

論文97-34C-8-3

## 새로운 수리형태학 필터 VLSI 구조 설계

### (Design of a New VLSI Architecture for Morphological Filters)

邕壽煥\*, 鮮于明勳\*

(Soohwan Ong and Myung Hoon Sunwoo)

#### 요 약

본 논문에서는 수리형태학 필터 칩의 새로운 VLSI 구조를 제안하고 이의 설계 및 구현에 대해 기술한다. 제안하는 구조는 연산의 중간값을 재사용하기 위해 궤환 루프를 이용하였으며 최대/최소값을 검출하기 위해 디코더/인코더쌍 비교기를 이용하였다. 이러한 궤환 루프 구조와 디코더/인코더쌍 비교기로 인해 제안하는 새로운 수리형태학 필터 VLSI 구조는 기존의 필터 구조에 비해 하드웨어를 대폭 줄였다. 뿐만 아니라 제안하는 구조는 동일한 하드웨어로 불림(dilation)과 녹임(erosion)을 모두 수행할 수 있으며 기존의 구조에 비해 연산수가 적다. 또한 3 x 3 이상의 큰 형태소에 대한 확장 구현시 적은 하드웨어만을 추가함으로써 구현이 가능해 확장성이 매우 좋다. 제안하는 VLSI 구조를 VHDL(VHSIC Hardware Description Language) 모델로 정립하였으며 SYNOPSIS™ CAD Tool을 이용하여 논리합성하였다. 논리합성시 SOG (Sea-Of-Gate) 셀 라이브러리를 이용하였으며 실제 칩으로 구현하였다. 총 게이트 수는 2,667개이며 동작 주파수는 실시간 처리요구를 만족하는 30 MHz이다.

#### Abstract

This paper proposes a new VLSI architecture for morphological filters and presents its chip design and implementation. The proposed architecture can significantly reduce hardware costs compared with existing architectures by using a feedback loop path to reuse partial results and a decoder/encoder pair to detect maximum/minimum values. In addition, the proposed architecture requires one common architecture for both dilation and erosion and fewer number of operations. Moreover, it can be easily extended for larger size morphological operations. We developed VHDL (VHSIC hardware description language) models, performed logic synthesis using the SYNOPSIS CAD tool. We used the SOG (sea-of-gate) cell library and implemented the actual chip. The total number of gates is only 2,667 and the clock frequency is 30 MHz that meets real-time image processing requirements.

#### I. 서 론

일반적으로 선형 시스템은 영상 신호를 처리할 때

영상의 구조 및 형태를 왜곡시키는 특성을 갖는다. 이러한 문제점을 보완하기 위해 수리형태학 등을 이용한 비선형 여파기가 많이 사용된다. 특히 수리형태학은 우수한 비선형 특성으로 인해 신호의 잡음 제거, 형태 보존, 윤곽선 보존 등에 우수한 특성을 갖기 때문에 영상 신호처리 분야에 적용되어 영상분할, 결함검출 등에 이용되고 있다<sup>[1-6]</sup>. 그러나 그레이 레벨 영상에

\* 正會員, 亞州大學校 工科大學 電機電子工學府  
(School of Electronical and Electronic Eng., Ajou Univ.)

接受日字:1997年1月15日, 수정완료일:1997年7月4日

대한 수리 형태학 연산은 방대한 실시간 영상 데이터의 양 때문에 매우 많은 연산을 요구하여 소프트웨어적인 방법으로는 실시간 처리가 매우 어렵다. 따라서 연산 시간을 줄이기 위해 많은 병렬 VLSI 구조가 제안되고 있다<sup>[7-9]</sup>.

기존에 제안된 VLSI 구조는 크게 두가지로 분류된다. 첫째는 threshold decomposition을 이용하여 그레이 레벨 영상을 이진 영상화하여 연산 처리하는 방법이다<sup>[7]</sup>. 이 구조는 논리 연산을 이용하여 수리 형태학의 기본 연산인 불림과 녹임을 수행한다. 수리 형태학 연산은 원영상을 형태소라 불리우는 계수의 집합체와 연산 처리한다<sup>[10]</sup>. 그러나 이 방법은 불림과 녹임을 위해 두 개의 복잡한 하드웨어 유니트가 필요하고 연산수와 하드웨어를 감소시키기 위해 형태소의 최대값을 고정시켜야 하기 때문에 형태소의 값이 변하면 연산이 불가능하다. 둘째는 이진 영상화하지 않고 그레이 레벨 영상을 직접 가산기를 이용하여 최대/최소값을 찾아냄으로써 연산을 수행하는 방법이 있다<sup>[8,9]</sup>. 이들은 threshold decomposition를 이용한 방법보다 적은 하드웨어 유니트와 연산 복잡도를 요구하지만 여전히 하드웨어 복잡도의 문제<sup>[8]</sup>와 데이터 변환 문제<sup>[9]</sup>를 내포하고 있다. Data reuse 구조<sup>[8]</sup>는 연산의 중간값을 재사용하기 위해 캐환 루프를 사용하는데, 중간값 저장을 위해 8개의 레지스터가 필요하며 많은 캐환 루프와 이를 제어하기 위한 복잡한 제어 유니트가 필요하다. Digit-serial 구조<sup>[9]</sup>는 redundant 수 체계를 이용하며 비트 시리얼 연산을 수행한다. 따라서 비트 패러렐 데이터와 비트 시리얼 데이터간의 데이터 변환이 필요하며 비트 시리얼 연산으로 인해 연산 속도가 느리다.

이러한 단점들을 줄이기 위해 본 논문에서는 데이터를 재사용하기 위한 간단한 캐환 루프와 감산기를 이용하는 일반적인 비교기 대신 최대/최소값을 검출하기 위해 디코더/인코더쌍 비교기를 이용하는 새로운 수리 형태학 필터 VLSI 구조를 제안한다. 또한 제안하는 구조는 적은 연산 수로 불림과 녹임을 동일한 하드웨어로 연산이 가능한 구조이며 형태소의 크기를 기하급수적으로 증가시켜도 하드웨어는 선형적으로 증가하는 뛰어난 확장성을 가지고 있다. 제안한 구조는 VHDL 모델로 정립하였으며 SYNOPSIS™ CAD Tool을 이용하여 논리합성하였다. 논리합성시 0.8  $\mu\text{m}$  SOG 셀 라이브러리를 이용하였으며 실제 칩으로 구현하였

다. 총 게이트 수는 2,667개이며 동작 주파수는 실시간 처리요구를 만족하는 30 MHz이다.

본 논문은 다음과 같이 구성된다. 2장에서는 수리 형태학의 기본 연산자의 정의 및 개념에 대해 기술한다. 3장에서는 제안하는 수리 형태학 필터 칩의 새로운 VLSI 구조에 대해 기술하며 4장에서는 제안하는 필터 칩의 설계 및 구현에 대해 기술한다. 끝으로 5장에서는 결론을 기술한다.

## II. 수리 형태학의 기본 연산자

이 장에서는 불림, 녹임, 열림(opening) 및 닫힘(closing)과 같은 기본적인 수리 형태학 연산자에 대해 기술한다<sup>[10]</sup>. 일반적으로 수리 형태학은 집합 이론에 근거하지만 그레이 레벨 영상에 대한 수리 형태학 연산은 집합보다는 함수의 개념이 강하게 내포되어 있다.

### 1. 불림(Dilation)과 녹임(Erosion)

그레이 레벨의 불림 연산은<sup>[10]</sup>에 정의되어 있다.  $f(s, t)$ 와  $b(s, t)$ 를 각각 원영상과 형태소에 대한 2차원 그레이 레벨 함수라고 하고  $D_f$ 와  $D_b$ 를 각각  $f(s, t)$ 와  $b(s, t)$ 의 도메인이라고 하면  $f(s, t)$ 를  $b(s, t)$ 로 불림하는 수식은  $(f \oplus b)(s, t)$ 로 표시되며 이는 식 (1)과 같이 표현될 수 있다.

$$(f \oplus b)(s, t) = \max\{f(s-x, t-y) + b(x, y)$$

$$|(s-x), (t-y) \in D_f \text{ and } (x, y) \in D_b\} \quad (1)$$

여기서  $f(-x, -y)$ 는  $f(x, y)$ 가 원점에 대칭되어 있음을 의미한다. 컨볼루션과 마찬가지로  $f(s-x, t-y)$  함수는  $s$ 와  $t$ 의 양의 방향에서 음의 방향으로 움직인다.  $f$ 의 도메인  $(s-x, t-y)$ 의 값이  $b$ 의 도메인에 포함되어야 한다는 조건은  $f$ 와  $b$ 가 중첩되어야 함을 의미한다. 불림이 형태소의 모양에 의해 정의된 이웃 화소들에서  $f+b$ 의 최대값을 찾아내는 연산이므로 불림 연산의 일반적인 효과는 다음과 같다.

- i) 만약 형태소의 모든 값이 양수라면 출력 영상은 입력 영상보다 밝아진다.
- ii) 영상의 어두운 부분은 형태소의 모양과 값의 크기에 따라 줄어들거나 사라진다.

$f \ominus b$ 로 표기되는 녹임 연산은 식 (2)와 같이 정의된다<sup>[10]</sup>.

$$(f \ominus b)(s, t) = \min \{ (s+x, t+y) - b(x, y) \mid (s+x, t+y) \in D_f \text{ and } (x, y) \in D_b \} \quad (2)$$

Correlation에서와 같이  $f(s+x, t+y)$  함수는  $s$  와  $t$ 의 음의 방향에서 양의 방향으로 움직인다.  $(s+x), (t+y) \in D_f$  이고  $(x, y) \in D_b$  여야 하는 조건은 형태소  $b$ 가 원영상  $f$ 에 완전히 포함되어야 함을 의미한다. 녹임 연산에 대한 일반적인 효과는 다음과 같다.

- i) 만약 형태소의 모든 값이 양수라면 출력 영상은 입력 영상보다 어두워진다.
- ii) 영상의 밝은 부분은 형태소의 모양과 값의 크기에 따라 줄어들거나 사라진다.

## 2. 열림(Opening)과 닫힘(Closing)

원영상  $f$ 를 형태소  $b$ 로 열림 연산을 수행하는 수식은 식 (3)과 같이 정의된다.

$$f \circ b = (f \ominus b) \oplus b \quad (3)$$

열림 연산은 원영상  $f$ 를 형태소  $b$ 로 녹임 연산을 수행한 후 그 결과를 다시 형태소  $b$ 로 불림 연산을 수행함으로써 구현할 수 있다. 이와 마찬가지로 닫힘 연산은 식 (4)와 같이 표현될 수 있다.

$$f \bullet b = (f \oplus b) \ominus b \quad (4)$$

열림과 닫힘은 불림과 녹임의 조합 연산으로 구현이 가능하기 때문에 우리는 불림과 녹임을 커널 연산으로 선택하여 이에 대한 VLSI 구조를 설계하였다.

## III. 제안하는 수리 형태학 필터 칩의 새로운 VLSI 구조

앞에서 기존의 수리 형태학 연산 구조의 단점들을 기술하였다. 이러한 단점들을 극복하기 위해 우리는 알고리즘을 정립하여 수리 형태학 필터 칩의 새로운 VLSI 구조를 제안한다.

### 1. 구조 추출을 위한 알고리즘

그림 1은 3 x 3 형태소와 입력 영상을 나타낸다.  $S_{mn}$ 과  $I_{ij}$ 는 각각  $mn$ 과  $ij$ 에 위치한 형태소와 입력 영상의 화소를 나타내고 있다.

식 (1)에 따라 불림 연산을 다르게 표현하면 식 (5)와 같다.

$S_{11}$	$S_{12}$	$S_{13}$
$S_{21}$	$S_{22}$	$S_{23}$
$S_{31}$	$S_{32}$	$S_{33}$

(a)

$I_{11}$	$I_{12}$	$I_{13}$
$I_{21}$	$I_{22}$	$I_{23}$
$I_{31}$	$I_{32}$	$I_{33}$

(b)

그림 1. 형태소와 입력 영상

(a) 형태소 ( $S_{mn}$ ), (b) 입력 영상 ( $I_{ij}$ )

Fig. 1. The Structuring Element and Input Image  
(a) Structuring Element ( $S_{mn}$ ), (b) Input Image ( $I_{ij}$ )

$$I \oplus S = \max \{ I_{11} + S_{33}, I_{12} + S_{32}, I_{13} + S_{31},$$

$$I_{21} + S_{23}, I_{22} + S_{22}, I_{23} + S_{21},$$

$$I_{31} + S_{13}, I_{32} + S_{12}, I_{33} + S_{11} \} \quad (5)$$

기본적으로 불림 연산은 9번의 덧셈과 두 개의 입력에 대한 8번의 비교가 필요하다. 만일 수식을 직접 구현한다면 불필요한 하드웨어 유니트들이 많이 포함될 것이다. 연산의 중간값을 캐환시켜 재사용한다면 많은 하드웨어 유니트를 감소시킬 수 있다.  $max_1$ 을 식 (5)의 첫 번째 열에 대한 최대값으로 정의하면  $max_1 = \max \{ I_{11} + S_{33}, I_{21} + S_{23}, I_{31} + S_{13} \}$ 이 된다. 마찬가지로  $max_2$ 를  $\max \{ I_{12} + S_{32}, I_{22} + S_{22}, I_{32} + S_{12} \}$ 라고 정의하고  $max_3$ 를  $\max \{ I_{13} + S_{31}, I_{23} + S_{21}, I_{33} + S_{11} \}$ 라고 정의하자. 그러면 식 (5)는 다음과 같이 표현될 수 있다.

$$I \oplus S = \max \{ max_1, max_2, max_3 \} \quad (6)$$

불림 연산은 세 단계로 나뉘어 수행될 수 있다. 첫 번째 단계는  $max_1$ 을 구하는 과정으로 식 (7)과 같이 표현될 수 있다.

$$max_1 = \max \{ I_{11} + S_{33}, I_{21} + S_{23}, I_{31} + S_{13} \} \quad (7)$$

두 번째 단계는  $max_2$ 가 계산되고  $max_1$ 과  $max_2$  중 최대값을 검출하는 과정이다.  $max_1$ 이  $max_2$ 와 비교되

기 위해  $max_1$ 은 저장된 후 비교기로 케환되어야 한다.  $max_{12}$ 를  $max_1$ 과  $max_2$  중 최대값으로 정의하면  $max_{12}$ 는  $\max\{max_1, max_2\}$ 로 표현될 수 있다. 이를 다시 풀어 쓰면 식 (8)과 같이 표현된다.

$$max_{12} = \max\{I_{12} + S_{32}, I_{22} + S_{22}, I_{32} + S_{12}, max_1\} \quad (8)$$

세 번째 단계에서  $max_3$ 가 연산되고  $max_{12}$ 와 비교된다.  $max_{12}$ 가  $max_3$ 과 비교되기 위해  $max_{12}$ 는 저장된 후 비교기로 케환되어야 한다. 세 번째 단계에서 불림 연산은 완료되며 이를 수식으로 표현하면 식 (9)와 같다.

$$I \oplus S = \max\{I_{13} + S_{31}, I_{23} + S_{21}, I_{33} + S_{11}, max_{12}\} \quad (9)$$

이와 같은 방법으로 녹임 역시 다시 정의될 수 있다. (2)식을 변환한 녹임 연산은 식 (10)과 같이 표현될 수 있다.

$$I \ominus S = \min \begin{matrix} \{I_{11} - S_{11}, I_{12} - S_{12}, I_{13} - S_{13}, \\ I_{21} - S_{21}, I_{22} - S_{22}, I_{23} - S_{23}, \\ I_{31} - S_{31}, I_{32} - S_{32}, I_{33} - S_{33}\} \end{matrix} \quad (10)$$

녹임 연산도 3 단계로 정의될 수 있다. 먼저  $min_1$ 을  $\min\{I_{11} - S_{11}, I_{21} - S_{21}, I_{31} - S_{31}\}$ 로 정의하고  $min_{12}$ 를  $\min\{I_{12} - S_{12}, I_{22} - S_{22}, I_{32} - S_{32}, min_1\}$ 라고 정의하자. 첫 번째 단계에서  $min_1$ 을 연산하고 두 번째 단계에서  $min_{12}$ 를 연산한다. 불림에서와 마찬가지로  $min_{12}$ 를 연산하기 위해  $min_1$ 은 저장된 후 비교기의 입력으로 케환되어야 한다. 세 번째 단계에서 녹임 연산은 완료되며 이를 위해  $min_{12}$ 는 저장된 후 케환되어야 한다.

### 2. 전체 구조

3 x 3 형태소에 대한 제안하는 수리 형태학 필터 칩의 전체 구조는 그림 2에 나와있다. 이 구조는 연산의 중간값을 재사용하기 위한 케환 루프를 포함하고 있다. 이러한 간단한 케환 루프를 사용하므로써 가산기와 비교기의 개수를 줄일 수 있다. 제안하는 필터 칩은 오직 3개의 가산기와 디코더/인코더쌍으로 구성된 4 입력 비교기 하나만을 포함하고 있다. 디코더/인코더쌍 비교기는 다음 절에서 설명한다. Data reuse 구조<sup>[8]</sup> 역시 케환 루프를 사용하지만 연산의 중간값을 저장하기 위한 8개의 레지스터와 5개의 2 입력 비교기 및 많은 케환 루프와 이를 제어하기 위한 복잡한 제어 유닛을 필요로 한다. 이에 반해 제안하는 구조

는 오직 4개의 레지스터로 연산의 중간값을 저장할 수 있으며 4 입력 비교기 하나와 하나의 케환 루프 및 이를 제어하기 위한 간단한 제어 유닛만을 요구하기 때문에 하드웨어의 크기를 대폭 줄일 수 있다.

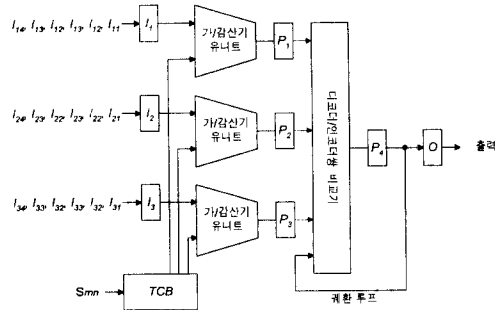


그림 2. 제안하는 수리 형태학 필터 칩의 구조  
Fig. 2. The Proposed Architecture of a Morphological Filter Chip.

입력 영상의 각 열은 입력 레지스터( $I_1 \sim I_3$ )로 입력되며 형태소는 3개의 원환 버퍼(Three Circular Buffers:TCB)에 순차적으로 저장된다. 입력 핀 수를 줄이기 위해 TCB를 이용하여 형태소를 저장한다. TCB는 형태소의 9개 화소를 하나씩 순차적으로 입력받기 때문에 형태소를 위한 입력 핀은 8개만 필요하다. 초기에 형태소를 모두 저장한 후 TCB는 형태소의 한 열을 동시에 가/감산기 유닛으로 출력한다. 가/감산기 유닛은 입력되는 형태소의 화소들과 입력 영상을 더하거나(불림) 빼준다(녹임).

앞 절에서 설명했듯 불림은 세 단계로 나뉘어 수행될 수 있다. 첫 번째 단계에서  $max_1$ 을 연산하기 위해 먼저 가/감산기 유닛의 중간 결과값은 중간값 레지스터  $P_1 \sim P_3$ 에 저장된다.  $P_1$ 에는  $I_{11} + S_{33}$ ,  $P_2$ 에는  $I_{21} + S_{23}$ ,  $P_3$ 에는  $I_{31} + S_{13}$ 이 각각 저장되어 있다. 이들은 비교기로 입력되어 비교기에서는 최대값( $max_1$ )을 검출하여  $P_4$ 에 저장된다. 두 번째 단계에서  $max_{12}$ 를 연산하기 위해 가/감산기 유닛에서는 다음 입력에 대해 연산을 수행한다. 따라서  $P_1$ 에는  $I_{12} + S_{32}$ ,  $P_2$ 에는  $I_{22} + S_{22}$ ,  $P_3$ 에는  $I_{32} + S_{12}$ 이 각각 저장되어 있다. 이들은 비교기로 입력되어  $P_4(max_1)$ 와 함께 4개의 값 중 최대값( $max_{12}$ )을 검출해 낸다. 검출된  $max_{12}$ 는 다시  $P_4$ 에 저장된다. 세 번째 단계에서  $P_1$ 에는  $I_{13} + S_{31}$ ,  $P_2$ 에는  $I_{23} + S_{21}$ ,  $P_3$ 에는  $I_{33} + S_{13}$ 이 각각 연산되어 저장된다. 그리고  $P_4$ 에 저장된  $max_{12}$ 와 비교기를 통해 최대값을 검출함으로써 불림

연산을 완료한다. 이러한 세 단계의 연산을 마친 후 최종 결과값은 출력 레지스터(O)에 저장된다.

이와 같은 방식으로 녹임 연산도 이 구조를 이용하여 구현될 수 있다. 가/감산기 유니트가 가산 대신 감산을 수행하고 비교기가 최대값 대신 최소값을 검출하게 되면 같은 구조를 이용하여 녹임 연산이 구현된다. 반면 불림과 녹임을 위해 개별적인 하드웨어를 필요로 하는 구조도 존재한다<sup>[7]</sup>.

3. 디코더/인코더쌍 비교기

디코더/인코더쌍 비교기는 효율적으로 최대/최소값을 검출하기 위해 제안하였다. 디코더/인코더쌍 비교기는 형태소의 크기가 커질 경우 적은 하드웨어만을 추가함으로써 구현이 가능하기 때문에 확장성이 매우 좋다.

녹임을 다른 수식으로 표현하면 식 (12)와 같이 표현될 수 있다.

$$f \ominus b (s, t) = \text{the number of 1's [ AND}\{2^{f_{xy} \cdot b_{yx}} - 1\} \{f_{xy} \in I_{xy} \text{ and } b_{yx} \in W_{yx}\}] \quad (12)$$

여기서  $I_{xy}$ 와  $W_{yx}$ 는 각각  $xy$  위치에 있는 입력 영상과 형태소의 화소를 나타낸다.  $(2^{f_{xy} \cdot b_{yx}} - 1)$ 은  $f_{xy} + b_{yx}$ 의 크기와 같은 수의 1의 갯수를 갖는 이진수로 표현되며 이들의 AND 연산은 1의 개수가 최소인 값을 찾아낸다. 따라서  $\{f_{xy} + b_{yx} | f_{xy} \in I_{xy} \text{ and } b_{yx} \in W_{yx}\}$  중 최소값을 검출할 수 있다. 녹임과 마찬가지로 불림을 표시하면 식 (13)과 같다.

$$f \oplus b (s, t) = \text{the number of 1's [ OR}\{2^{f_{xy} \cdot b_{yx}} - 1\} \{f_{xy} \in I_{xy} \text{ and } b_{yx} \in W_{yx}\}] \quad (13)$$

DeMorgan의 법칙에 의해 A OR B는 NOT  $\{(\text{NOT } A) \text{ AND } (\text{NOT } B)\}$ 이므로 녹임에서 사용되는 AND 게이트를 불림에서도 재사용하기 위해 식 (13)을 고쳐쓰면 다음과 같다.

$$f \oplus b (s, t) = \text{the number of 1's [ NOT [ AND}\{(\text{NOT}\{2^{f_{xy} \cdot b_{yx}} - 1\}) \{f_{xy} \in I_{xy} \text{ and } b_{yx} \in W_{yx}\}\}] \quad (14)$$

디코더/인코더쌍 비교기는 그림 3에 나타난다. 이 비교기는 감산기를 이용하는 일반적인 비교기와 달리 4개의 디코더와 45개의 2 입력 AND 게이트와 하나의 인코더로 구성되어 있다. 비교기에 사용되는 디코더는 일반적인 디코더를 조금 변형한 형태의 디코더를

사용하였다.  $2^{f_{xy} + b_{yx}} - 1$ 에서 알 수 있듯이 만일 입력 영상과 형태소를 더한 값이 3이라고 하면  $2^3 - 1$ 을 이진수로 표시하면 '111111'이 된다. 따라서 디코더의 출력 중 '1'의 개수가 입력의 크기인 3을 나타낸다. 디코더 출력의 상태중 모두 '0'인 상태를 추가하였기 때문에 디코더의 입력이 n 비트이면 출력은  $2^n - 1$  비트이다. 만일 디코더의 입력이 8 비트이면 출력은 255 비트가 된다. 이는 매우 많은 수의 wire를 요구하고 매우 복잡한 상호연결망을 형성하게 된다. 이러한 복잡도를 줄이기 위해 디코더의 입력을 8 비트 대신 두 개의 4 비트 데이터로 나누어 사용하였다. 따라서 디코더의 출력 비트수는 15 비트이다. 인코더는 불림의 경우 최대값을, 녹임의 경우 최소값을 출력한다. 이러한 디코더/인코더쌍 비교기는 게이트 지연 시간을 줄일 수 있고 일반적인 비교기에 비해 확장성이 뛰어나다. 확장성은 다음 절에서 설명된다.

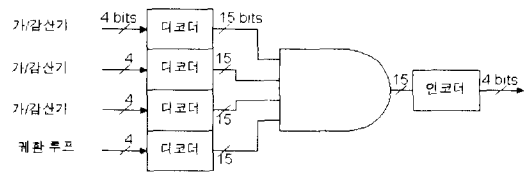


그림 3. 디코더/인코더쌍 비교기  
Fig. 3. The Decoder/Encoder Pair Comparator.

4. 확장성과 설계 및 구현

제안하는 구조는 케환 루프와 디코더/인코더쌍 비교기로 인해 확장성이 매우 뛰어나다. 그림 4는 형태소의 크기가 5 x 5인 경우에 대한 구조를 나타낸다. 그림에서 알 수 있듯이 5개의 가/감산기 유니트, 6개의 4 비트 디코더, 하나의 15 비트 인코더, 연산 중간값을 저장하기 위한 6개의 중간값 레지스터 그리고 75 개의 2 입력 AND 게이트로 구성된다. 3 x 3의 형태소에 대해 구현한 구조와 비교할 때 증가하는 하드웨어는 2개의 가/감산기 유니트, 2개의 디코더, 30개의 2 입력 AND 게이트 뿐이다. 7 x 7 형태소에 대해 구현하면 제안하는 구조에서 소모되는 하드웨어는 9개의 가/감산기 유니트, 10개의 디코더, 하나의 인코더, 8개의 중간값 레지스터와 105개의 2 입력 AND 게이트이다. 이처럼 형태소의 크기가 기하급수적으로 증가하더라도 가/감산기 유니트의 개수는 케환 루프에 의해 적게 소모되며 비교기는 약간의 AND 게이트와 디코더만 증가하면 되므로 하드웨어는 선형적인 증가를

보인다.

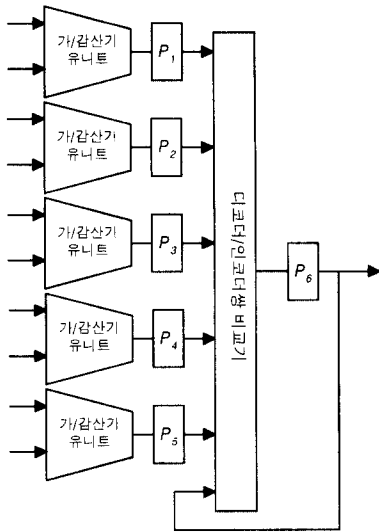


그림 4. 5 x 5 형태소에 대한 구조  
Fig. 4. The Architecture for the 5 x 5 Structuring Element.

표 1. 제안하는 구조와 data reuse 구조의 확장성 비교  
Table 1. The Comparison between Two Architectures for Scalability

개수	제안하는 구조	data reuse 구조
레지스터 (중간값 저장)	6	14
가/감산기 유니트	5	5
비교기 (게이트수)	6개 디코더, 1개 인코더, 75개 AND 게이트 (990)	9개 2 입력 비교기 (1377)
캐환 루프	1	14

제안하는 필터 칩 구조의 확장성을 비교하기 위해 data reuse 구조<sup>[8]</sup>와 비교하였다. 5 x 5 형태소에 대한 구조에서 제안하는 구조와 data reuse 구조<sup>[8]</sup>는 동일한 수의 가/감산기 유니트를 필요로 한다. 그러나 data reuse 구조<sup>[8]</sup>는 14개의 레지스터, 14개의 2 입력 비교기와 많은 캐환 루프를 필요로 한다. 반면 제안하는 구조는 오직 6개의 중간값 레지스터와 상대적으로 적은 게이트로 이루어진 디코더/인코더쌍 비교기 하나만 필요로 한다. 따라서 제안하는 구조가 data reuse 구조에 비해 확장성이 뛰어난을 알 수 있다. 표 1은 제안하는 구조와 data reuse 구조의 확장성을 비교한 표이다.

우리는 VHDL을 이용한 top-down IC 설계 방식에 따라 제안하는 구조에 대한 동작 및 구조 모델을

설계하였다. 설계된 모델을 SYNOPSIS™ CAD Tool의 VHDLDBX(VHDL Simulator)를 이용하여 시뮬레이션을 수행하였다. 먼저 각 유니트들의 동작 모델을 구현하였고 그 후 각 유니트들을 포트 매핑하여 구조 모델을 구현하였다. 구현된 VHDL 모델로 불림과 녹입에 대해 모두 시뮬레이션을 수행하여 기능을 검증하였다. 논리합성은 구현된 VHDL 모델을 이용하여 SYNOPSIS™의 Design Analyzer를 사용하여 수행하였다. 0.8 μm SOG 셀 라이브러리를 이용하여 논리합성하였으며 기능 및 타이밍 시뮬레이션을 수행하였다. 논리합성된 모델을 바탕으로 실제 칩으로 구현하였으며 제안한 수리 형태학 필터 칩의 총 게이트 수는 2,667개이고 동작 주파수는 실시간 영상처리 요구량인 초당 30 프레임을 처리할 수 있는 30 MHz이다. 그림 5는 제안한 수리형태학 필터 칩의 사진이다.



그림 5. 제안한 수리 형태학 필터 칩의 논리합성도  
Fig. 5. The Photograph of the Proposed Morphological Filter Chip.

#### IV. 결론

본 논문에서는 기존의 구조들<sup>[7-9]</sup>에 비해 하드웨어의 크기를 줄이고 확장성이 뛰어난 새로운 수리 형태학 필터 VLSI 구조를 제안하였으며 제안한 필터 칩의 설계 및 구현에 대해 기술하였다. 본 논문에서 제안한 캐환 루프는 가/감산기 유니트의 수를 크게 줄였으며 3 x 3 이상의 큰 형태소에 대해 형태소의 크기가 기하급수적으로 증가하여도 가/감산기 유니트의 수는 선형적으로 증가하기 때문에 확장성이 우수함을 알 수 있다. 또한 디코더/인코더쌍 비교기는 감산기를 이용한 일반적인 비교기에 비해 게이트 지연 시간을 줄일 수 있었으며 형태소의 크기를 확장 구현하였을 때

에도 적은 AND 게이트와 디코더만을 증가시키면 되므로 확장성이 뛰어나다. 제안한 수리 형태학 필터 칩은 top-down IC 설계 방식에 따라 설계되었으며 설계시 SYNOPSIS™ CAD Tool을 사용하였다. SOG 셀 라이브러리를 이용하였으며 총 게이트 수는 2,667개이다. 동작 주파수는 30 MHz이며 이는 실시간 영상처리 요구를 만족한다. 추후에는 연산 속도 개선에 관한 연구와 형태소의 크기가 변하는 다해상도 응용 영역에 적합한 새로운 병렬 구조에 대해 연구할 것이다.

### 참 고 문 헌

- [1] Robert M. Haralick, Xinhua Zhuang, Charlott Lin, James S. J. Lee, "The Digital Morphological Sampling Theorem", *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 2067-2090, December 1989.
- [2] Philippe Salembier, "Comparisioin of Some Morphological Segmentation Algorithms Based on Contrast Enhancement Application to Automatic Defect Detection", *Signal Processing V : Theories and Application*, pp. 833-836, 1990.
- [3] Philippe Salmbier and Jean Serra, "Morphological Multiscale Image Segmentation", *Proc. SPIE vol. 1818 Visual Communication and Image Processing '92.*, pp. 620-631, 1992.
- [4] R. L. Stevenson and G. R. Arce, "Morphological Filters : Statistics and Further Syntactic Properties", *IEEE Trans. on Circuits and Systems*, vol. CAS-34, pp. 1292-1305, November 1987.
- [5] P. Maragos and R. W. Shafer, "Morphological Filters - Part II: Their Relations to Median, Order-Statistic, and Stack Filters", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no. 8, August 1987.
- [6] Robert M. Haralick, Stanley R. Sternberg and Xinhua Zhuang, "Image Analysis Using Mathematical Morphology", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, pp. 532-550, July 1987.
- [7] F. Y. Shih, O. R. Miychell, "Threshold Decomposition of Gray-Scale Morphology into Binary Morphology", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol II, no. 1, January 1989.
- [8] M. Sheu, J. Wang, J. Chen, A. Suen, Y. Jeang and J. Lee, "A Data-Reuse Architecture for Gray-Scale Morphologic Operations", *IEEE Trans. on Circuits and Systems-II*, vol. 39, no. 10, October 1992.
- [9] L. Lucke, C. Chakrabarti, "A Digit-serial Architecture for Gray-Scale Morphological Filtering", *IEEE Trans. on Image Processing*, vol. 4, no. 3, March 1995.
- [10] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Addison Wesley, 1992.

### 저 자 소 개



邕壽煥(正會員)

1994년 2월 아주대학교 전자공학 학사. 1996년 2월 아주대학교 전자공학 석사. 1996년 3월 ~ 현재 아주대학교 전자공학 박사과정. 관심분야는 영상, 통신 및 신호처리용 ASIC 설계



鮮于明勳(正會員)

1980년 2월 서강대학교 전자공학 학사. 1982년 2월 한국과학기술원 전자공학 석사. 1982년 3월 ~ 1985년 8월 한국전자통신연구소(ETRI) 연구원. 1985년 9월 ~ 1990년 8월 Univ. of Texas at Austin 전자공학 박사. 1990년 8월 ~ 1992년 8월 Motorola, DSP Chip Division, 미국. 1992년 8월 ~ 1996년 10월 아주대학교 전기전자공학부 조교수. 1996년 10월 ~ 현재 아주대학교 전기전자공학부 부교수