

論文2002-39SC-5-5

## 완전탐색 블럭정합 알고리즘을 위한 일차원 시스톨릭 어레이의 구조

## (An Architecture of One-Dimensional Systolic Array for Full-Search Block Matching Algorithm)

李守鎭\*, 禹鍾鎬\*

(Su-Jin Lee and Chong-Ho Woo)

## 요약

본 논문에서는 움직임 추정을 위한 블럭정합 알고리즘의 고속 처리를 위한 VLSI 어레이의 구조를 설계하였다. 완전탐색 블럭정합 알고리즘의 데이터 의존관계로부터 일차원 시스톨릭 어레이를 유도했다. 제안된 일차원 시스톨릭 어레이에 입력된 데이터와 제어신호는 인접한 처리요소를 통해서 전달되어 재사용된다. 따라서 제안된 시스톨릭 어레이는 시간 및 공간적 지역성을 만족한다. 데이터와 제어신호의 입출력 핀은 일차원 어레이의 시작과 끝의 처리요소에만 존재한다. 이 구조는 입력포트의 수가 적으며, 모듈러 확장성을 갖는다. 기준블럭과 최대탐색거리가 확장된 경우에 제안된 어레이를 연결하여 움직임 추정기를 구성할 수 있다.

## Abstract

In this paper, we designed the VLSI array architecture for the high speed processing of the motion estimation used by block matching algorithm. We derived the one dimensional systolic array from the full search block matching algorithm. The data and control signals of the proposed systolic array are passed through adjacent processing element. So proposed architecture has temporal and spatial locality. The I/O ports exists only in the first and last processing elements of the array. This architecture has low pin counts and modular expandability. So the proposed array architecture can be cascaded for different block size and search range.

**Keyword** : 움직임추정, 블럭정합, 로컬패스, 시스톨릭 어레이

## I. 서론

여러개의 연속적인 프레임들로 구성되는 동영상은 많은 시간적·공간적 중복성을 갖는다. 동영상 데이터의 효율적인 저장과 전송을 위해 이러한 중복성을 제

거해야한다. DCT(Discrete Cosine Transform)를 이용하여 프레임 내의 공간적인 중복성을 제거하고, 움직임 추정을 통해 프레임들 사이의 시간적인 중복성을 제거한다<sup>1-2)</sup>. 특히 움직임추정을 구현하기 위한 블럭정합 알고리즘은 많은 계산량을 요구하여 동영상 압축에 필요한 시간의 대부분을 차지한다. 이러한 문제를 해결하기 위해 블럭정합 알고리즘의 병렬성을 이용하여 움직임추정의 고속처리를 위한 병렬프로세서가 많이 제안되었다<sup>3-9)</sup>.

블럭정합 알고리즘의 고속처리를 위한 어레이프로세서들은 데이터의 입출력 핀의 수를 줄이고 블럭이나 탐색영역의 크기 변화에 따른 모듈러 확장이 용이한 일차원 어레이 또는 일차원 어레이의 변형을 통한 이

\* 正會員, 釜慶大學校, 電子컴퓨터情報通信工學部  
(Division of Electronics Computer and Communication Eng., Pukyong National University)

※ 이 논문은 2001년도 부경대학교 연구원교수지원에 의하여 연구되었음.

接受日字:2002年3月5日, 수정완료일:2002年8月26日

차원 어레이들이 제안되었다. Komarek<sup>[3]</sup> 등은 완전탐색 블록정합 알고리즘(FBMA: Full search Block Matching Algorithm)의 처리를 위한 시스템릭 어레이의 구조들을 제안하였고, Yang<sup>[4]</sup> 등은 하드웨어구조가 간단한 일차원 어레이를 제안하였다. Hsieh<sup>[5]</sup> 등은 쉬프트 레지스터를 이용하여 데이터의 직렬입력을 갖는 이차원 어레이를 제안했다. 이 구조들은 처리속도가 늘어 동영상의 실시간 처리에는 적합하지 않다. Pan<sup>[6]</sup> 등은 2단계 BMA 구조를 제안하였으며, Yea<sup>[7]</sup> 등은 인접한 두 기준블럭이 탐색영역의 일부를 공유하는 것을 이용한 이차원 어레이 구조를 제안했다. Lai<sup>[8]</sup> 등은 쉬프트 레지스터 어레이를 이용한 일차원 어레이를 제안했다. 이 구조들은 고속처리가 가능하며 어레이의 모듈러 확장이 가능하다. 그러나 이러한 고속처리를 위한 어레이들이 데이터 전달을 위해 브로드캐스팅 방법을 이용하고 있다. Kittitornkun<sup>[9]</sup> 등은 Yea의 구조를 변경하여 브로드캐스팅을 제거하였으나 다음 행으로 데이터를 전달하기 위한 룹패스를 갖고 있다. 데이터의 브로드캐스팅과 룹패스는 전력의 소비가 많고 동기화를 위한 클럭의 속도의 증가에 제약이 있다<sup>[9-10]</sup>.

본 논문에서는 완전탐색 블록정합 알고리즘의 고속처리를 위한 일차원 시스템릭 어레이의 구조를 제안한다. 제안된 어레이는  $(2p)^2$ 개의 처리요소로 구성되어 있으며, 어레이의 시작과 끝부분의 처리요소를 통해서만 데이터와 제어신호가 입출력된다. 제안된 구조는 기준블럭과 탐색영역의 데이터를 입력하기 위한 3개의 입력포트를 갖는다. 데이터와 제어신호를 전달하기 위한 경로들은 시간 및 공간적 지역성을 갖는다. 제안한 어레이는  $N^2$ 시간마다 기준블럭에 대한 이동벡터를 구할 수 있다. 또한 기준블럭과 최대탐색거리가 확장된 경우에 대해 제안된 어레이의 모듈러 확장을 통해 움직임추정기를 구현할 수 있다.

본 논문의 II장에서는 블록정합 알고리즘과 블록정합 알고리즘의 고속처리를 위한 일차원 시스템릭 어레이의 설계에 대해 살펴본다. III장에서는 결과 및 고찰과 블록정합 알고리즘 처리를 위한 기존의 구조와의 성능비교를 제시한다. 그리고 IV장에서 결론을 맺는다.

## II. 블록정합 알고리즘과 어레이의 설계

### 1. 완전탐색 블록정합 알고리즘

움직임 추정은 현재 프레임을 블럭으로 나누고 각

블럭을 이전 프레임의 블럭과 비교해서 시간적 중복성을 제거하기 위해 이용된다. 현재 프레임을  $N \times N$  픽셀 크기의 블럭으로 나누고 각 블럭을 기준블럭(reference block)라고 한다. 이전 프레임에서 기준블럭의 좌표를 기준으로  $\pm p$  거리의 영역을 탐색영역(search area)라고 하고, 이때의  $p$ 를 최대탐색거리(maximum search range)라고 한다. 움직임 추정을 위한 블록정합 알고리즘은 기준블럭과 이전 프레임의 탐색영역의 후보블럭(candidate block) 중 정합하는 블럭을 찾는다. 완전탐색 블록정합 알고리즘은 탐색영역 내의 모든 후보블럭에 대해 정합여부를 검사하는 방식이다.

기준블럭과 탐색영역 내의 후보블럭의 정합여부를 결정하기 위한 방법으로 연산이 가장 간단한 SAD(Sum of Absolute Difference)를 이용한다. SAD는 식 (1)과 같이 구해지며, 가장 작은 값을 갖는 블럭의 쌍이 정합된 블럭으로 선택된다. 이때의 인덱스를 이동벡터  $MV$ (Motion Vector)로 결정한다.

$$SAD(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i, j) - y(i+m, j+n)| \quad (1)$$

$$MV = (m, n)_{\min(SAD_{m, n})}$$

이 식에서  $x(i, j)$ 와  $y(i, j)$ 는 각각 기준블럭과 후보블럭을 구성하는 픽셀을 의미한다.

### 2. 단일할당코드와 데이터의존그래프

시스템릭 어레이는 모듈성, 규칙성이 있으며 각 처리요소들이 시간 및 공간적 지역성을 가지며 상호접속된 고도의 파이프라인으로 구성된 동기화 다중처리기이다<sup>[10]</sup>. 시스템릭 어레이를 설계하기 위해서 알고리즘으로부터 단일할당코드를 구하고, 데이터 의존성을 표현한 데이터의존그래프를 투영하여 시간 및 공간적 지역성을 갖는 어레이 프로세서를 유도한다. 이때의 투영방향은 생성된 어레이를 구성하는 처리요소의 구조가 간단하고 입출력 핀의 수가 적은 방향을 선택한다<sup>[10]</sup>.

기준블럭의 크기가  $N \times N$ 이고, 최대탐색거리가  $p$ 인 경우의 블록정합 알고리즘을 인덱스 확장을 통해 병렬 프로그램 표현법 중의 하나인 단일할당코드로 바꾸면 그림 1과 같다. 단일할당코드는 각 변수에 단 한번만 값을 대입시키는 병렬알고리즘의 표현형태로써 각 변수들이 종속성을 갖지 않으므로 각 문장을 병렬로 처리할 수 있다. 그림 1의 인덱스  $k$ 는 식 (1)의 인덱스  $i$ 와  $j$ 를 표현한 것이며, 인덱스  $l$ 은 인덱스  $m$ 과  $n$ 을 표현한 것이다. 그리고 변수  $x_i$ 는 기준블럭을 구성하는

```

for ( k = 0 ; k < N2 ; k++ )
  j = k / N
  for ( l = 0 ; l < (2p)2 ; l++ )
    n = l / (2p)
    ye(l, k) = ye(l+1, k-1)
    yo(l, k) = yo(l+1, k-1)
    xi(l, k) = xi(l+1, k)
    iff ( j is even )
      iff ( n is even )
        SAD(l, k) = SAD(l, k-1) + |xi(l, k) - ye(l, k)|
      else
        SAD(l, k) = SAD(l, k-1) + |xi(l, k) - yo(l, k)|
      end if
    else
      iff ( n is even )
        SAD(l, k) = SAD(l, k-1) + |xi(l, k) - yo(l, k)|
      else
        SAD(l, k) = SAD(l, k-1) + |xi(l, k) - ye(l, k)|
      end if
    end if
  end for
end for

```

그림 1. 블럭정합 알고리즘의 단일할당코드

Fig. 1. The single assignment code of the block matching algorithm.

픽셀을 나타내고, 변수  $ye$ 와  $yo$ 는 각각 탐색영역의 짝수열과 홀수열의 픽셀을 의미한다. 기준블럭과 탐색영역의 두 값의 차를 누적하여 변수  $SAD$ 에 저장한다. 그림 1의 단일할당코드에서는 열을 우선으로 기준블럭과 탐색영역의 후보블럭을 비교한다. 탐색영역을 구성하는 인접한 후보블럭들은 중첩되는 데이터를 포함한다. 중첩되는 데이터를 재사용하여 데이터의 필요없는 재입력을 줄이고 파이프라인의 처리율을 향상시킬 수 있다.

단일할당코드로부터 데이터의 의존성을 그래프로 표현한 그림 2의 데이터의존그래프를 유도할 수 있다. 그림 2의 데이터의존그래프는  $(2p)^2$ 개의 행으로 구성되어 있다. 각 행의 노드에서 연산된 값은  $[k \ l]=[1 \ 0]$  방향으로 전달되며 누적되어 각 후보블럭에 대한  $SAD$ 를 계산한다. 회색 원으로 표현된 노드는 연산이 이루어지는 노드이며, 흰색 원으로 표현된 노드는 단순한 데이터의 전달을 위한 노드이다. 기준블럭을 구성하는 픽셀들의 열에 따라 후보블럭을 구성하는 짝수 또는 홀수열의 값과 번갈아 연산이 일어난다. 입력된 기준블럭과 후보블럭의 픽셀은 각각  $[0 \ -1]$ ,  $[1 \ -1]$  방향으로 전달되어 다음 후보블럭에 대한 연산에 재사용된다. 따라서 그림 2의 데이터의존그래프는 식 2와 같은 아크(arc)  $\vec{e}$ 가 존재한다. 데이터의존그래프의 각 행에서 구해진

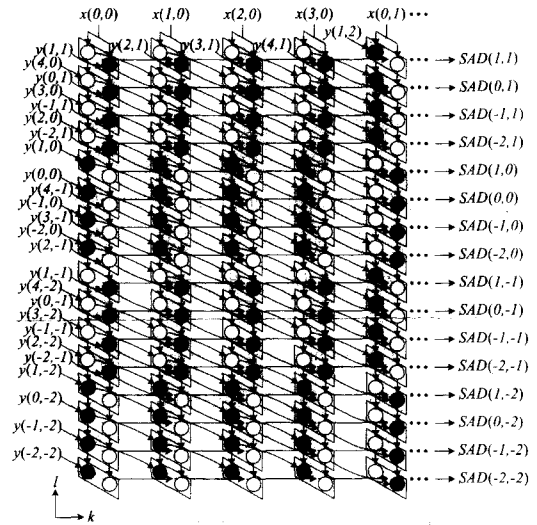
그림 2. 데이터의존그래프( $N=4, p=2$ )

Fig. 2. The data dependence graph( $N=4, p=2$ ).

$SAD$  중 최소값을 갖는 인덱스가 이동벡터로 선택된다.

$$[\vec{e}_1 \ \vec{e}_2 \ \vec{e}_3] = \begin{bmatrix} 0 & 1 & 1 \\ -1 & -1 & 0 \end{bmatrix} \begin{matrix} k \\ l \end{matrix} \quad (2)$$

### 3. 시스톨릭 어레이의 유도

정확한 계산의 순서를 유지하며 시간 및 공간적인 지역성을 만족하기 위해 스케줄 벡터  $\vec{s}$ 와 투영벡터  $\vec{a}$ 는 식 (3)과 (4)의 조건을 만족해야 한다<sup>[10]</sup>.

$$\vec{s}^T \vec{e}_i > 0 \quad \text{for } i=1,2,3 \quad (3)$$

$$\vec{s}^T \vec{a} > 0 \quad (4)$$

그림 2의 데이터의존그래프에서 식(3)을 만족하는 최적의 스케줄 벡터  $\vec{s}$ 는 식 (5)와 같다.

$$\vec{s} = [1 \ -1] \quad (5)$$

투영벡터  $\vec{a}$ 의 선택기준은 식(4)를 만족하면서 투영된 결과의 신호흐름그래프(SFG : Signal Flow Graph) 또는 시스톨릭 어레이의 처리요소의 수와 입출력 핀수가 적으며, 처리요소의 내부구조가 간단한 방향을 선택한다. 투영벡터  $\vec{a}$ 를 식 (6)과 같이 선택하는 경우 파이프라인 주기  $a$ 는 식 (7)과 같이 구해진다. 파이프라인주기가 1이라는 것은 입력데이터의 사이에 지연시간 없이 연속적으로 시스톨릭 어레이에 입력된다는 것을 의미한다.

$$\vec{d} = [1 \ 0] \tag{6}$$

$$a = \vec{s}^T \vec{d} = 1 \tag{7}$$

설계하는 어레이의 공간변환벡터  $P^T$ 는 식 (8)과 같이 투영벡터  $\vec{d}$ 에 직교하는 벡터로 선택된다.

$$P^T = [0 \ 1] \tag{8}$$

공간변환벡터  $P^T$ 와 데이터의존그래프를 이루는 에지의 내적으로 식 (9)과 같이 투영된 결과로 나타나는 시스톨릭 어레이나 신호흐름그래프의 에지  $\vec{e}_i$ 를 구할

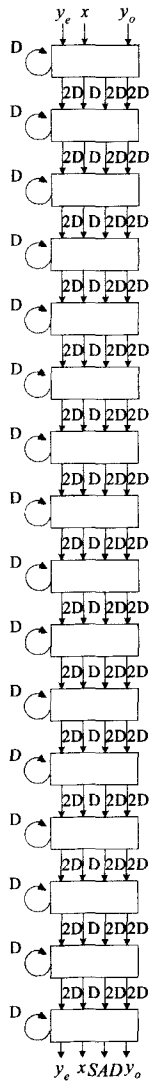


그림 3. 일차원 시스톨릭 어레이의 구조( $N=4, p=2$ )  
 Fig. 3. The architecture of one dimensional systolic array( $N=4, p=2$ ).

그림 3의 일차원 시스톨릭 어레이는 호스트 컴퓨터로부터 프레임을 구성하는 기준블럭과 탐색영역 내의 후보블럭에 대한 픽셀을 입력받아 SAD를 연산하고 처리요소 내에 기억한다. 각 후보블럭에 대해 계산된 SAD의 값은 인접 노드를 통해 외부로 전달된다.

4. 일차원 시스톨릭 어레이의 구조  
 기준블럭의 크기( $N \times N$ )가  $4 \times 4$ 이고, 최대탐색거리  $p$

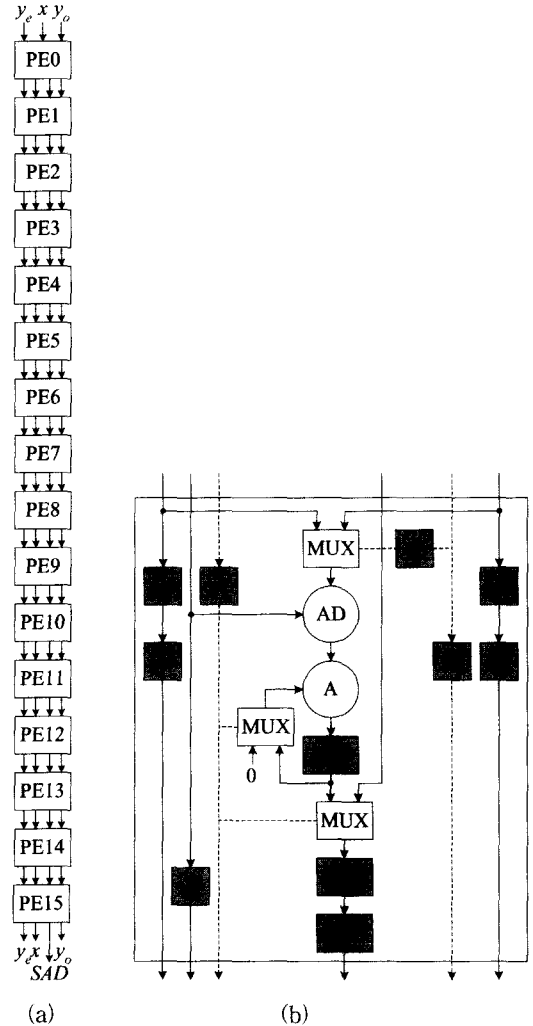


그림 4. 시스톨릭 어레이의 구조와 처리요소의 내부 조직( $N=4, p=2$ ) (a) 시스톨릭어레이의 구조 (b) 처리요소의 내부 조직

Fig. 4. The architecture of one dimensional systolic array and the internal organization of the processing element( $N=4, p=2$ ). (a) The architecture of one dimensional systolic array (b) The internal organization of the processing element

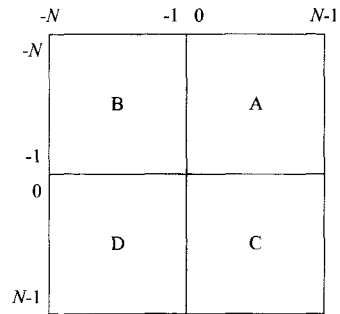
가 2인 경우에 대한 그림 3의 일차원 시스톨릭 어레이의 구조는 그림 4와 같이 구현될 수 있다. 설계한 블럭정합 알고리즘의 처리를 위한 일차원 시스톨릭 어레이는  $(2p)^2$ 개의 처리요소로 구성된다. 최대탐색거리  $p$ 가 2인 경우, 어레이는 16개의 처리요소로 구현되며, 각 처리요소는 해당하는 이동벡터에 대한 SAD값을 계산하여 저장한다. 그림 4의 (b)는 처리요소의 내부 조직을 표현한 것이다. 회색사각형은 레지스터와 플립플롭과 같은 기억 소자이며, AD와 A는 각각 두값의 차의 절대값과 가산기의 기능을 갖는 조합회로이다. 설계한 어레이 프로세서는 총 3개의 데이터입력 포트를 갖는다. 기준블럭의 값을 입력받기 위한 입력포트  $x$ 와 후보블럭의 짝수와 홀수 열의 값을 입력받기 위한 입력포트  $y_e, y_o$ 가 각각 존재한다. 어레이 프로세서의 출력포트는 기준블럭과 후보블럭의 값을 출력하기 위한 출력포트  $x, y_e, y_o$ 와 계산된 SAD를 출력하기 위한 출력포트 SAD가 존재한다. 외부에서 어레이 프로세서를 제어하기 위한 제어신호  $cr$ 과  $ct$ 가 PEO에 입력되어 다음 처리요소로 순차적으로 전달된다. 제어신호  $cr$ 은 새로운 기준블럭의 계산이 시작될 때 SAD의 값을 누적하기 위한 레지스터의 초기화와 계산된 SAD의 전달기능을 제어한다. 제어신호  $ct$ 는 후보블럭의 짝수와 홀수열 중 기준블럭과 연산이 일어나는 값을 선택하는 기능을 수행한다. 처리요소 내부의 플립플롭  $sy$ 는 짝수열과 홀수열 중 하나를 선택하는 MUX를 제어한다. 제어신호  $ct$ 는 플립플롭  $sy$ 의 값을 토글시킨다. 제어신호  $cr$ 과  $ct$ 의 전달도 시간 및 공간적 지역성을 만족한다.

5. 기준블럭과 탐색영역의 크기 변화에 따른 어레이의 모듈러 확장

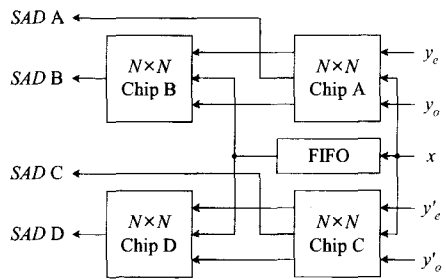
움직임추정을 위해 블럭정합 알고리즘을 이용하는 다양한 응용에 따라 서로 다른 블럭의 크기나 최대탐색거리를 요구한다. 따라서 움직임추정을 위해 설계된 어레이 프로세서가 다른 크기의 블럭 또는 최대탐색거리에 적용가능한 것이 효율적이다<sup>[7-8]</sup>. 본 논문에서 제안한 어레이 프로세서를 연결하여 블럭의 크기나 최대탐색거리가 확장된 응용에 적용할 수 있다.

최대탐색거리가 2배로 확장되어  $p=N$ 인 경우 그림 5와 같이 후보블럭을 4개의 부영역으로 나누고 4개의 칩을 연결하여 움직임추정기를 구현할 수 있다. 그림 4의 어레이 프로세서에 적용할 수 있도록 후보블럭을 그림 5의 (a)와 같이 분할한다. 그리고 각 부분을 4개

의 칩으로 구성된 움직임추정기에 적용하여 각 후보블럭에 대한 SAD값을 계산한다. 부영역 A와 B에 해당하는 후보블럭을 이루는 짝수 및 홀수열의 데이터는 입력된  $y_e, y_o$ 로 입력되고, 부영역 C와 D에 해당하는 후보블럭을 이루는 짝수 및 홀수열의 데이터는 입력된  $y'_e, y'_o$ 로 입력된다. 이 값들은 각각 칩 A와 칩 C를 통해 출력되어 칩 B와 칩 D로 입력된다. 기준블럭의 데이터는 입력된  $x$ 를 통해 칩 A와 칩 C에 입력된다. 그리고  $(2p)^2$  크기의 FIFO를 통과해서 칩 B와 칩 D에 입력된다. 처리요소의 동작을 제어하기 위한 제어신호  $cr$ 과  $ct$ 는 칩 A와 칩 C를 통해 입력된다. 칩 A와 C를 통해 출력되는 이 제어신호가 다시 칩 B와 칩 D에 전달된다. 따라서 기준블럭의 크기가 2배로 확장되는 경우 설계된 4개의 어레이 프로세서를 연결하여 구성하



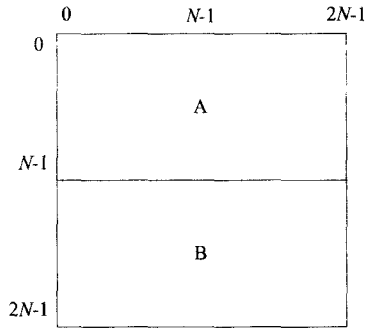
(a)



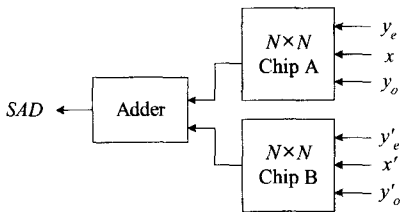
(b)

그림 5. 최대탐색거리가 확장된 경우의 움직임추정기 (기준블럭크기 :  $N \times N$ , 최대탐색거리( $p$ ) :  $N$ ) (a) 탐색영역의 영역분할 (b) 4개의 칩을 연결한 구조

Fig. 5. The motion estimator when the maximum search range is expanded.(The size of the reference block :  $N \times N$ , the maximum search range( $p$ ) :  $N$ ) (a) The partition of the search area (b) An architecture for cascading 4 chips



(a)



(b)

그림 6. 기준블럭의 크기가 확장된 경우의 움직임추정기(기준블럭크기 :  $2N \times 2N$ , 최대탐색거리( $p$ ) :  $N/2$ ) (a) 기준블럭의 영역분할 (b) 2개의 칩을 연결한 구조

Fig. 6. The motion estimator when the size of the reference block is expanded The size of the reference block :  $2N \times 2N$ , the maximum search range( $p$ ) :  $N/2$ . (a) The partition of the reference block (b) An architecture for cascading 2 chips

면, 5개의 데이터 입력포트로 구성된 움직임추정기를 구성할 수 있다.

블럭의 크기가 2배로 확장되어  $2N \times 2N$  크기인 경우 그림 6의 (a)와 같이 기준블럭을 N행씩 두 영역으로 나누고 그림 6의 (b)와 같이 두 개의 칩에 각각 입력하여 SAD의 부분값을 구하고 두 SAD의 부분값을 더해 최종 SAD의 값을 계산할 수 있다. 이때의 움직임추정기는 6개의 데이터입력포트를 가지며, 제어신호 cr과 ct는 두개의 칩에 동일하게 입력된다. 처리요소 내의 레지스터 SAD의 값을 초기화하는 제어신호 cr은  $2N^2$  시간마다 활성화된다.

### 6. 시뮬레이션

설계한 움직임추정기를 위한 시스템릭 어레이의 정확한 동작을 검증하기 위해 ALTERA의 APEX EP20K400BC652-3의 라이브러리를 이용하여 Quartus 1.1의

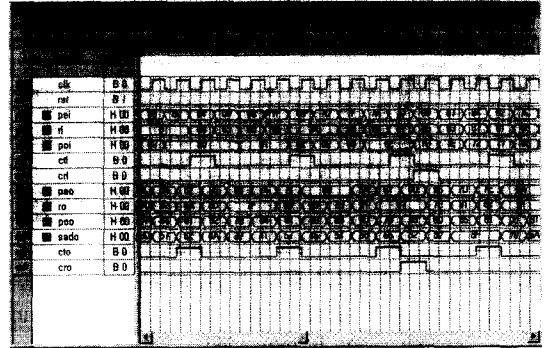


그림 7. 일차원 어레이의 시뮬레이션 결과

Fig. 7. The simulation result of the one dimensional array.

VHDL로 어레이 프로세서를 설계하였다. 기준블럭의 크기( $N \times N$ )가  $4 \times 4$ 이고, 최대탐색거리( $p$ )가 2인 경우에 대해 시뮬레이션을 수행했다.

그림 7은 움직임추정기를 위한 일차원 어레이 프로세서의 시뮬레이션 결과이다. 클럭에 동기를 맞추어 기준블럭과 후보블럭을 구성하는 데이터와 제어신호가 입력된다. 후보블럭을 구성하는 짝수 또는 홀수열의 선택을 제어하기 위한 제어신호 ct는 N시간마다 활성화되고 처리요소내의 레지스터 SAD 값을 초기화하는 제어신호 cr은  $N^2$  시간마다 활성화된다. 어레이 프로세서를 구성하는 처리요소의 내부 레지스터에 저장된  $(2p)^2$ 개의 SAD 값은 다음 처리요소로 차례로 전달되어 최종단의 처리요소 PE15의 출력포트 SAD를 통해 출력된다.

최대탐색거리가 두배로 확장된 경우의 움직임추정기의 동작을 검증하기 위한 시뮬레이션 결과는 그림 8과 같다. 제어신호 ct와 cr의 동작은 그림 7과 동일하다. 움직임추정기를 구성하는 칩중 앞단의 칩에 제어신호와 데이터가 입력되고 이 값들이 다음 칩으로 전달되어 계산이 이루어진다. 각 칩의 출력 포트 SAD를 통하여 탐색영역의 각 부영역에 대한 SAD들이 출력된다. 이 시뮬레이션 결과를 통해 최대탐색거리가 두배로 확장된 경우 4개의 칩을 연결하여 각 후보블럭들에 대한 SAD를 구할 수 있음을 검증할 수 있다.

기준블럭의 크기가 두배로 확장된 경우에 대한 시뮬레이션의 결과는 그림 9와 같다. 제어신호 ct의 동작은 앞의 경우와 동일하고, 제어신호 cr은  $2N^2$  시간마다 활성화된다. 두 칩의 출력인 부분적인 출력 포트 SAD 값을 더하여 각 후보블럭에 대한 SAD 값을 구할 수 있다.

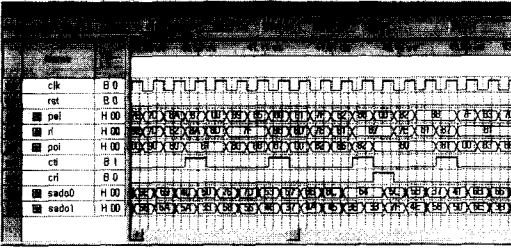


그림 8. 최대탐색거리가 확장된 경우의 시뮬레이션 결과  
Fig. 8. The simulation result when the maximum search range is expanded.

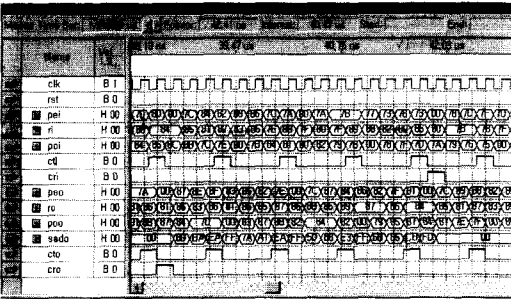


그림 9. 기준블럭의 크기가 확장된 경우의 시뮬레이션 결과  
Fig. 9. The simulation result when the size of reference block is expanded.

### III. 결과 및 고찰

설계한 어레이 프로세서의 성능평가에는 처리요소의 수, 계산시간, 입출력 포트의 수 등의 기준이 사용된다. 어레이의 크기(A)와 계산시간(T)의 곱인 AT는 가격대 성능비를 평가하기 위한 기준으로 사용된다<sup>[10]</sup>. 제안한 시스톨릭 어레이를 구성하는 처리요소의 수, 계산시간, 입력 포트의 수와 특성은 다음과 같다.

#### 1. 처리요소의 수

시스톨릭 어레이의 처리요소의 수는 그림 4와 같이 데이터의존그래프의 투영방향에 의해 결정된다. 기준블럭의 크기가  $N \times N$ 이고 최대탐색거리가  $p$ 인 경우, 완전탐색 블럭정합 알고리즘을 처리하기 위한 제안된 어레이의 처리요소의 수( $N_p$ )는 식 (11)과 같다.

$$N_p = (2p)^2 \tag{11}$$

#### 2. 계산시간

하나의 기준블럭에 대한 후보블럭들의 SAD 값의 연산시간은 그림 2의 데이터의존그래프로부터 구할 수

있다. 어레이의 연산시간( $T_A$ )은 식(12)와 같이 데이터의존그래프 내에서 시간적으로 가장 멀리 떨어진 두 노드의 계산시간 차이와 같다. 그림 2의 데이터의존그래프에 데이터 입력핀의 수를 줄이기 위한 가상노드가 추가된 상태에서 가장 멀리 떨어져 있는 두 노드는  $((2p)^2 - 1, -(2p)^2 + 1)$ 과  $(0, N^2 - 1)$ 이다. 따라서 첫 번째 기준블럭에 대한 후보블럭들의 SAD 값은 식 (12)와 같이  $N^2 + 8p^2 - 2$  시간에 구해지며, 최종 처리요소를 통해 차례로 SAD 값이 출력된다.

$$\begin{aligned} T_A &= \max_{b, q} \{ \vec{s}^T (\vec{b} - \vec{q}) \} + 1 \\ &= [ 1 \quad -1 ] \left[ \begin{matrix} (2p)^2 - 1 \\ -(2p)^2 + 1 \end{matrix} - \begin{matrix} 0 \\ (N^2 - 1) \end{matrix} \right] + 1 \tag{12} \\ &= N^2 + 8p^2 - 2 \end{aligned}$$

첫 번째 기준블럭에 대한 데이터가 입력된 후 연속적으로 다음 기준블럭에 대한 데이터가 입력된다. 따라서 첫번째 기준블럭에 대한 SAD가 연산된 후 블럭과 이프라인주기  $\beta$  시간마다 다음 블럭에 대한 SAD의 연산이 이루어진다. 제안된 시스톨릭 어레이의 블럭과 이프라인주기는 식 (13)과 같이 기준블럭의 크기와 같다.

$$\beta = N^2 \tag{13}$$

따라서 제안한 어레이는 초당 식 (14)의  $f_{\max}$  프레임에 대한 움직임추정을 구현할 수 있다. 이때  $N_v$ 와  $N_h$ 는 프레임을 구성하는 영상의 세로와 가로로 블럭의 수이다. 이 어레이의 클럭 주기를 20ns로 하고  $720 \times 576$  크기의 프레임을 갖는 CCIR 601의 동영상을 기준블럭의 크기(N)이 16, 최대탐색거리(p)가 8인 경우의  $f_{\max}$ 는 120 Hz이다.

$$\begin{aligned} f_{\max} &= \frac{1}{\text{clock period} \times \{ N^2 + 8p^2 - 2 + (N_v \times N_h - 1) \times N^2 \}} \tag{14} \\ &= 120 \text{ Hz} \end{aligned}$$

동영상 표준인 CCIR 601은 초당 30개의 프레임으로 구성된다<sup>[7]</sup>. 따라서 제안한 어레이는 CCIR 601의 조건을 만족하며, 더 높은 해상도나 초당 프레임 수를 요구하는 동영상에도 적용 가능하다.

#### 3. 입력 포트의 수

제안한 일차원 시스톨릭 어레이는 가장자리에 위치한 하나의 처리요소를 통해서만 데이터를 입력받고, 이프라인을 통해 재사용한다. 이 처리요소에는 탐색영

역을 구성하는 홀수와 짝수번째 열의 데이터를 각각 입력하는 두 포트와 기준블럭의 데이터를 입력하는 포트를 갖는다. 즉 어레이의 전체 입력 포트의 수( $N_{IP}$ )는 식 (15)와 같다.

$$N_{IP} = 3 \tag{5}$$

4. 어레이의 특성과 성능비교

블럭정합 알고리즘의 고속처리를 위한 제안된 시스템 어레이의 처리요소는 일차원으로 연결되어 있으며, 어레이를 구성하는 모든 경로는 시간 및 공간적으로 지역성을 갖는다. 또한 기준블럭 또는 탐색영역의 크기가 확장되는 경우 제안된 어레이의 모듈러 확장을 통한 움직임 추정이 가능하다.

표 1은 기존의 연구에서 제안된 구조와 본 논문에서 제안하는 구조의 성능을 비교한 것이다. 기준블럭의 크기가  $N \times N$ 이고, 최대탐색거리가  $p$ 인 경우에 대해 비교하였다. 본 논문에서 제안한 블럭정합 알고리즘의 고속처리를 위한 시스템 어레이는 입력포트와 처리요소의 수, 계산시간 및 연결성에서 우수한 성능을 보인다. 또한 어레이를 구성하는 모든 패스가 시간적·공간적으로 지역성을 가지므로 브로드캐스팅과 롱패스를 갖는 기존의 구조에 비해 동기를 맞추기 위한 클럭의 속도를 향상시킬 수 있고, 전력소비 및 발열량이 적은 장점을 얻을 수 있다.

표 1. 성능비교

Table 1. The comparison of performance.

Architecture	Num of Input Ports	Num of Processing Elements	Proc. Time per Block	Cascadability	draw-back
T. Komarek [3]	$N$	$N^2 + N + 1$	$\frac{(2p+1)}{(N+2p)}$	No	Low Speed
K. M. Yang [4]	3	$N$	$N^2(2p)$	Yes	Low Speed
C. H. Hsieh [5]	2	$N^2$	$(N+2p)^2 + 1$	No	Low Speed
H. G. Yea [7]	3	$N^2$	$N^2$	Yes	Broadcasting
Y. K. Lai [8]	3	$(2p)^2$	$N^2$	Yes	Broadcasting
Kittitornkun [9]	3	$(2p+1)^2$	$N^2$	Yes	Long path
Proposed Architecture	3	$(2p)^2$	$N^2$	Yes	

IV. 결론

본 논문에서는 움직임추정을 위한 완전탐색 블럭정합 알고리즘의 고속처리를 위한 일차원 시스템 어레이의 구조를 제안했다. 제안된 일차원 시스템 어레이에서 데이터와 제어신호의 전달을 위한 경로는 시간 및 공간적 지역성을 갖는다. 따라서 브로드캐스팅이나 롱패스를 갖는 기존의 구조에 비해 더 높은 클럭에 동기를 맞추어 동작시킬 수 있으며, 전력소비와 발열량이 적은 장점을 얻을 수 있다. 제안된 시스템 어레이 구조는 기준블럭과 탐색영역에 대한 SAD를 연산하는  $(2p)^2$ 개의 처리요소로 구성되며, 기준블럭과 탐색영역의 데이터를 입력하기 위한 3개의 입력 포트를 갖는다. 데이터의 입출력 핀은 일차원 어레이의 시작과 끝부분의 처리요소에만 존재한다. 이 구조는  $N^2$  시간마다 하나의 기준블럭에 대한 이동벡터를 구할 수 있다. 어레이를 구성하는 처리요소의 내부구조가 간단하고 데이터전달이 규칙적이고 모듈화되어 시스템의 확장에 용이하다. 또한 기준블럭의 크기나 최대탐색거리가 확장된 응용에 대해 제안된 어레이를 연결하여 움직임 추정기를 구성할 수 있다. 본 연구의 결과는 동영상압축을 위한 완전탐색 블럭정합 알고리즘의 고속처리를 위한 움직임 추정기의 구현에 이용될 수 있을 것이다.

참고 문헌

[1] Vijay K. Madiseti, VLSI digital signal processors, Butterworth-Heinemann, 1995.  
 [2] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, D. J. Legall, MPEG video compression standard, Chapman and Hall, 1996.  
 [3] T. Komarek, P. Pirsch, "Array architecture for Block matching algorithms," IEEE Trans. on Circuits and Systems, Vol. 36, No. 10, pp. 1301-1308, Oct. 1989.  
 [4] K. M. Yang, M. T. Sun, L. Wu, "A Family of VLSI Designs for the Motion Compensation Block Matching Algorithm," IEEE Trans. on Circuits and Systems, Vol. 36, No. 10, pp. 1317-1325, Oct. 1989.



- [5] C. H. Hsieh, T. P. Lin, "VLSI architecture for block-matching motion estimation algorithm," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 2, No. 2, pp. 169-175, June. 1992.
- [6] S. B. Pan, S. S. Chae, and R. H. Park, VLSI architectures for block matching algorithm, IEEE Trans. Circuits and Systems for Video Technology, vol. 6, pp. 67-73, Jan. 1995.
- [7] H. G. Yea, Y. H. Hu, "A novel modular systolic array architecture for full-search block matching motion estimation," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 5, No. 5, pp. 407-416, Oct. 1995.
- [8] Y. K. Lai, Y. L. Lai, Y. C. Liu, P. C. Wu, L. G. Chen, VLSI implementation of the motion estimator with two-dimensional data-reuse, IEEE Trans. on Consumer Electronics, Vol. 44, No. 3, pp. 623-628, AUG. 1998.
- [9] S. Kittitornkun and Y. H. Hu, Frame-level pipelined motion estimation array processor, IEEE Trans. Circuits and Systems for Video Technology, vol. 11, No. 2, pp. 248-251, Feb. 2001.
- [10] S. Y. Kung, VLSI array processors, Prentice Hall, Englewood Cliffs, NJ, 1988.

---

 저 자 소 개
 

---



李守鎭(正會員)

1995년 2월 : 부산수산대학교 전자공학과 졸업(공학사). 1997년 2월 : 부경대학교 대학원 전자공학과 졸업(공학석사). 2002년 8월 : 부경대학교 대학원 전자공학과 박사 졸업(공학박사) <주관심분야 :

병렬처리, 영상압축, 컴퓨터구조>

馬鍾鎬(正會員) 第39卷 SC編 第3號 參照