

論文 90-27-5-16

2진 영상의 고속 세션화 장치 구현에 관한 연구

(A Study on Fast Thinning Unit Implementation of Binary Image)

許允碩*, 李載春*, 郭允植*, 李大寧*

(Yoon Seok Heo, Jae Choon Lee, Yoon Sick Kwak, and Dae Young Lee)

要 約

본 논문에서는 Stanley R. Sternberg가 제안한 파이프라인 기법을 변형하여 영상 표현 및 패턴인식의 전처리로 사용되는 세션화를 고속처리할 수 있는 장치를 구현하였다.

구현된 세션화 장치는 인터페이스부, 국부 메모리부, 주소 발생부, 세션화 처리부, 제어부로 구성되었으며, 세션화 처리부는 SAP의 단점인 지연 파이프라인과 윈도우 파이프라인 수가 영상의 열크기와 같아야 함을 본 구현에서는 테이블 맵핑 기법을 이용하여 룩-업 테이블을 만들어 줌으로서 세션화가 이루어지는 부분을 간략화 시켰다. 메모리부는 독립적으로 두개를 구성시켜 데이터의 흐름을 이르게 하였다.

구현된 세션화 장치는 구조의 복잡성을 피하였으며 간단한 논리비트의 변경에 의하여 영상의 크기의 용통성을 갖는다.

Abstract

In this paper we implemented the fast thinning unit by modifying the pipeline architecture which was proposed by Stanley R. Sternberg.

The unit is useful in preprocessing such as image representation and pattern recognition etc. This unit is composed of interface part, local memory part, address generation part, thinning processing part and control part.

In thinning processing part, we shortened the thinning part which performed by means of look-up table using window mapping table.

Thus we improved the weakness of SAP, in which the number of delay pipeline and window pipeline are equal to image column size.

Two independent memorys using tri-state buffer enable the two direction flow of address generated by address generation part. This unit avoids the complexity of architecture and has flexibility of image size by means of simple modification of logic bits

I. 서 론

영상 처리 기술의 발전은 인식을 위한 특징 추출 등의 방대한 데이터량을 신속히 처리하기 위한 방법

이 요구되었으며 종래에는 가격면이나 사용성에서 우수한 소프트웨어에 의존했으나, 속도면에서 대단히 우수한 하드웨어 의존적 처리방법으로 전환하려는 연구가 계속 되어왔다. 이 중에서 가장 활발히 연구가 진행되는 분야는 병렬처리(parallel processing) 기법을 이용하는 것으로 공간적 병렬처리 방식과 시간적 병렬처리 방식으로 나눌 수 있다. 공간적 병렬처

*正會員, 慶熙大學校 電子工學科
(Dept. of Elec. Eng., Kyunghee Univ.)

接受日字: 1989年 12月 19日

리 방식은 SIMD(single instruction multiple data) 구조 방식이며, 시간적 병렬처리 방식은 파이프라인(pipeline)기법을 이용한 것이다. SIMD 구조는 다수 개의 PE(processing element)를 사용하여 같은 처리를 동시에 수행할 수 있으며 MIMD 구조는 다수 개의 PE가 독립적인 처리를 동시에 수행할 수 있다.^[1]

영상 처리에 있어 SIMD과 파이프라인 구조를 이용한 시스템은 평활화(smoothing), 윤곽추출(edge detection), 세선화(thinning) 등과 같은 전처리 과정을 고속으로 처리하는데 적합하며, MIMD 구조를 이용한 시스템은 영상의 분석, 인식에 이용하기에 적합하다.

특히 전처리는 영상 데이터에서 추출할 수 있는 정보를 결정짓는 중요한 단계로서 대부분의 전처리 기법이 윈도우 연산(window operation)으로 처리 가능한 단순 반복적인 알고리즘으로 영상 데이터량이 방대하므로 시간이 많이 소요하게 된다.

SIMD 구조를 이용한 고속 시스템은 다수 개의 PE에 영상 데이터를 나누어 주고 동일한 처리를 동시에 처리하도록 하는 시스템으로 PE의 갯수만큼 속도 향상과 다양한 처리가 구현될 수 있는 장점이 있는 반면, 구조가 복잡하여 시스템 구현, 상용화가 어렵다.

그러므로 이를 극복하여 저렴한 비용으로 구현이 가능하며 전처리 과정에서 병렬성이 충분히 고려된 병렬처리 시스템의 개발이 요구된다. 이러한 요구를 만족시키려면 여러개의 프로세서를 사용하지 않으면서 병렬성을 도출하여야 하므로 공간적 병렬처리 방식 보다는 시간적 병렬처리 방식인 파이프라인(pipeline) 기법을 사용하여야 한다.^{[1][4][5][6][7][8][9]}

본 논문에서는 파이프라인 구조를 기본으로 하고, 테이블 맵핑기법을 이용하여 룩-업 테이블을 구성하므로써 세선화 장치를 간략화 시켰다. 또한 간단한 논리회로의 변경으로 영상 크기에 따라 변형이 가능하다. 이를 응용하여 영상처리 알고리즘을 하드웨어 화하여 범용의 영상처리 시스템 구현 가능성을 확인하였다.

II. 세선화 알고리즘

세선화는 2진 영상을 단일화소 두개의 골격 패턴으로 재구성하는 것을 의미하며 이는 패턴 인식의 중요한 전처리 단계로 응용되고 있다

● 세선화 알고리즘은

- 첫째, 패턴의 연결성이 유지되어야 하며,
- 둘째, 윤곽점으로 구분된 점은 중심축을 기준으로

대칭적으로 제거 되어야 한다. 는 관점에서 연구, 제안되어 왔다.^{[2][3][11][12][13][14][15]}

영상 데이터는 2진 영상을 사용함으로써 불체인 "1"과 배경을 나타내는 "0"으로 구분하며, 사용된 윈도우는 3×3의 국부연산자로 중심화소값에 대한 8개의 주위화소를 조사하여 중심화소의 제거 여부를 판단하게 된다.

세선화 과정은 중심화소 P1(i, j)의 값이 "1"의 값을 취할 때 수행하게 된다. 세선화 과정은 이미 결정된 템플릿에 순차적으로 연산을 적용 비교하여 골격을 추출하는 방법으로, 직접적인 로직 연산에 의한 골격 추출방법이 아닌 세선화 알고리즘에 의한 화소점의 제거 기준 결정값이 룩-업 테이블(lookup-table:LUT)에 저장되어 각 해당하는 조건에 따라 결정값을 취하는 테이블 맵핑(table mapping)법을 도입, 사용함으로써 하드웨어 지향적인 장치 구현에 용이성을 제공하였다. Chen과 Heu가 제안한 병렬세선화 알고리즘은 Zhang과 Suen의 것에 비교하여 복잡도는 높아졌으나 테이블 맵핑기법에 의하여 그 수행 효율은 증대시키면서 복잡도는 상대적으로 상쇄시킬 수 있는 방법이다.

그림 2는 중심화소 P1(i, j)에 대한 제거 조건을 위한 8방향 이웃에 주소 비트 할당과 이 해당 주소에 대한 룩-업 테이블에 적용 방법의 개략도를 보였으며, 그림3, 그림 4는 각각 1st 부분반복의 세선화 LUT와 2nd 부분반복의 세선화 LUT를 작성하여 도시하였다. LUT는 그림 2(a)에 나타난 것과 같이 8-이웃화소에 대하여 각각에 주소비트를 설정하고 세선화 알고리즘에 기초한 마스크 형태에 따라 생성되는 상위, 하위 주소를 그림 3에 수직방향과 수평방향으로 교차하는 부분의 값에 따라서 세선화가 이루어진다.

III. Serial Array Processor 설계

영상처리에 있어서 n번째 반복에서 화소에 주어질 새로운 값은(n-1)번째 반복에서 그 자신의 값과 주

P9 (i-1, j-1)	P2 (i-1, j)	P3 (i-1, j+1)
P8 (i, j-1)	P1 (i, j)	P4 (i, j+1)
P7 (i+1, j-1)	P6 (i+1, j)	P5 (i+1, j+1)

그림 1. 3×3 윈도우
Fig. 1. 3×3 window.

A 7	A 0	A 1
A 6	P(i, j)	A 2
A 5	A 4	A 3

(a)

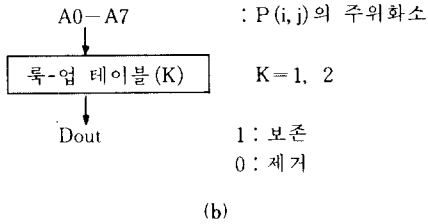


그림 2. (a) 8-이웃 화소와 주소 비트
(b) 주소 비트의 LUT
Fig. 2. (a) 8-neighborhood pixel & address bit,
(b) LUT diagram of address bit.

low address: A3A2A1A0

2nd	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	1	1	1	0	1	1	0	0	1	1	1	1	0	1	0	1	0
1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	1	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	1	1	1	1	1	1	1	0	1	1	1	0	1	0	1	0
4	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
6	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
7	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0
8	1	1	0	1	0	1	1	0	1	1	1	1	1	1	1	1	0
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
C	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
D	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
E	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
F	0	1	1	1	1	1	1	0	1	1	1	0	1	0	1	0	1

high address: A7A6A5A4

그림 4. 2st 부분 반복 세선화 LUT
Fig. 4. Thinning LUT for 2nd iteration.

low address: A3A2A1A0

1st	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	1	1	0	1	0	0	0	1	1	1	1	0	0	0	0
1	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	1	0	1	0	1	0	1	0	1	1	1	1	0	1	0	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C	0	0	1	0	1	1	0	1	0	1	1	1	1	1	1	0
D	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	0
F	0	0	1	0	1	1	1	0	0	1	0	1	1	1	1	1

high address: A7A6A5A4

그림 3. 1st 부분 반복 세선화 LUT
Fig. 3. Thinning LUT for 1st iteration.

위 8 화소의 값에 영향을 받게된다. (n-1) 번째의 부분 반복에서 좌표 (i, j)의 값을 P(i, j, n-1)로 나타내면 n 번째의 부분 반복에서 그 좌표의 화소값은 식 1과 같다.

$$P(i, j, n) = \text{FUNC}\{P(i-1, j-1, n-1), P(i-1, j, n-1), P(i-1, j+1, n-1), P(i, j-1, n-1), P(i, j, n-1), P(i, j+1, n-1), P(i+1, j-1, n-1), P(i+1, j, n-1), P(i+1, j+1, n-1)\} \quad (1)$$

여기서 FUNC{ }는 적용하는 병렬 연산 처리기법을 나타내며 본 논문에서는 그림 3, 4의 LUT가 된다.

CYTO 컴퓨터란 이름으로 명명되었던 시스템은 시리얼 어레이 프로세싱(serial array processing; SAP) 단들의 연결로 구성시켜 각 단은 단일 클럭 펄스 기간중에 단일 화소의 변환된 값을 생성시키는 일을 수행하도록 한다.

이웃 시리얼 어레이 프로세싱(neighborhood serial array processing)단은 이웃 논리 변환기(neighborhood logic translator)와 윈도우(행×행)의 데이터를 제공받기 위함과 이웃 논리 변환기에 적절한 데이터 배열을 제공하여 국부 연산자를 구성시켜 줄 윈도우 파이프 라인(window pipeline)과 지연 파이프 라인(delay pipeline)으로 구성된다. 이는 영상을 세선화하는데 소요되는 시간의 대부분이 각 화소의 주위값을 메모리로부터 읽어오는 시간이라는 점을 감안하면, 한번 읽어온 데이터를 연속하여 사용하는 본 구성이 효율적임을 알 수 있다.

그림 5(a)는 이웃논리 변환기로 적용하는 처리기법

의 알고리즘을 프로그래밍 할 수 있으며 파이프라인이 모두 차게된 순간부터 매 클럭마다 변환된 값을 출력하게 된다. 그림 5(b)는 윈도우 파이프 라인으로 윈도우 레지스터 내용을 동일 클럭 간격에 중심 화소의 변형된 값을 계산할 이웃논리 변환기에 병렬로 9개의 값을 투입하게 된다. 그림 5(c)는 지연 파이프 라인으로 메모리에서 열 방향으로 읽혀진 데이터를 윈도우 파이프라인으로 메모리에서 각 클럭 간격으로 적절하고 새로운 이웃을 구성하도록 한다.

그림 6은 영상 데이터가 다섯 개의 화소폭을 가진 2차원 배열로 표현되었을 때 연속적인 2개의 SAP 처리 단계를 나타낸 것이다. 단계 1에서 변환된 영상 데이터 8'이 출력되며, 단계 2에서는 변환 영상 데이터 9'이 순차적으로 출력되게 된다. 그림과 같이 영상 데이터가 불필요해질 때까지 지연 파이프 라인과 윈도우 파이프 라인에 보존되므로 각 화소마다 8이웃 데이터를 메모리로부터 읽을 필요없이 한번씩만 읽어오면 된다.

따라서 파이프라인이 차게 되는 순간부터 매 단계마다 이웃논리변환기에 의해 변환된 영상 데이터가 순차적으로 출력하게 된다.

파이프 라인의 열크기는 처리할 영상 데이터 배열의 열 크기와 같으며 행의 크기는 이웃 논리 변환기의 윈도우 연산자의 행 크기와 동일하다. 처리할 영

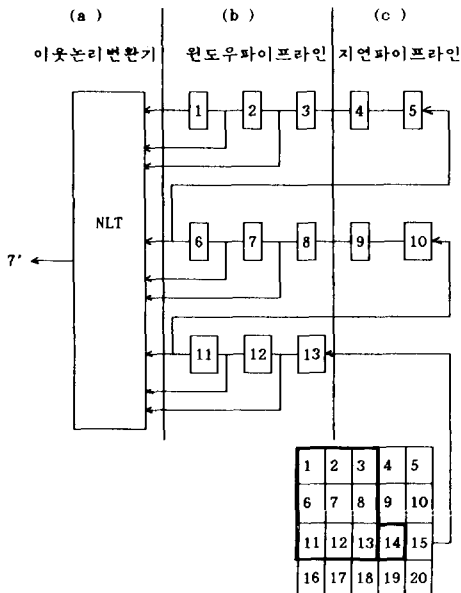


그림 5. SAP 구성도
Fig. 5. Block diagram of SAP.

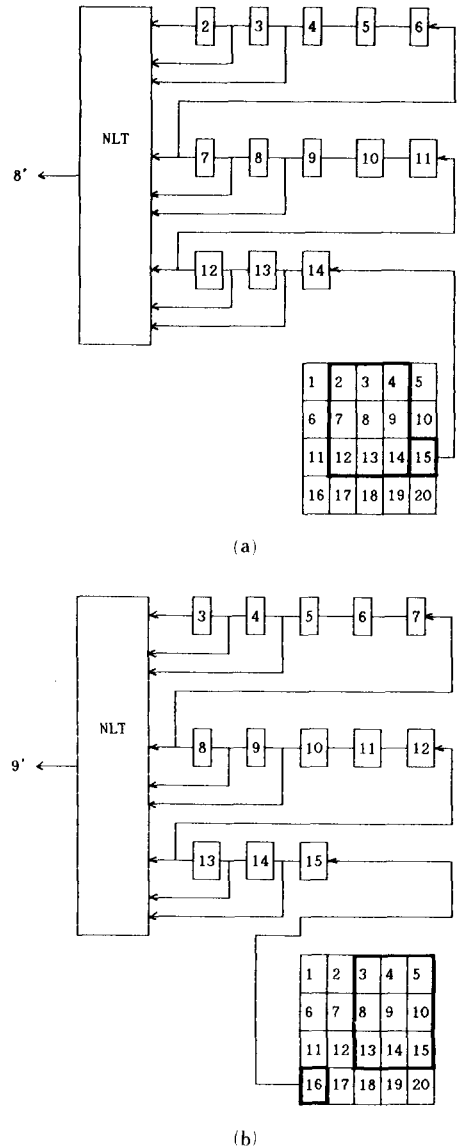


그림 6. SAP의 단계별 예
Fig. 6. Example of SAP for iteration.

상 데이터의 크기가 $M \times N$ ($M=N$) 배열이고, 윈도우 파이프 라인의 크기를 W , 지연 파이프라인의 크기를 D , 연산자의 크기를 LO 라고 하면 다음과 같은 식 2가 성립한다.

$$\begin{aligned} W &= LO \\ M &= D + W \end{aligned} \tag{2}$$

이때 모든 파이프라인이 차서 최초의 변환된 화소값이 출력되는 시간 T_d 는 식 3과 같이 표시할 수 있다.

$$Td=2 \times (D+W)+LO \quad (3)$$

그러나 영상 데이터 배열의 크기가 커지면 지연과 이프 라인이 해당 윈도우 파이프라인에 적절한 데이터를 입력시키기 위해 영상데이터 배열의 열 크기에 비례하여야 하는 단점과 파이프 라인이 차서 최초의 변환된 데이터가 출력되는 시간이 길어진다는 단점이 있다.

IV. 세션화 장치

본 구현된 장치는 32×32, 64×64, 128×128, 256×256등 4가지의 해상도에 융통성을 부여하여 영상의 특성에 따라 영상 크기를 선택하여 세션화를 수행하도록 하였다. 또한 PC의 본체 클럭을 사용하지 않음으로 구현장치 자체에서 원하는 만큼의 처리속도 향상을 얻을 수 있으며 두개의 독립된 메모리를 뒀으로서 처리된 데이터를 보조 메모리로부터 주메모리로 이동시키는 과정없이 다음 처리 단계로 진행하는 장점을 가지고 있다.

본 장치의 특성은 설계의 용이성을 가지고 있는 마이크로 프로그램 기법이 병렬 세션화 알고리즘의 파이프라인 구조에는 적합하지 않아 PC와의 데이터 교환을 제외한 모든 논리를 구현 장치 자체내에서 하드웨어로 구현, 처리 속도면에서 고속화를 이루었다. 그림 7에 세션화 장치의 구성도를 나타내었다.

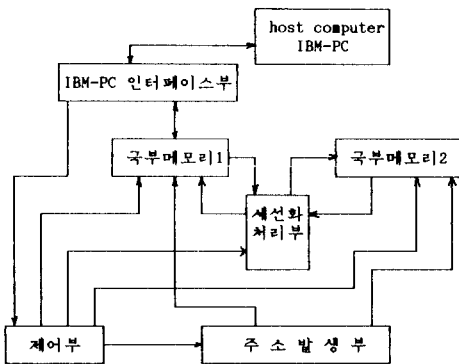


그림 7. 세션화 장치 구성도
Fig. 7. Block diagram of thinning unit.

IBM-PC 인터페이스부를 통해 영상 데이터를 국부 메모리 1로 저장한 다음 제어부에 출발 신호를 줌으로써 PC와는 독립적으로 구현된 장치내 제어부의 제어하에 세션화 과정을 수행하게 된다.

국부 메모리 1에서 열방향으로 읽혀진 영상 데이터는 세션화 처리부의 윈도우 레지스터 부분으로 해당 행을 고려하여 입력되고 이는 중심화소 P(i, j)가 "1"과 "0"일 경우를 구분하여 패턴 "1"일 경우만 LUT에 제거여부를 묻게된다. 그러므로 처리부는 논리 연산부나 프로세서를 사용하지 않고 윈도우 파이프라인에서 공급되는 8개의 주위 화소를 조합하여 LUT의 주소 비트로 만들면된다. P(i, j)의 결과는 국부 메모리 2에 저장하게 되며 하나의 부분 반복 체계에 따른 세션화 과정이 끝나면 국부 메모리 2에서 국부 메모리 1 방향으로 두번째의 부분 반복 체계를 적용시켜 결과값의 흐름이 일어나게 된다. 이와같은 과정은 더 이상의 제거점이 존재하지 않을때까지 반복 수행된다.

1. 인터페이스 부

IBM-PC에 저장되어 있는 영상 데이터를 세션화 장치 국부 메모리로 이동, 저장시키며 세션화 장치가 IBM-PC측에서 볼 때 추가된 프로세서로 동작을 이루어지도록하며, 8255 포트별로 역할을 구분하여 필요한 명령을 주도록 세션화 장치의 초기화, 시작 신호등과 같은 제어신호를 제어부에 전달한다. 구현된 장치의 시작신호는 장치내에 자체 클럭 발생기를 가지고 있으므로 PC에서 국부 메모리 1로 데이터 이동은 PC에 의한 외부 클럭으로 동작하게 하고, 초기화 후는 클럭 발생기에 절체신호를 줌으로써 장치의 내부 수행이 이루어지도록 하였다. 그림 8은 인터페이스부의 구성도이다.

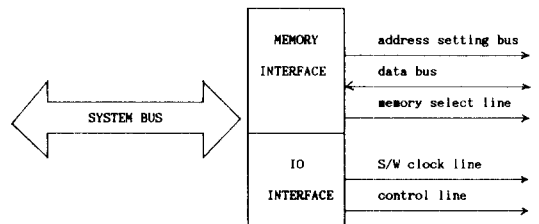


그림 8. 인터페이스 구성도
Fig. 8. Block diagram of interface

2. 국부 메모리부

국부 메모리 1은 PC측으로부터 초기 영상 데이터를 입력받아 저장하게 되며 제어부의 시작신호 후에는 국부 메모리 2와 상호 부분 세션화가 이루어진 데이터를 교환하며 LUT에서 출력되는 결과를 저장하게 된다. 열에 따라 해당 행방향을 고려하며 윈도우

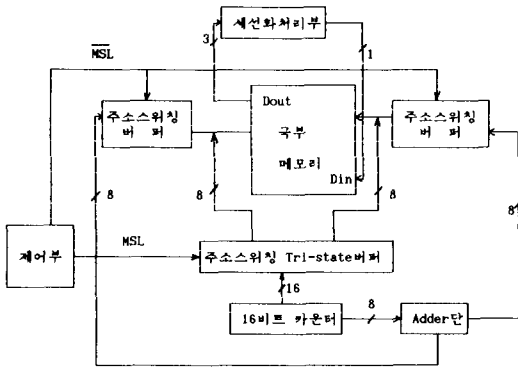
우 레지스터에 데이터를 입력시켜야 하기 때문에 행의 크기인 3개의 국부 메모리 소자를 한조로 이루도록 했다.

WRITE 시에는 3개의 국부 메모리 소자에 동시에 결과 데이터가 기입되지만 READ 시에는 3개의 국부 메모리 소자가 해당 행번지 데이터만을 윈도우 레지스터에 입력하도록 했다.

6개의 메모리 소자는 각 64 Kbyte, 70nsec인 GM76 C64-70 STATIC RAM이다.

3. 주소 발생부

1클럭 주기마다 1씩 증가하는 카운터로 구성되었으며 2개의 메모리 부분에 적절한 주소를 공급하도록 주소 스위칭 역할의 tri-state 버퍼를 사용하였다. LS393과 LS161의 조합에서 발생하는 16비트 카운터는 제어부의 조절을 받아 데이터 값을 읽어 윈도우 레지스터로 공급하는 메모리에 원 카운터의 주소 비트를 제공하고 세션화 결과값의 데이터를 저장하는 메모리에는 [영상 최해열의 크기+세션화를 수행하는 열의 위치]에 기입할 수 있도록 Adder단을 통한 조절된 주소 비트를 제공받도록 하였다. 국부 메모리의 주소 발생에 대한 기본 구성을 그림9에 보였다.



MSL: Memory Select Line. (READ/WRITE)
RMSL: Reverse Memory Select Line.

그림 9. 국부 메모리부와 주소발생부
Fig. 9. Block diagram of local memory and addressing part.

4. 처리부

세션화가 행해지는 부분으로 메모리로부터 읽어 온 데이터를 1클럭 주기마다 1쉬프트 데이터 전송을 이루도록 LS164로 구성시켜 윈도우 파이프 라인을 담당하게 하였으며 중심 화소가 영상 패턴 "1"인 경우에만 주위 8화소 값이 LUT의 주소 비트로 작용하

여 제거 판단의 결과값을 LUT로 부터 출력 받을 수 있도록 기본 논리 게이트를 조합시켜 주었다. LUT는 병렬 세션화 알고리즘을 적용하였을 때의 조건 결과 값인 그림 3, 4를 각 적용 방향에 따라 2부분의 2732A EPROM에 LUT의 값을 저장 시켰으며, 그림 10에 세션화 처리부의 구성도를 보였다.

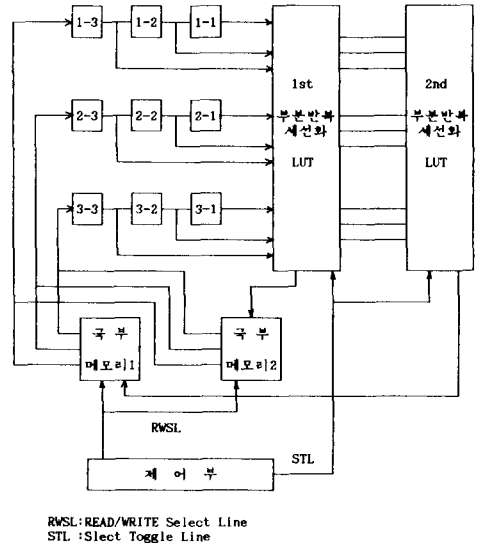


그림 10. 세션화 처리부 구성도
Fig. 10. Block diagram of thinning processing part.

5. 제어부

장치 초기화 이후 인터페이스부를 통해 PC로 부터 모든 제어에 관한 역할을 넘겨 받은 제어부는 주소 발생부에서 생성된 주소 비트를 2개의 구분된 메모리에 적절히 공급하기 위하여 주소 스위칭 버퍼에 tri-state를 발생시켜 줌과 동시에 해당 부분 수행에 따른 메모리의 상태를 조절하여 준다.

또한 세션화 알고리즘의 특성상, 2개의 LUT로 구성시켰으므로 이들 LUT의 선택 신호도 발생시켜 주며, 국부 연산자의 이동에 따른 M×N 2차원 영상 배열에서의 행렬 변화를 고려하여 i=1 or i=M or j=1 or j=N인 경우는 중심 화소의 값에 변화를 주지 않는 논리로 구성시켰다. 제어부는 tri-state 버퍼와 플립플롭, 기본 논리 게이트, 단안정 멀티바이브레이터, 멀티플렉서등 다수의 논리회로로 구성시켰으며 수행 과정이 종료되었을 시는 최종 결과 데이터를 국부 메모리 1로 이동시켜 PC쪽으로 영상 데이터를 넘겨주게 된다. - 그림11은 제어부에 대한 구성도를 보였다.

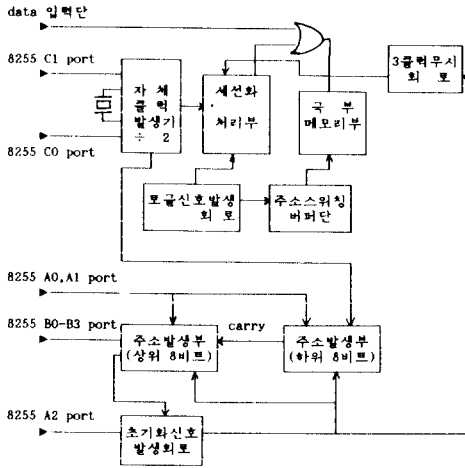


그림 11. 제어부 구성도
Fig. 11. Block diagram of controller part.

V. 실험 및 결과 고찰

본 실험에서는 그림12와 같은 장치 구조를 334×108mm EPOXI 62핀 확장 보드에 구성시켜 이를 IBM-PC 확장 슬롯에 장착하고, 세션화 전용 프로세서로 동작하도록 구현하여 실험하였다. 영상 데이터는 스캐너를 이용하여 64×64, 128×128, 256×256 해상도를 가진 2진 영상을 획득하여 실험에 사용하였다. 구현된 세션화 장치의 초기화 및 처리 종료후의 결과·데이터를 관리하는 주 프로그램은 어셈블리어와 C언어로 작성하여, IBM-PC 주 컴퓨터에서 수행하도록 하였다. 일반적으로 병렬처리 구조를 이용한 시스템의 성능 평가 중에서 수행시간(processing time:PT) 과 시스템 산출량(throughput)에 대하여 본 연구에서 구현한 장치를 이 기준에 따라 영상크기 64×64, 반복횟수 5회인 문자 영상 패턴에 대하여 평가를 행하였다.

1. 수행시간(PT)

구현된 장치에는 LS급 TTL소자를 사용하였으며 소자고유의 전파지연시간(propagation delay time)을 감안하여 4MHz를 50% duty로 2분주 한 500 nsec를 1클럭 주기로 자체 클럭 발생부에서 공급하였다. 파이프라인 구조의 장점은 파이프라인이 차게되면 단수에 관계없이 1클럭 주기마다 하나의 변형 결과를 출력하게 된다는 것이다.

영상 프레임의 수행시간 PT는 다음과 같이 고려하여 계산할 수 있다.

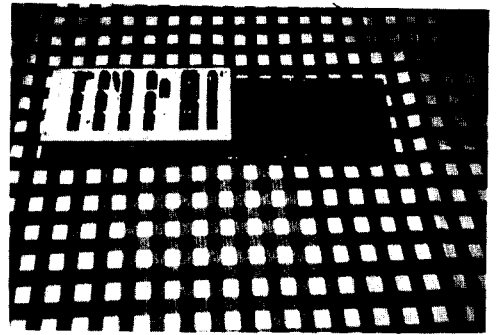


그림 12. 세션화 장치
Fig. 12. Thinning processing unit.

$$PT = (N+I) \times T \times R \tag{4}$$

여기서 “N”은 영상 크기, “I”는 초기화에 의한 윈도우 파이프 라인이 차게 되는 지연시간을 표현하며, “T”는 클럭사이클(500 nsec) 또는 화소율(pixel rate)이라 부르며, “R”은 반복수행 연산의 수를 의미한다. 따라서 본 구현 장치에 대한 수행시간은

$$PT = (4096+3) \times 500 \text{ nsec} \times 5 = 0.01 \text{ sec} \text{가 된다.}$$

2. 산출량

단위 시간당 파이프 라인 프로세서에 의해 수행되는 결과의 양을 산출량(TP)이라고 한다. 이 양은 파이프 라인 프로세서의 계산 능력을 뜻하며 다음과같이 정의한다.

$$TP = \frac{N}{(N+I) \times S \times P} = \frac{4096}{0.01 \text{ sec}} = 0.41 \text{ E6/sec} \tag{5}$$

본 연구에서 구현한 세션화 장치가 그림 12이며, 한글 문자 패턴에 대한 원 영상과 세션화 결과 영상을 그림13에서 보였고, 지문 영상에 대한 세션화 결과는 그림14에 보았다.

세션화 장치의 산출량은 영상 패턴 “1”의 총 화소수에 의존하는 것이 아닌 수행 반복의 수에 영향을 미치는 2진 영상의 획의 최대 굵기에 의존한다.

본 고속 세션화 장치의설계 구조는 영상처리를 위해 많은 장점을 가지고 있다. 2진 영상일 경우, 어떠한 3×3 연산자 병렬 알고리즘(평활화, 특징추출, 세션화등)에 대해서도 테이블 맵핑화가 가능하다면 사용 가능하다. 고속 연산 수행이 가능한 본 구현장치의 장점은 (1)기존의 프로세서 장치가 CPU를 처리부로 사용하여 많은 수행 명령어와 이를 디코드 하

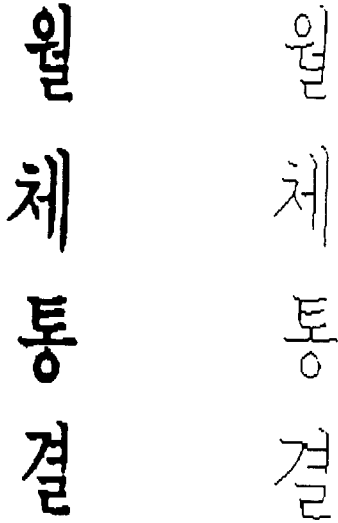


그림 13. 한글 문자 영상
Fig. 13. Image of character.

는데 사용되었던 시간적 손실을 본 장치에서는 단지 클럭펄스 공급에 의해서만 수행하며 CPU를 채택하지 않았다는 것이다.

(2) 간단한 I/O 포트 비트를 조작함으로써 특성에 맞는 화상 크기를 선택할 수 있도록 4종류 크기에 융통성을 부여하였다.

(3) 병렬처리 시스템이 안고 있었던 구조의 복잡도를 배제하고 순수한 하드웨어 지향구조를 지니고 있으며, 소자구입이 용이하여 실용화가 가능하다.

나아가 본 장치의 성능 향상을 위해서는 전파지연 시간이 짧은 소자로의 대체와 주소 발생부에 대한 일부 회로 추가, 변경 및 메모리 용량확장에 의해 고해상도의 처리가 고려되어야 한다. 또한 현재의 VLSI 기술을 도입시킨다면 본 설계와 제작 장치가 CUSTOM IC로 상용화가 쉬우며 그 처리 능력은 5배 이상(10MHz 사용시)의 향상을 기할 수 있다.

VI. 결 론

본 논문에서는 파이프라인 기법을 변형하여 영상 표현 및 패턴 인식등의 전처리 단계로 사용되는 세선화를 고속처리 할 수 있는 장치를 구현하였다. 구현된 세선화 장치는 IBM-PC인터 페이스부, 국부 메모리부, 주소 발생부, 세선화 처리부, 제어부로 구성시켰다.

세선화 처리부는 SAP의 단점인 지연 파이프라인과 윈도우 파이프라인 수만을 고려하여 구성시켰다. 또한 테이블 맵핑기법을 이용하여 록-업테이블을 만들어 줌으로써 세선화가 이루어지는 부분을 간략화시켰다. 메모리 부는 두개의 독립적 기억장소로 구성시켜 주소 발생부에서 발생하는 주소를 tri-state를 이용하여 양 방향성 데이터 흐름을 이루도록 하였다.

구현된 세선화 장치는 병렬처리 시스템 구조의 복잡성을 피하였으며 향상된 처리속도를 기할 수 있고, 간단한 논리 비트 변경에 의해 영상 크기에 융통성을 부여한 장점을 가진다. 또한 일부 회로를 변경만 한다면 영상 크기의 확장 가능성은 충분하다.

본 장치는 전처리 과정에서 소모되는 시간적 소모를 최소화하는 실시간 전처리부를 구성, 적용시킨 것으로 만족할만 한 처리속도를 얻었다. 구현된 장치 세어가 간단한 논리에 의한 하드웨어 지향적 시스템으로 디지털 영상 처리 알고리즘의 고속화를 이룩할 수 있는 범용의 영상처리 시스템 구현 가능성은 충분하리라 본다.



그림 14. 지문 패턴 영상
Fig. 14. Image of fingerprint.

參 考 文 獻

- [1] Hwang K., Briggs F.A., *Computer Architecture and Parallel Processing*, pp 1-229, McGraw Hill, 1984.
- [2] Chen Y.S., Hus W.H., "A modified fast parallel algorithm for thinning digital pattern," *Pattern Recognition Letters* 7, pp. 99-106, Feb. 1988.
- [3] Zhang T.Y., Suen C.Y., "A fast parallel algorithm for thinning digital patterns," *Comm. ACM. vol. 27* no. 3, pp. 236-239, Mar. 1984.
- [4] Stanley R. Sternberg, "Pipeline Architectures for image processing," *Multi-computer and Image Processing*, pp 291-305, Academic Press, 1982.
- [5] Chen Y.S., W.H. Hsu, "A multi-function parallel processor for binary image processing," *Proceeding of National computer symposium* 1985.
- [6] Siegel L.J., Siegel H.J., Swain P.H., "Parallel Algorithm Performance Measures," *Multicomputer and Image Processing*, pp 241-251, Academic Press, 1983.
- [7] Fountain T.J., "A survey of Bits-Serial Array Processor Circuits," *Computing structure for Image Processing*, ed by M.J. B.Duff, pp 1-13, Academic Press, 1983.
- [8] Ramamoorthy C.V., Li H.F., "Pipeline Architecture," *Computing Surveys*, vol 9, no. 1, pp 61-101, 1977.
- [9] Jagadish H.V. et al, "Survey of Pipelining in Computing Arrays," *IEEE Computers*, vol C-35, no. 5, May 1984.
- [10] Anthony P.R., "Survey: parallel computer Architecture for image processing," *CVGIP* 25, pp 68-88, 1984.
- [11] Holt C.M. et al "An improved Parallel Thinning Algorithm," *Comm. ACM* vol. 30, no. 2, Feb. 1987.
- [12] Naccache N.J., Shinghal R., "SPTA: A proposed algorithm for thinning Binary Pattern," *IEEE SMC. vol. SMC-14*, no. 3, May, 1984.
- [13] Stefanelli R., Rosenfield A., "Some Parallel Thinning Algorithm for digital pictures," *JACM*, vol. 18, no. 2, pp 255-264, Apr. 1971.
- [14] Chu Y.K., Suen C.Y., "Alternate Smoothing and Stripping Algorithm for Thinning Digital Patterns," *Signal Processing* 11, pp 207-222, 1986.
- [15] Arcelli C., "A condition for digital point removal," *Signal Processing* 1, pp 283-285, 1979.
- [16] 박일남, "용통성을 갖는 디지털 콘볼루션 처리 전용 시스템 프로세서에 관한 연구" 경희대학교 대학원 석사학위 논문, 1988.
- [17] Gonzales, R.C., Wints, P., *Digital Image Processing*, Addison-Wesley, 1977.
- [18] Rosenfield, A., Kak, A.C., *Digital Picture Processing*, 2nd ed., Academic Press, 1982.

著 者 紹 介

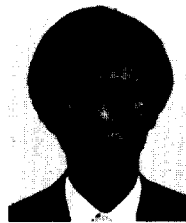


許 允 碩 (正會員)

1964年 2月 26日生. 1987年 2月 경희대학교 전자공학과 졸업 (공학사). 1990年 2月 경희대학교 대학원 전자공학과 졸업 (공학석사) 1990年 1月~현재 신도리코 기술연구소 연구원. 주관심분야는 영상처리, 병렬처리, 패턴인식 등임.

李 載 春 (正會員)

1965年 2月 7日生. 1988年 2月 경희대학교 전자공학과 졸업 (공학사). 1988年~현재 전북대학교 물리학과 표면 연구실에서 STM controller 개발. 주관심분야는 병렬처리, STM, DSP 등임.



郭 允 植 (正會員)

1962年 2月 17日生. 1984年 2月 청주대학교 전자과 졸업. 1986年 9月 경희대학교 전자과 석사 졸업. 1990年~현재 경희대학교 전자과 박사과정 중. 주관심분야는 영상처리 및 Pattern Recognition.

李 大 寧 (正會員)

1940年 3月 18日生. 1968年 9月~1970年 3月 캘리포니아주립대학원 (공학석사). 1976年 9月~1979年 9月 연세대학교 대학원 전자공학과 (공학박사). 1971年 9月 경희대학교 공과대학 전자공학과 조교수. 1977年 3月 경희대학교 공과대학 전자공학과 부교수. 1982年 3月~현재 경희대학교 공과대학 전자공학과 교수. 1988年 1月~현재 통신학회 이사.