

완전탐색 블럭정합 알고리즘의 고속 처리를 위한 VLSI 어레이 프로세서의 구조

정회원 이 수 진*, 우 중 호**

A VLSI Array Processor Architecture for High-Speed Processing of Full Search Block Matching Algorithm

Su-jin Lee*, Chong-ho Woo** *Regular Members*

요 약

본 논문에서는 완전탐색 블럭정합 알고리즘의 고속처리를 위한 VLSI 어레이 프로세서의 구조를 제안한다. 완전탐색 블럭정합 알고리즘으로부터 인덱스 공간을 확장한 단일할당코드를 변환 후, 이것으로부터 데이터의존그래프를 구하고, 최적의 방향으로 투영시켜 신호흐름그래프를 얻는다. 신호흐름그래프에 시간 및 공간적인 지역성을 추가하여 이차원 VLSI 어레이를 구하였다. 탐색영역의 후보블럭이 행과 열로 중첩되므로, 중복되는 데이터를 재사용해서 데이터 입력횟수를 줄이고 처리 속도를 향상시켰다. 블럭의 크기가 N 이고 최대탐색거리가 p 인 경우, 제안한 VLSI 어레이의 처리요소는 $(N^2+1) \times (2p+1)$ 개이고, 입력포트는 $(N+2p)$ 개이다. 첫 번째 기준블럭에 대한 이동벡터를 구하는 시간은 $(N^2+2(p+1)N+6p)$ 이고, 매 $(3N+4p-1)$ 단위시간마다 다음 기준블럭에 대한 이동벡터가 구해진다.

ABSTRACT

In this paper, we propose a VLSI array architecture for high speed processing of FBMA. First of all, the sequential FBMA is transformed into a single assignment code by using the index space expansion, and then the dependance graph is obtained from it. The two dimensional VLSI array is derived by projecting the dependance graph along the optimal direction. Since the candidate blocks in the search range are overlapped with columns as well as rows, the processing elements of the VLSI array are designed to reuse the overlapped data. As the results, the number of data inputs is reduced so that the processing performance is improved. The proposed VLSI array has $(N^2+1) \times (2p+1)$ processing elements and $(N+2p)$ input ports where N is the block size and p is the maximum search range. The computation time of the first reference block is $(N^2+2(p+1)N+6p)$, and the block pipeline period is $(3N+4p-1)$.

1. 서 론

동영상 데이터는 많은 양의 시간적, 공간적 중복성을 갖고 있다. MPEG 등의 동영상 압축 알고리즘은 공간적인 중복성과 시간적 중복성을 제거하기 위하여 DCT (Discrete Cosine Transform)와 움직임 보상(motion compensation)을 각각 사용한다.

연속되는 프레임들 사이의 시간적 중복성을 제거

하기 위한 움직임보상을 구현하기 위해 움직임 추정(Motion Estimation)이 필요하다.^[1] 움직임 추정을 위해 완전탐색, 3단계 탐색, 로그 탐색, 공액방향 탐색, APD(Alternating Pixel-Decimation) 탐색, SAPD(Subsampled Motion-Field Search with Alternating Pixel-Decimation Patterns) 탐색 등의 다양한 알고리즘이 제안되었다.^[2] 여러가지 알고리즘 중 완전탐색 블럭정합 알고리즘(FBMA : Full Search Block Matching Algorithm)이 가장 우수한

* 부경대학교 전자공학과 (geeny21@mail1.pknu.ac.kr), ** 부경대학교 전자컴퓨터정보통신공학부 (chwoo@pknu.ac.kr)

논문번호 : 010156-0627, 접수일자 : 2001년 6월 27일

화질을 제공하므로 가장 널리 사용되지만 계산량이 과다해서 실시간 처리가 불가능한 단점이 있다. 이러한 단점을 극복하기 위하여 알고리즘의 고속처리를 위한 VLSI 어레이들이 제안되어왔다.

T. Komarek 등은 완전탐색 블럭정합 알고리즘 구현을 위한 기본적인 VLSI 구조를 제안했다.^[3] Yang 등은 1차원 세미시스톨릭 어레이를 제안했으나 속도의 제한으로 인하여 HDTV 등의 고속처리에는 적합하지 않다.^{[4][5]} C. H. Hsieh 등은 쉬프트 레지스터를 이용하여 순차적인 데이터 입력이 가능한 시스톨릭 어레이를 제안했으나 데이터의 활용도가 낮고 각 열에서 계산된 값을 누적하기 위한 계산기의 회로가 복잡한 단점이 있다.^[6] Y. S. Jehng 등은 트리구조의 VLSI를 제안하였으나 입출력핀 수가 너무 많아 실제 구현에 어려움이 있다.^[7] H. G. Yea 등은 인접하는 기준블럭에 대한 탐색영역의 일부가 중복되는 성질을 이용하여 입출력핀 수가 적으며 계산속도를 향상시킬 수 있는 구조를 제안했다.^[8] 그러나 처리요소에 값을 전달하기 위해 브로드캐스팅과 전역패스(global path)가 존재하므로 VLSI의 동작속도를 향상시키는데 한계가 있다.

본 논문에서는 S. Y. Kung의 시스톨릭 어레이 설계방법에 따라 완전탐색 블럭정합 알고리즘의 처리 속도 향상을 위한 VLSI 어레이를 설계하였다.^[9] 알고리즘으로부터 병렬알고리즘 표현법 중 하나인 단일할당코드로 변환하고, 데이터의 의존관계를 나타내는 데이터의존그래프를 구한다. 데이터의존그래프를 최적의 방향으로 투영하여 신호흐름그래프를 구하고, 각 처리요소가 시간 및 공간적 지역성을 갖도록 변형하여 시스톨릭 어레이를 유도한다. 따라서 브로드캐스팅과 전역패스는 갖는 기존의 구조에 비해 높은 클럭에서 동작 가능하다. 제안한 VLSI 어레이의 구조는 탐색영역(search range) 내의 후보블럭(candidate block)을 구성하는 픽셀들이 인접하는 후보블럭들에서 중복되므로 이러한 데이터를 재사용함으로써 데이터 입력의 횟수를 줄이고 처리속도를 향상시켰다.

II. 완전탐색 블럭정합 알고리즘

블럭정합 알고리즘은 기준블럭(reference block)에 대한 이동벡터(motion vector)를 구하는 것이다. 기준블럭이란 그림 1과 같이 현재 프레임의 $N \times N$ 픽셀 크기로 나눈 각각의 블럭이다. 각 기준블럭에 대한 이동벡터란 기준블럭에 정합하는 이전 프레임의

블럭으로의 쉬프트된 거리이다. 이전 프레임의 탐색영역은 기준블럭의 좌표를 기준으로 $\pm p$ 만큼 떨어진 블럭들로 구성된다. 완전탐색 블럭정합 알고리즘은 탐색영역의 모든 후보블럭에 대해 탐색하는 방식이다.

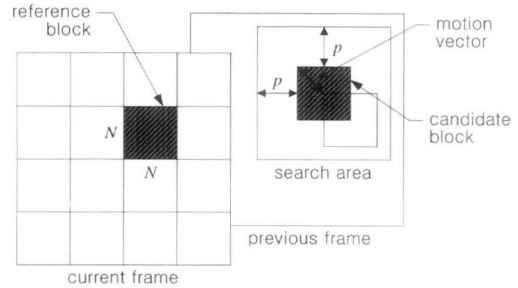


그림 1. 완전탐색 블럭정합 알고리즘의 개념

기준블럭과 정합되는 블럭은 후보블럭들 중 기준블럭과의 오차가 가장 작은 블럭으로 선택된다. 이때의 인덱스 차이가 이동벡터이며 식(1)을 만족한다.

$$MV = (m, n)_{\min(SAD(m, n))} \quad (1)$$

$$\begin{aligned} \text{where } SAD(m, n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i, j) - y(i+m, j+n)| \\ &, -p \leq m, n \leq p \end{aligned}$$

이때 $x(i, j)$ 는 기준블럭의 데이터를 나타내고, $y(i, j)$ 는 후보블럭의 데이터를 나타낸다. 블럭의 오차를 측정 방법은 SAD(Sum of Absolute Difference)를 이용한다.

III. 어레이의 설계

1. 알고리즘과 단일할당코드의 유도

시스톨릭 어레이는 모듈성, 규칙성이 있으며, 각 처리요소간에 시간 및 공간적인 지역성을 가지며 상호접속된 고도의 파이프라인 및 동기화 다중처리기이다.^[9] S. Y. Kung이 제안한 방법으로 시스톨릭 어레이를 설계하기 위해서는 알고리즘으로부터 단일할당코드를 구하고, 데이터의 의존성을 표현한 데이터의존그래프를 투영시켜 신호흐름그래프를 유도한다. 이 때의 투영방향은 생성된 처리요소의 구조가 간단하고 입출력핀의 수가 적은 방향으로 선택한다. 신호흐름그래프를 시간 및 공간적인 지역성을 갖도록 변형하여 시스톨릭 어레이를 얻는다.

```

for (v=0 ; v<=Nv-1 ; v++) {
  for (h=0 ; h<=Nh-1 ; h++) {
    MV(h,v) = (0,0)
    Dmin(h,v) = ∞
    for (m=-p ; m<=p ; m++) {
      for (n=-p ; n<=p ; n++) {
        SAD(m,n) = 0
        for (i=0 ; i<=N-1 ; i++) {
          for (j=0 ; j<=N-1 ; j++) {
            SAD(m,n) += | x(i,j)-y(i+m,j+n) |
          }
        }
        if (Dmin(h,v) > SAD(m,n)) {
          Dmin(h,v) = SAD(m,n)
          MV(h,v) = (m,n)
        }
      }
    }
  }
}

```

그림 2. 6-레벨 중첩된 루프로 구성된 블럭정합 알고리즘

그림 2는 완전탐색 블럭정합 알고리즘으로서 블럭의 크기가 $N \times N$ 이고, 최대탐색거리가 p 인 경우에 식 (1)을 처리한다. 변수 x 와 y 는 각각 기준블럭과 탐색영역의 픽셀을 나타내고, 블럭의 차를 누적한 값은 SAD에 저장한다. 그리고 최소의 SAD 값을 Dmin에 저장하고, 이때의 후보블럭의 인덱스 m, n 을 이동벡터로 선택한다.

그림 2의 알고리즘을 인덱스 확장을 통해 단일할당코드로 변환하면 그림 3과 같다. 단일할당코드는 각 변수에 단 한번만 대입시키는 알고리즘의 표현 형태로서 각 변수들이 중속성을 갖지 않으므로 알고리즘을 병렬로 처리할 수 있다. 탐색영역의 좌우 및 상하로 인접한 후보블럭들은 중첩되는 데이터를 포함한다. 이러한 중복 데이터를 파이프라인을 통해 재사용해서 데이터 입력횟수를 줄이고 파이프라인의 처리율을 높인다.

2. 데이터의존그래프의 유도

하나의 기준블럭에 대한 그림 3의 단일할당코드로부터 데이터의 의존성을 표현한 데이터의존그래프는 그림 4와 같다. 여기서 아크(arc)는 식(2)와 같이 5가지가 존재한다.

```

for (v=1 ; v <=Nv ; v++) {
  for (h=1 ; h<=Nh ; h++) {
    MV(h,v) = (0,0)
    Dmin(h,v) = ∞
    for (m=-p ; m<=p ; m++) {
      for (n=-p ; n<=p ; n++) {
        SAD(m,n,0) = 0
        for (k=0 ; k<=N*N-1 ; k++) {

```

```

          i = k / N
          j = k % N
          if ((j%N==0 && m==1)
              || n==1 || n==N*N) {
            y(m,n,k) = y(m,0,k)
          }
          else if (j%N==0 && m>1) {
            y(m,n,k) = y(m-1,n,k+N)
          }
          else {
            y(m,n,k) = y(m,n-1,k+1)
          }
          x(m,n,k)=x(m,n-1,k)
          SAD(m,n,k+1) = SAD(m,n,k)
            + |x(i,j)-y(i+m,j+n)|
        }
      }
      if (Dmin(m,n) > SAD(m,n,N*N)) {
        Dmin(m,n) = SAD(m,n,N*N)
      }
      if (Dmin(m,n) > Dmin(m,n-1)) {
        Dmin(m,n) = Dmin(m,n-1)
      }
      if (Dmin(m,2p+1) > Dmin(m-1,2p+1)) {
        Dmin(m,2p+1) = Dmin(m-1,2p+1)
        MV(h,v) = (m,n)Dmin(2p+1,2p+1)
      }
    }
  }
}

```

그림 3. BMA의 단일할당코드

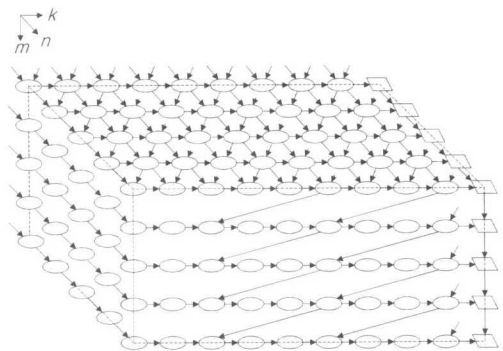


그림 4. 3차원 데이터의존그래프 (N=3, p=2)

$$e = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & -N & 0 \end{bmatrix} \begin{matrix} m \\ n \\ k \end{matrix} \quad (2)$$

그림 4는 블럭의 크기 N 이 3, 최대탐색거리 p 가 2인 경우에 해당하는 3차원 데이터의존그래프이다. 일반적인 동영상 표준안에서는 N 이 16, p 가 16 또는 8의 값을 갖는다. 원을 표현된 노드는 SAD를 구하는 기능을 갖고, 사각형으로 표현된 노드는 구해진 SAD 중 최소의 값을 갖는 후보블럭과 이에 해당하는 이동벡터를 구한다. 이 데이터의존그래프

는 $(2p+1)$ 개의 층(layer)으로 구성되어 있다. 각 층은 위층으로부터 중복되는 탐색영역의 픽셀을 전달받아 다음 블록의 연산에 재사용한다.

그림 5는 데이터의존그래프의 한 층을 표현한 것으로 각 층은 $N \times N$ 개의 열과 $(2p+1)$ 개의 행으로 구성된다. 가장자리의 노드와 N 번째 노드에서 탐색영역의 값을 입력받고, 이 값을 $[n \ k]=[1 \ -1]$ 방향으로 전달하여 재사용한다. 가장자리의 노드를 통해 기준블럭의 값을 입력받아 $[1 \ 0]$ 방향으로 전달하고 계산된 SAD 값을 $[0 \ 1]$ 방향으로 전달된다. 각 층별로 SAD 값을 비교해서 가장 작은 값을 선택하고, 다음 층으로 전달하여 전체 탐색영역 중 최적인 후보블럭을 선택한다.

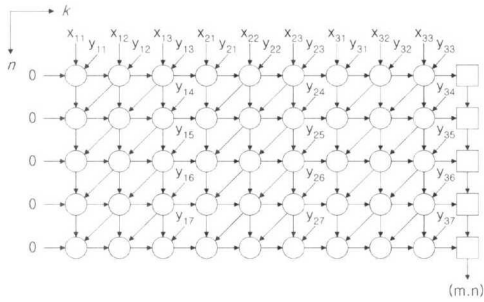


그림 5. 데이터의존그래프의 한 층 ($N=3, p=2$)

3. 시스톨리어레이의 유도

정확한 계산의 순서를 유지하며 시간 및 공간적인 지역성을 만족하기 위해 스케줄 벡터 \vec{s} 와 투영 벡터 \vec{d} 는 식 (3)과 (4)를 만족해야 한다.¹⁹⁾

$$\vec{s}^T \vec{e} > 0 \quad (3)$$

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

식 (3)을 만족하는 최적의 스케줄 벡터 \vec{s}^T 는 $[N+1 \ 2 \ 1]^T$ 이 된다. 투영벡터 \vec{d} 의 선택기준은 식 (4)를 만족하면서 투영된 결과의 신호흐름그래프 또는 시스톨릭 어레이의 처리요소 수가 적고, 입출력 핀수가 적으며, 처리요소의 내부구조가 간단한 방향을 선택한다. 투영벡터 \vec{d} 를 식 (5)와 같으면 파이프 라인 주기 α 는 식 (6)과 같이 구해진다. 따라서 데이터의 입력 사이에 한단위의 지연시간이 요구된다.

$$\vec{d} = [0 \ 1 \ 0] \quad (5)$$

$$\alpha = \vec{s}^T \vec{d} = 2 \quad (6)$$

데이터의존그래프에 식 (5)의 투영벡터를 적용하면 그림 6의 시스톨리어레이가 구해진다. 기준블럭의 데이터는 각 처리요소에 적재되어 있고, 탐색영역의 데이터가 입력되어 인접 처리요소로 전달되며 SAD를 계산한다. 각 층별로 최소의 SAD를 선택하고, 마지막 층에서 전체 탐색영역의 SAD 중 최소인 경우의 이동벡터 (m,n) 를 구한다.

4. 데이터 패스의 지역화

VLSI 어레이의 구현에서 교차하는 패스가 존재하면 평면구조의 VLSI에서 레이아웃(layout) 패턴이 복잡해지고, 이에 따른 결함의 증가로 인한 비용이 증가될 수 있다. 또한 전역패스는 신호의 전달지연을 초래하여 고속화의 장애 요인이 될 수 있다. 따라서 VLSI 어레이구조에서는 가능한 이러한 패스들을 제거하고 지역화시킬 필요가 있다.

그림 6의 VLSI 어레이구조는 탐색영역을 전달하기 위한 전역패스가 존재하며, 기준블럭을 전달하기 위한 패스와 교차가 발생한다. 탐색영역의 값을 전달하기 위한 $[m \ k]=[1 \ -N]$ 의 전역패스를 제거하기 위해 그림 7과 같이 각 행을 오른쪽으로 쉬프트시켜 탐색영역의 값을 $[1 \ 0]$ 방향으로 전달되도록 한다. 어레이의 각 행은 동일한 기준블럭의 값을 선적 재한다. 기준블럭의 값을 $[1 \ N]$ 방향의 전역패스를 이용하지 않고 주변의 처리요소로 전달시켜 전역패스와 패스의 교차를 제거한다. 어레이의 첫 번째 행을 구성하는 처리요소 A는 호스트 컴퓨터로부터 기준블럭과 탐색영역의 데이터를 입력받는다. 이때 입력핀의 수를 감소시키기 위해 기준블럭과 탐색영역의 데이터의 입력핀을 공유한다. 따라서 기준블럭의 내용을 먼저 입력하여 처리요소 내에 저장시킨 후 탐색영역의 데이터를 입력받는다. 입력받은 기준블럭의 데이터를 $[0 \ -1]$ 방향으로 전달하며 저장시킨다. 처리요소 A에 저장된 기준블럭의 데이터를 각

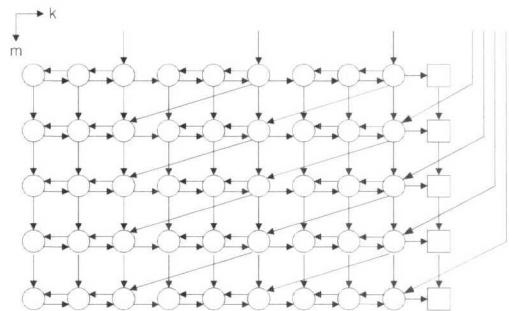


그림 6. 2차원 시스톨리어레이 ($N=3, p=2$)

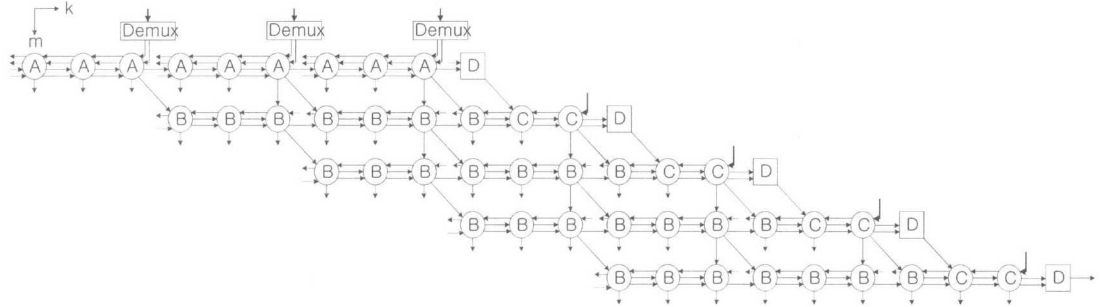


그림 7. 지역화된 어레이 구조 ($N=3, p=2$)

N 번째 열에서 [1 1] 방향으로 다음 행으로 전달한다. 탐색영역의 데이터는 [0 -1] 방향으로 전달되며 SAD를 계산한다. 두 번째 행에서부터 처리요소 B는 기준블럭의 값을 [0 1] 방향으로 전달하는 것이 처리요소 A와 차이가 있다. 처리요소 D에서 각 행별로 계산된 SAD 중 최소값을 선택한다. 각 행별로 선택된 SAD 값은 [1 1] 방향에 위치한 처리요소 C를 통해서 다음 행으로 전달된다. 처리요소 C는 처리요소 B의 구조에 상위 행으로부터 전달된 SAD를 전달하는 기능을 포함하고 있다. 그림 6의 구조를 그림 7과 같이 데이터 전달 패스의 교차와 전역패스를 제거하더라도 동작의 정확성과 계산시간에는 영향을 미치지 않는다.

그림 7의 어레이는 데이터의 전달이 규칙적이고 모듈성이 있어 시스템 확장이 용이하다. 기준블럭의 크기가 커지면 VLSI 어레이의 각 행을 구성하는 처리요소를 추가하고, 탐색거리가 증가하면 어레이를 구성하는 행을 추가하여 쉽게 확장할 수 있다.

5. 처리요소의 내부구조

제한한 어레이는 그림 7과 같이 4 종류의 처리요소들로 구성되며 각 처리요소의 내부구조는 그림 8과 같다.

(1) 처리요소 A

그림 8-(a) 처리요소 A는 어레이의 첫 번째 행을 구성하며, 호스트 컴퓨터로부터 기준블럭과 탐색영역의 데이터를 입력받아 SAD를 구한다. 음영으로 표현된 사각형은 레지스터를 의미한다. 기준블럭과 탐색영역의 데이터가 레지스터 x 와 y 에 각각 저장되며, 두 값 차의 절대값을 누적하여 레지스터 SAD에 저장한다. 다음 행의 처리요소들로 기준블럭의 값을 전달하는 순서를 맞추기 위해 레지스터 x' 이 추가되었다.

(2) 처리요소 B

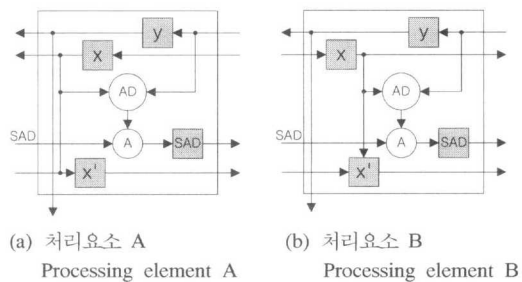
VLSI 어레이의 두 번째 행부터는 그림 8-(b)와 (c)의 처리요소 B와 C가 SAD를 연산하는 기능을 수행한다. 처리요소 A가 기준블럭의 값을 레지스터 x 를 통해 [0 -1] 방향으로 전달하고 레지스터 x' 를 통해 [0 1] 방향으로 전달시킨다. 처리요소 B는 레지스터 x 의 내용을 [0 1] 방향으로 전달시키고 레지스터 x' 를 통해 다음 행으로 기준블럭의 값을 전달한다.

(3) 처리요소 C

처리요소 C는 처리요소 B의 구조에 이전 행에서 최소 SAD 값으로 선택한 D_{min} 값을 전달하기 위한 레지스터 D_{min} 이 추가되었다.

(4) 처리요소 D

그림 8-(d)의 처리요소 D는 누적된 SAD 중 최적의 값을 선택하는 기능을 갖는 처리요소이다. 각 행별로 최소의 SAD를 선택해서 레지스터 D_{min} 에 저장하고, 다음 행의 처리요소 C를 통해서 전달한다. 그림 8-(d)에는 D_{min} 에 대한 이동벡터를 구하는 부분은 생략이 되어 있다. 어레이의 최하행의 처리요소 D에서 선택된 D_{min} 에 해당하는 인덱스가 이동벡터로 선택된다.



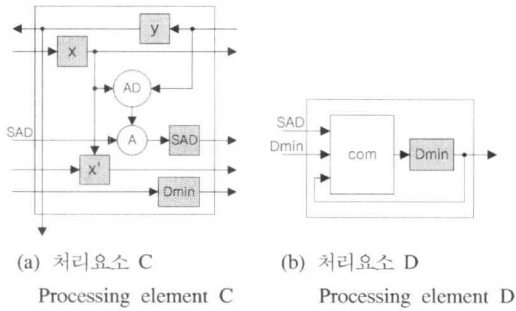


그림 8. 처리요소의 내부 구조

IV. 결과 및 고찰

설계한 VLSI 어레이의 성능을 평가하기 위한 주요 평가측도에는 처리요소의 수, 계산시간, 입력포트의 수 등이 있다.

(1) 처리요소의 수

제한한 VLSI 어레이의 각 행은 한쌍의 기준블럭과 후보블럭의 SAD를 연산하기 위한 N^2 개의 처리요소와 최소의 SAD를 선택하기 위한 하나의 처리요소가 필요하다. 전체 탐색영역에 대해 데이터를 재사용하면서 최적의 이동벡터를 찾기 위해 $(2p+1)$ 개의 행이 필요하다. 따라서 프레임을 $N \times N$ 크기의 기준블럭으로 나누고 최대탐색거리가 p 로 주어질 경우 어레이를 구성하는 처리요소의 수는 식 (7)과 같다.

$$\text{Number of PEs} = (N^2 + 1) \cdot (2p + 1) \quad (7)$$

(2) 계산시간

첫 번째 기준블럭에 대해 초기 지연시간을 고려한 이동벡터를 구하는 시간 T 와 블럭 파이프라인 주기 β 는 각각 식 (8)과 (9)와 같다.

$$T = (N^2 + 2pN + 6p + 2N) \quad (8)$$

$$\beta = 3N + 4p - 1 \quad (9)$$

따라서 식 (8)의 T 단위시간에 첫 번째 기준블럭에 대한 이동벡터가 구해지고, 식 (9)의 매 β 단위시간마다 다음 기준블럭에 대한 이동벡터가 구해진다.

(3) 입력포트의 수

VLSI의 첫 번째 행에는 기준블럭과 탐색영역의 데이터를 입력하기 위한 N 개의 입력포트가 존재하

고, 다음 행부터는 각각 탐색영역의 데이터를 입력하기 위한 하나의 입력포트가 필요하다. 즉, VLSI 어레이의 입력포트의 수는 식 (10)과 같다.

$$\text{Number of I/O ports} = (N + 2p) \quad (10)$$

표 1. 성능비교 (이미지 크기 : 720×576 , $N=16$, $p=8$)

Type	number of PEs	number of PINs	number of Clock cycles required	
			per first block	per frame
Komarek[3] (AB2)	256	136	496	803,520
Komarek[3] (AS2)	528	392	272	440,640
Yang[4]	16	32	4,069	6,635,520
Hsieh[6]	256	16	961	1,556,820
Jehng[7]	515	2,056	266	414,730
Yea[8]	256	32	256	423,936
proposed method	4369	256	592	128,493

표 1은 720×576 크기의 영상을 16×16 크기의 기준블럭으로 나누고 최대탐색거리 p 를 8로 선택한 경우에 기존의 연구결과와 성능을 비교한 것이다. VLSI 어레이의 성능평가 기준으로 가격대성능비를 평가하는 처리요소의 수(A)와 처리시간(T)의 곱인 AT 와 AT^2 이 많이 이용된다. 특히 어레이의 처리율이 중요한 응용에서는 처리시간(T)에 우선 순위를 둔 AT^2 을 많이 이용한다. 본 논문에서 제안한 VLSI 어레이는 기존의 어레이들에 비해 AT^2 의 기준에서 우수한 성능을 나타낸다. H. G. Yea이 제안한 구조가 본 논문에서 설계한 어레이보다 AT^2 의 평가기준에서 다소 우수하지만 데이터를 동시에 여러 처리요소에 전달하기 위한 전역패스가 존재하며 입력데이터를 브로드캐스팅하므로 VLSI의 동작속도는 높이는 데 제한을 받는다. 그리고 제안한 구조는 파이프라인 주기가 2이므로 인접한 두 처리요소를 통합하여 처리요소의 수를 50% 정도 줄일 수 있다. 본 논문에서 설계한 VLSI 어레이는 하드웨어의 복잡도는 매우 크지만 처리속도면에서는 가장 빠르다. VLSI의 집적기술의 향상으로 인하여 처리속도가 하드웨어 복잡도의 문제보다 중요하게 다루어지고 있다.

V. 결론

본 논문에서는 완전탐색 블럭정합 알고리즘의 고속 처리를 위한 2차원 VLSI 어레이의 구조를 설계

했다. 기준블럭에 대해 중복되는 탐색영역의 데이터를 효율적으로 재사용해서 데이터 입력횟수를 줄여서 처리속도를 향상시켰다. 기존의 제안된 어레이 구조에 비해서 요구되는 처리요소의 수는 증가했지만 처리속도는 향상되었다. 또한 설계한 어레이의 구조에서 처리요소의 내부가 간단하고 데이터의 전달이 규칙적이고 모듈화되어 있어 시스템의 확장이 용이하다.

본 연구의 결과는 동영상압축을 위한 움직임 추정 알고리즘의 고속처리를 위한 VLSI 어레이의 구현에 이용될 수 있을 것이다.

참 고 문 헌

[1] Yoshihiro Noguchi, Jun Furukawa, Hitoshi Kiya, "A Fast Full Search Block Matching algorithm for MPEG-4 Video," *Proceedings of the 1999 International Conference on Image Processing*, Vol 1, pp. 61-65, Oct. 1999.

[2] S. C. Cheng, H.M. Hang, "A Comparison of Block-Matching Algorithms Mapped to Systolic-Array Implementation," *IEEE Transactions on Circuits & Systems for Video Technology*, Vol.7 No. 5, pp. 741-757, Oct. 1997.

[3] T. Komarek, P. Pirsch, "Array architecture for Block matching algorithms," *IEEE Transactions on Circuits & 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 Transactions on Circuits & Systems*, Vol. 36, No. 10, pp. 1317-1325, Oct. 1989.

[5] 김기현, 김기철, "선형 시스토크 어레이를 이용한 완전탐색 블럭정합 이동 예측기의 구조", *한국통신학회논문지*, Vol. 21, No. 2, pp. 313-325, Feb. 1996.

[6] C. H. Hsieh, T. P. Lin, "VLSI architecture for block-matching motion estimation algorithm," *IEEE Transactions on Circuits & Systems for Video Technology*, Vol. 2, No. 2, pp. 169-175, June. 1992.

[7] Y. S. Jehng, L. G. Chen, T. D. Chiueh, "An efficient and simple VLSI tree architecture for

motion estimation algorithms," *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, pp. 889-899, Feb. 1993.

[8] H. G. Yeo, Y. H. Hu, "A novel modular systolic array architecture for full-search block matching motion estimation," *IEEE Transactions on Circuits & Systems*, Vol. 5, No. 5, pp. 407-416, Oct. 1995

[9] S. Y. Kung, *VLSI array processors*, Prentice Hall, pp. 110-294, 1988.

이 수 진(Su Jin Lee)

정회원



1995년 2월 : 부경대학교 전자공학과 학사
 1997년 2월 : 부경대학교 전자공학과 석사
 1997년 3월 ~ 현재 : 부경대학교 전자공학과 박사과정

<주관심 분야> 어레이 프로세서, 컴퓨터구조, 영상압축

우 종 호(Chong Ho Woo)

정회원

한국통신학회논문지 제26권 제12A호 참조