 필요로 하는 대부분의 응용에서는 병렬처리를 수반하여야 한다. 초기 영상처리 알고리즘들은 자체의 규칙적인 국부성을 이용하여 병렬처리가 가능하며 대규모의 병렬 하드웨어의 설계로 연결할 수 있는 반면에 상위수준의 알고리즘의 대부분은 영상신호 전반에 대한 정보를 필요로 하여 국부적인 병렬처리가 아닌 태스크 수준(Task-level)의 병렬화가 가능하다.

현대의 컴퓨터는 폰 노이만식의 간단한 구조(Von Neumann's Architecture)로부터 다수의 처리장치를 포함하는 멀티프로세서 시스템에 이르기 까지 다양하게 분류할 수 있다. 단일 프로세서에 의존하는 폰 노이만 컴퓨터들은 구조상 영상신호의 많은 데이터를 처리하기에는 적합하지 못하며 영상신호의 처리는 알고리즘에 따라 변화하나 대개 국부적인 계산이

많아 병렬 하드웨어에 쉽게 접목할 수 있다. 영상처리용 하드웨어는 여러 목적에 사용 가능한 범용하드웨어(General-purpose Hardware)와 특정한 용도만을 지닌 전용하드웨어(Dedicated Hardware)로 구분할 수 있다.^[2] 일반적인 컴퓨터 구조와 더불어 발전해온 영상처리용 하드웨어는 파이프라이닝(Pipelining), 어레이 프로세서(Array Processor), 다중프로세서(Multiprocessor) 등과 같은 병렬처리 구조를 포함한다. 이들 병렬 하드웨어들은 연결 토폴로지에 따라 특정 알고리즘에 대한 성능이 달라지게 된다. 또한 병렬 하드웨어들은 인터커넥션 네트워크를 구성하며 처리 알고리즘의 요구사항이 달라질 때마다 재구성이 가능하게 설계된 시스템(Reconfigurable System)과 고정된 연결을 가진 시스템(Fixed-Connection System)으로 분류가 가능하다.

근래의 초고집적회로 기술의 발전에 따라 파이프라인 처리와 어레이프로세서 구조를 함께 도입하고 입출력을 감소시킬 수 있는 구조를 지닌 시스톱릭 어레이 프로세서(Systolic Array Processor)와 웨이브프런트 어레이 프로세서(Wavefront Array Processor)가 고속의 초기 영상처리 하드웨어로 대두되고 있다.^[3,4] 한정된 수의 주변 처리장치(Processing Element)들과 연결을 가지며 한번 입력된 데이터는 계속적인 처리를 하도록 알고리즘에 따라 설계된 이 하드웨어들은 최근 영상통신의 표준화 노력에 맞추어 빠르게 개발되고 있다.

본 논문에서는 영상처리 알고리즘의 병렬화에 대하여 2장에서 간단히 설명하고 3장에서는 주로 범용 영상처리 하드웨어 구조에 대하여 설명하였다. 특정한 영상처리 알고리즘을 위한 전용하드웨어를 4장에서 수리 형태학적 연산자 하드웨어 설계 예를 이용하여 기술하였으며 5장이 결론이다.

Ⅱ. 영상 신호의 병렬 처리

디지털 영상 신호는 적용하는 알고리즘이 복잡하면 높은 데이터량 때문에 기존의 단순 구조를 지닌 컴퓨터를 위한 알고리즘을 사용할 경우 실시간 처리의 요구에 대한 부응이 어려워진다. 따라서 알고리즘의 병렬화가 실시간 처리를 위하여 필수적이며 영상 신호 처리의 병렬화는 크게 데이터 수준(Data-level)과 태스크 수준(Task-level)으로 나눌 수 있다. 초기 영상 처리는 대부분 국부적인 데이터에 대하여 작업을 수행하게 되어 데이터 수준의 대량의 병렬화가 가능하다. 그러나 상위수준의 컴퓨터 시각에서는 초기 영상처리에서 얻어진 정보들을 종합하여 이루어 지는 태스크 수준의 병렬화가 가능하다.

하위 수준의 초기 영상 처리는 기본적으로 규칙적이며 국부적인 특성을 가지고 있으며 전단에서 처리된 데이터를 반복 처리하는 특성을 지니고 있어서 대규모의 병렬 처리를 용이하게 한다. 하위 수준 영상 처리에는 여러 가지 종류의 알고리즘으로 분류할 수 있다. 그레이 준위 변환과 히스토그램 등화 등과 같이 하나 하나의 데이터에 적용되는 알고리즘이 있는가 하면 일정 크기의 윈도우를 사용하여 윈도우 내부의 데이터에 필터링과 같은 일을 수행하는 알고리즘도 있다. 또한 영상 데이터를 공간 영역에서 이용하기 쉬운 다른 영역으로 변환하는 알고리즘도 영상신호의 압축 등에서 사용되며 랭크오더 필터(Rank-order Filter)와 같이 정렬(Sorting)을 필요로 하는 알고리즘도 있다. 이러한 알고리즘들은 모두 영상 신호의 국부적인 정보를 이용하여 처리하므로 영상신호 전체에 대한 병렬화가 가능하게 된다. 초기 영상 신호처리는 또한 대부분의 경우 한번 처리된 데이터를 다시 다른 처리를 하거나 같은 처리를 반복한 경우가 많으며 이러한 경우에는 시간적 병렬화인 파이프라인 처리가 가능하다.

상위수준의 영상 처리에서는 초기 영상 처리 과정에서 얻어진 정보를 서로 혼합하여 이루어지는 알고리즘이 많아서 국부성이 약해지며 데이터간의 의존도가 많이 존재하고 이 경우 소수 큰 테두리의 태스크 수준 병렬화가 가능하다. 각 태스크들이 서로 다른 프로세서에서 처리 되도록 병렬화 알고리즘을 구성할 수 있다. 이와 같이 영상 신호처리는 여러 측면에서 병렬화가 가능하며 병렬화된 알고리즘들을 병렬 하드

웨어로 수행하게 되면 데이터량이 많은 디지털 영상도 고속으로 처리하게 된다.

Ⅲ. 영상 처리 하드웨어 구조

영상 신호 처리를 위한 병렬 하드웨어는 알고리즘에 따라 재구성이 가능한 구조와 특별한 처리 알고리즘에 적합하도록 구성된 구조의 전용 하드웨어로 나눌 수 있다. 특별 목적의 전용 하드웨어는 처리 시간이 빠르나 사용 범위가 좁으며 비용이 많이 든다. 반면에 재구성이 가능한 하드웨어는 여러 가지 알고리즘들에 적용이 가능하나 하드웨어 복잡도가 높아지고 처리 속도가 전용 하드웨어에 비교하여 상대적으로 떨어지게 된다. 하드웨어 복잡도는 병렬 처리 하드웨어간의 통신 방법 등에 따라 달라지며 처리 소자 간의 연결 방법에 따라 또한 변화된다. 본 장에서는 주로 범용의 재구성 가능한 하드웨어에 대하여 설명하고 4장에서 특수한 목적의 전용 하드웨어에 대하여 설명하였다.

범용의 영상 처리 하드웨어 구조는 컴퓨터 구조와 더불어 발전해 왔으며 많은 연구가 진행되어 왔다. 영상처리를 위하여 제안된 대부분의 하드웨어는 SIMD(Single Instruction-stream Multiple Data-stream) 또는 MIMD(Multiple Instruction-stream Multiple Data-stream)로 분류되는 구조를 지니고 있다. 다수의 데이터에 대하여 한가지 일을 하도록 구성된 SIMD계열의 구조는 국부성과 규칙성을 갖는 하위 수준의 영상처리 알고리즘, 즉 데이터 수준의 알고리즘의 수행에 적합하다. 그림 1에서는 일반적인 SIMD 계열 시스템의 구조를 보이고 있다. 그림에서 보듯이 이러한 시스템은 하나의 제어기와 다수의 처리장치(Processing Element)를 가지고 있으며 메모리도 다수의 메모리 유닛을 포함한다. 대부분의 설계에서는 근접해 있는 처리 장치들과 연결을 하여 하드웨어의 복잡성을 낮추도록 하고 있다. CLIP(Cellular Logic Image Processor)와 MPP(Massively Parallel Processor)와 같은 SIMD 계열의 어레이 프로세서가 만들어졌으며 초기 영상 처리의 빠른 수행을 위해 설계되었다.¹⁾ 한 예로서 그림 2에서는 CLIP의 처리 장치들 간의 연결을 보이고 있다. 이러한 SIMD계열의 구조는 하나의 동일한 일을

처리 장치들이 동시에 수행하는 에지검출이나 필터링과 같은 데이터 수준의 영상처리 알고리즘을 병렬로 처리하는 것에는 매우 적합하나 태스크 수준의 서로 다른 작업을 동시에 수행해야 하는 경우에는 하나의 제어기 밖에 없으므로 비효율적이다.

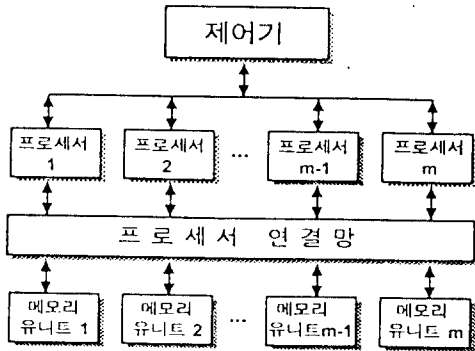
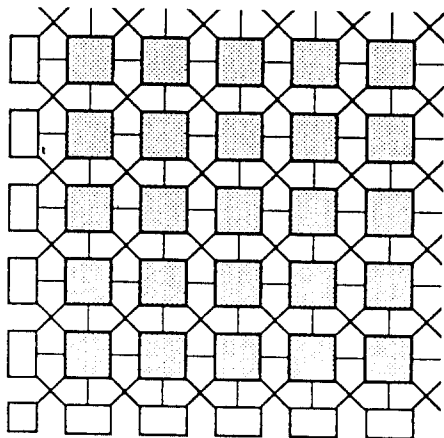


그림 1. 일반적인 SIMD 구조



□ : 처리소자

그림 2. SIMD 구조 처리소자의 연결 예

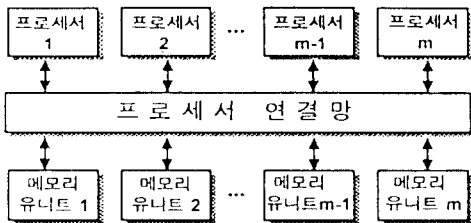


그림 3. 일반적인 MIMD의 구조

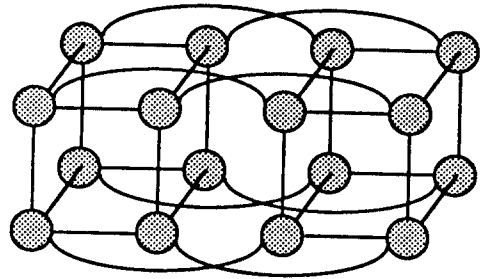


그림 4. 하이퍼 큐브 구조

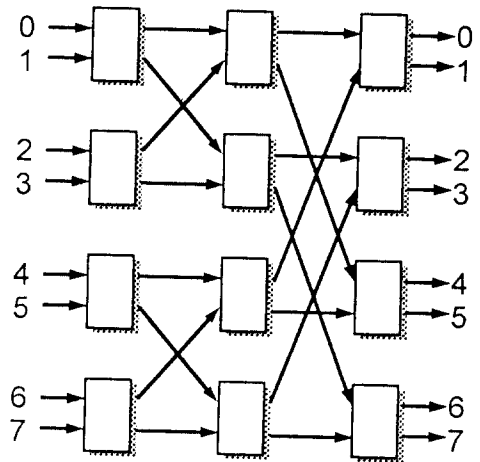


그림 5. 다단 연결망

MIMD 계열의 구조는 다수의 프로세서들을 포함하여 다양한 일을 각 프로세서에서 병렬 수행하도록 설계되었으며 그림 3에서 MIMD 시스템의 일반적인 구조를 보이고 있다.

각 프로세서들이 기존의 SISD(Single Instruction-stream Single Data-stream) 시스템의 역할을 하도록 설계된 MIMD 계열의 구조는 SIMD 계열의 구조에 비하여 훨씬 더 복잡하며 다수의 프로세서들과 공유 메모리 유니트들이 상호 연결망을 통하여 정보를 교환한다. 어떠한 연결 토폴로지를 사용하는가에 따라 다양한 형태의 MIMD 구조가 만들어지며 알고리즘의 효율성도 달라지게 된다. 그림 4,5에서는 최근에 많이 사용되는 하이퍼큐브(3-degree Hypercube)와 다단계망의 구조를 보이고 있다. 초기 영상처리의 알고리즘들은 동일한 일을 수행하므로 MIMD 계열의 시스템에서도 수행 가능하지만 SIMD

계열의 하드웨어가 대부분의 경우 보다 효율적이며 MIMD 구조는 상위수준 영상처리에서의 서로 다른 태스크를 각 프로세서에서 수행하는 형태인 태스크 수준의 병렬화 알고리즘에 적합하다.

많은 컴퓨터 시각의 문제들은 에지 검출이나 필터링과 같은 데이터 수준의 국부적인 계산을 필요로 하는 동시에 추출된 정보를 이용하여 태스크 수준의 서로 다른 심볼 처리 (Symbolic Processing)를 필요로 한다. 따라서 이 두 가지 다른 형태의 병렬화를 효율적으로 수용하기 위하여 SIMD 구조와 MIMD 구조를 포함하는 구조들이 제안되고 건설되고 있다. 이러한 혼합 구조는 트리(Tree)나 피라미드(Pyramid) 형태를 포함하며 대표적인 예로써 SPHINX의 피라미드 구조를 그림 6에서 보이고 있다.^[1] 그림에서 각 계층은 이차원 SIMD 형태의 구조이며 각 계층은 별도의 제어기로서 일을 수행할 수 있도록 고안되어 분류상 다중 SIMD구조(Multiple SIMD)의 시스템 형태를 이룬다.

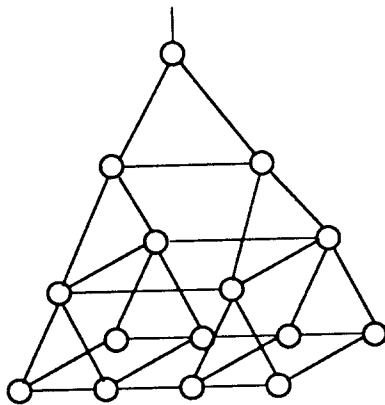


그림 6. SPHINX 피라미드 구조

최근의 초고집적회로의 발전은 시스틀릭 어레이나 웨이브프런트 어레이 프로세서를 영상처리 하드웨어로서 가능케 하고 있다. 이들 구조는 처리소자들을 어레이 프로세서의 형태로 구성하는 동시에 파이프라인 처리를 이용하여 시간적인 병렬화를 달성한다. 처리소자간에는 국부적인 연결만을 가지나 데이터를 일단 읽으면 파이프라인 처리로 계속 사용하여 입출력을 최소화 하도록 설계한다. 시스틀릭 어레이는 하나의 동기화된 처리를 수행하게 되어 파이프라인에서 시간이 가장 많이 걸리는 작업에 수행 성능이 좌우되게 된다. 이러한 단점을 보완한 웨이브프런트 어레이

에서는 종래의 제어 흐름(Controlflow)에 의하여 모든 것이 동기화되는 것을 지양하고 데이터 흐름(Data-flow)의 개념이 도입되었다. 데이터가 처리장치 사이에서 비동기적으로 전달되며 미리 처리장치의 국부 메모리에 저장된 명령어에 데이터가 유입하면 즉시 수행되는 방식을 이용하므로 시스틀릭 어레이의 단점이 보완된다. 그림 7이 시스틀릭 또는 웨이브프런트 어레이 구조의 예를 보이고 있다. 이러한 구조는 범용이 아닌 DCT 변환(Discrete Cosine Transform)과 같은 특수 목적의 고속 하드웨어들을 위하여 주로 설계되고 있다.

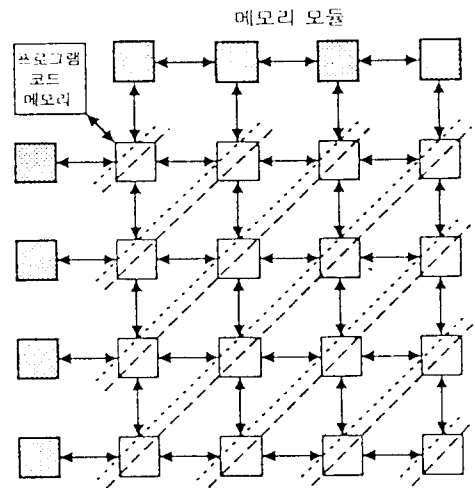


그림 7. 시스틀릭 어레이 구조

IV. 전용 영상처리 하드웨어

영상처리 전용 하드웨어는 특수한 목적에만 쓸 수 있도록 최적의 하드웨어 구조를 갖는 시스템을 의미한다. 이는 결과적으로 범용 하드웨어에 비해 속도가 빠르지만 응용 범위가 좁게 된다. 최근 초고집적회로의 발전으로 하드웨어 개발 비용이 급속히 하락하고 집적도가 높아짐에 따라 전용 하드웨어의 개발이 많아지고 있다. 특히 알고리즘의 특성이 뛰어나지만 소프트웨어로 구현시 시간이 많이 걸리는 알고리즘들을 위하여 빠른 연산 처리가 가능하도록 전용 하드웨어화가 이루어지고 있다. 예를 들면 수리 형태학 연산자들은 비선형적인 필터로서 우수한 특성을 지니나 정렬을 받

복적으로 필요로 하여 처리 시간이 많이 요구된다. 이 연산자들을 위한 전용 하드웨어가 건설되면 고속화되어 실시간 처리가 가능하고 비선형 필터의 우수성을 이용할 수 있게 된다. 따라서 이 장에서는 수학적 형태학의 병렬 하드웨어 설계에 관하여 설명하여 영상 처리 전용 하드웨어의 예로 들고자 한다.

1. 수리 형태학의 기본 개념

영상은 예지와 같은 영상 정보의 필수적인 요소뿐만 아니라 잡음 등 불필요한 요소도 포함하고 있다. 이러한 영상의 불필요한 요소를 제거시키고 인식하고자 할 때 흔히 이용되는 방법은 필터링이다. 필터링을 이용하여 영상을 처리하는 방식에는 선형 필터 방식과 비선형 필터 방식으로 나눌 수 있다. 비선형 필터 방식은 많은 처리 시간을 요구하나 영상 인식의 중요한 부분을 유지시키면서 잡음 등과 같은 불필요한 요소를 선택적으로 제거하는 장점을 가지고 있기 때문에 잡음 추출, 이미지 고양과 같은 간단한 영상 처리로부터 부호화, 의료 사진의 인식, 산업 현장에서의 오류검출 등 복잡한 컴퓨터 시각 문제 등에 적용되고 있다.⁶⁾

비선형 필터 중 수리 형태학(Mathematical Morphology)⁵⁾은 필요한 정보를 선택적으로 유지하고 불필요한 부분을 효과적으로 제거 하는 우수한 특성을 가지고 있다. 수리 형태학은 화소 단위로 동일한 연산을 반복적으로 수행하며 소프트웨어로 구현할 경우 처리 시간이 매우 길어지게 된다. 따라서 실시간 처리가 요구되는 응용 분야에서는 형태학의 우수한 특성을 활용하기 어렵게 된다. 그러나 동일한 연산을 모든 화소에 적용하는 형태학에서는 병렬 적인 하드웨어의 구현이 가능하며 처리 시간을 실시간 응용에 맞게 감소시킬 수 있다.

수리 형태학에서는 형태소(Structuring Element)라 불리는 작은 영상을 이용하여 원 영상에 대해서 다양한 연산 처리를 수행한다. 이 연산 처리에는 영상의 모든 부분을 확장 시켜주는 Dilation과 이와 반대로 축소 시켜주는 Erosion이 있다. 또한 이 두 가지 기본 연산 작용을 반복 적용함으로써 영상의 윤곽선을 완만히 해주고 길게 돌출된 부분과 좁은 지협을 제거시켜 주는 Opening이 있고 이와 반대로 작은 구멍을 제거하고 윤곽부분에 있어서는 겹을 채워주는 Closing이 있다.⁵⁾

A는 원 영상, B를 형태소 이진 영상이라 할 때 Dilation 연산은 식 (1)과 같이 표현된다.

$$A \oplus_b B = \{c \in E^N \mid c = a + b, a \in A \text{ and } b \in B\} \quad (1)$$

또한 Dilation 연산과 반대의 개념인 Erosion 연산은 식 (2)처럼 정의한다.

$$A \ominus_b B = \{X + b \in A, \forall b \in B\} \quad (2)$$

2-차원 유클리디언 공간에서 f는 원 영상, k는 형태소 영상이라 할 때, 그레이 준위 Dilation 연산은 영상의 화소 값과 형태소 화소의 값을 더한 후 그중 최대값을 찾는 연산이며 식 (3)처럼 정의된다.

$$(f \oplus_k k)(x, y) = \max_{(i, j)} \{f(x-i, y-j) + k(i, j)\} \quad (3)$$

Erosion 연산은 원 영상의 화소의 값에서 형태소 화소의 값을 빼고 난 후 최소값을 구하는 연산이며 식 (4)과 같다.

$$(f \ominus_k k)(x, y) = \min_{(i, j)} \{f(x+i, y+j) - k(i, j)\} \quad (4)$$

2. 수리 형태학의 단위 하드웨어 설계

실 시간 처리의 기본 요소가 될 연산 지연 시간을 감소시키기 위하여 이진 영상으로 재 구성하는 방법을 이용하여 설계하고 형태소의 크기에 영향을 받지 않는 하드웨어를 설계할 수 있는 방법에 대하여 전개한다.

이진 영상은 논리 게이트만을 이용하여 쉽게 연산자들을 구현할 수 있다. F는 2-차원 원 이진 영상, K는 형태소의 크기가 (N × N)인 영상인 경우에 Dilation과 Erosion 연산자들을 하드웨어로 표현하였다. 이진 영상에 대한 Dilation 연산과 Erosion 연산을 표현한 식 (5)과 (6)으로부터 그림 8과 그림 9와 같이 이진 영상의 수리 형태학의 하드웨어를 나타낼 수 있다.

$$(F \oplus_k K)_{(x, y)} = OR_{i, j} [F_{(x-i, y-j)} \cdot K_{(i, j)}] \quad (5)$$

$$(F \ominus_k K)_{(x, y)} = AND_{i, j} [F_{(x+i, y+j)} + \overline{K_{(i, j)}}] \quad (6)$$

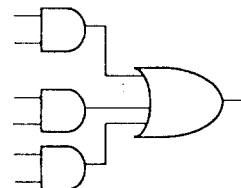


그림 8. Dilation 하드웨어

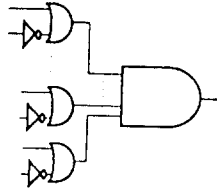


그림 9. Erosion 하드웨어

그레이 준위 영상의 하드웨어는 최대값 또는 최소값을 이용하여 구현하는 것보다 이진 영상을 이용하여 설계를 하는 것이 시간 지연과 제어의 복잡성을 줄일 수 있다. 그레이 준위 영상의 값을 Umbra를 취하여 이진 영상의 형태로 변환시킨 후 구현하기 간단한 이진 영상의 하드웨어를 이용하는 방법을 사용한다. 그레이 준위 영상에 적용하기 앞서 기본 용어를 정의하면 아래와 같다.

Slice 는 입력값이 y(그레이 준위)와 같고 값의 크기가 클 때 식 (7)로 나타낼 수 있다.

$$S[f_i]_{(x,y)} = \begin{cases} 1 & \text{if } y=i \text{ and } f(x) \geq y \\ 0 & \text{elsewhere} \end{cases} \quad (7)$$

Umbra 는 입력값이 그레이 준위보다 큰 경우는 '1'로 작은 경우는 '0'으로 하는 연산으로 식 (8)과 같이 표현된다.

$$U[F] = \{ (x,y) \mid y \leq f(x) \} \quad (8)$$

Top 은 Umbra의 그레이 준위 최대값을 나타내며 식 (9)로 표현된다.

$$TOP[F] = \max \{ y \mid (x,y) \in A \} \quad (9)$$

일반화된 단위 하드웨어는 연산 처리시간을 단축하기 위해서 이진 영상을 이용한 방법을 사용하고 있으며, 형태소의 최대값에 영향을 받지 않도록 설계하였다. 또한 Dilation 연산이나 Erosion 연산시 가산기나 감산기를 사용하지 않는 하드웨어가 되어 전체 연산 시간의 감소효과를 얻을 수 있다. 일반화된 Dilation 연산과 Erosion 연산은 식(10)과 식(11)과 같이 표현할 수 있다.

$$f \oplus_g k = \pi \bigcup_{j=0}^{M-1} \bigcup_{i=0}^N (S[f_j] \oplus_b S[k_i]) \\ = \{ \{ \overline{f_0}_b \oplus_b \overline{k_1}_b \} \\ + \{ \{ \overline{f_1}_b \oplus_b \overline{f_1}_b \} \cup \{ \overline{f_0}_b \oplus_b \overline{f_0}_b \} \} \\ + \dots \\ + \{ \{ \overline{f(M-1)}_b \oplus_b \overline{k_1}_b \} \cup \dots \cup \{ \overline{f(M-N)}_b \oplus_b \overline{N_1}_b \} \} \} \quad (10)$$

$$f \oplus_g k = \pi \bigcup_{j=0}^{M-1} \bigcup_{i=0}^N (S[f_j] \oplus_b S[k_i]) \\ = \{ \{ \overline{f_0}_b \oplus_b \overline{k_1}_b \} \\ + \{ \{ \overline{f_1}_b \oplus_b \overline{k_1}_b \} \cup \{ \overline{f(M-1)}_b \oplus_b \overline{k_0}_b \} \} \\ + \dots \\ + \{ \{ \overline{f(M+1)}_b \oplus_b \overline{k_N}_b \} \cup \dots \cup \{ \overline{f_1}_b \oplus_b \overline{k_0}_b \} \} \} \quad (11)$$

따라서 위의 식들을 이용하여 하드웨어를 설계하면 형태소의 값에 영향받지 않는 하드웨어의 구현이 가능하다.

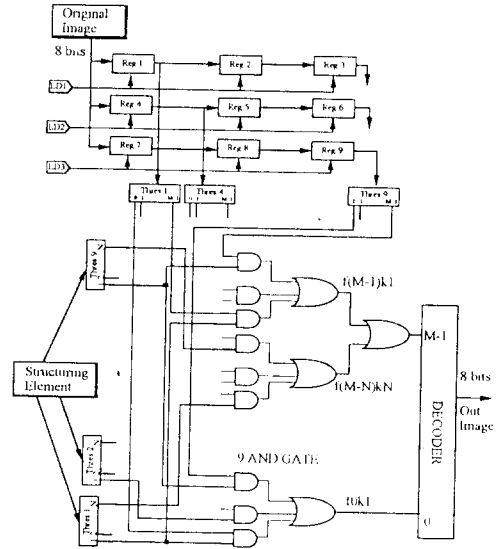


그림 10. Dilation 하드웨어

그림 10은 식(10)을 사용하여 설계한 그레이 준위 영상처리가 가능하고 형태소의 크기가 3 X 3 인 일반화된 Dilation 연산 하드웨어의 블록도이다. 입력 부분을 행렬 형태로 구성하여 전체적인 연산 처리 시간을 크게 단축할 수 있는 이점이 있다. Thres 블록은 그레이 준위값을 이진 영상의 형태로 바꾸어 주는 블록이며 Decoder 블록은 이진 영상 형태의 값을 그레이 준위값으로 전환 시켜주는 블록이다.

V. 결론

디지털 영상신호처리 알고리즘들을 소프트웨어로 기존의 단순 구조의 컴퓨터 시스템에 구현하면 처리속도가 각 분야 응용에서 요구하는 수준을 만족하지 못하

는 경우가 많다. 그러나 영상신호처리 알고리즘들은 국부성, 규칙성과 반복성을 가지므로 어레이 처리나 파이프라인 처리와 같은 병렬화가 가능하며 이에 따른 병렬 하드웨어를 구현하면 실시간 처리의 요구사항을 만족시킬 수 있다. 병렬 컴퓨터 구조의 발전과 최근 초고집적회로의 발전에 힘입어 특정 알고리즘을 위한 전용 하드웨어들이 영상통신등의 여러 분야에서 설계되어 구현되고 있으며 실시간 처리의 요구를 만족시키고 있다. 특성이 우수하나 시간의 복잡도(time complexity)가 높은 영상처리 알고리즘들의 하드웨어화는 앞으로 빠르게 진행될 것으로 전망된다.

參 考 文 獻

[1] Zahid Hussain, Digital Image Processing, Ellis Horwood, 1991.

[2] Sing-Tze Bow, Pattern Recognition and Image Processing, Mavcel Dekker, 1992.

[3] S. Y. Kung, VLSI Array Processors, Prentice Hall, 1988.

[4] 경종민, 김성래, 박구호, 이황수, 디지털 신호 처리를 위한 VLSI 구조, 방한 출판사, 1987.

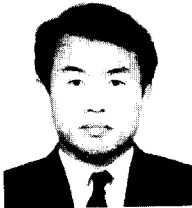
[5] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, 1988.

[6] 이 임걸, 최 인아, "Gray scale 형태학적 변환법과 산업응용을 위한 특징 점 검출", 대한전자공학회 하계 학술 발표 논문집, 제 9 권, 1호, 1993

[1] Zahid Hussain, Digital Image Processing,

筆 者 紹 介

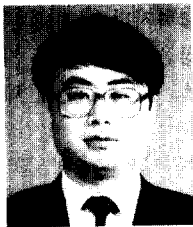
朴 東 宣



1955年 3月 7日生
 1979年 2月 고려대학교 전자공학과 졸업
 1984年 12月 미국 미주리 주립 대학원 공학 석사
 1990年 12月 미국 미주리 주립 대학원 공학 박사

1991年 3月 ~ 현재 전북대학교 정보통신공학과 조교수

李 俊 煥



1957年 7月 26日生
 1980年 2月 한양대학교 전자공학과 졸업
 1982年 2月 한국 과학 기술원 공학 석사
 1990年 8月 미국 미주리 주립 대학원 공학 박사

1985年 4月 ~ 현재 전북대학교 전자공학과 조교수

주관심 분야 : 영상 처리, 컴퓨터 비전, 인공지능