

# 전탐색 블럭정합 움직임추정 VLSI에서 클럭사이클수를 줄이는 효율적 구조

윤종성\*, 장순화\*\*

\*한국통신 가입자망연구소, \*\*이노플러스

## An Efficient Clock Cycle Reducing Architecture in Full-Search Block Matching Motion Estimation VLSI

\*Jong Seong Yoon, \*\*Soon Hwa Jang

\*Korea Telecom Access Network Lab., \*\*InnoPlus

E-mail : \*jsyoun@kt.co.kr

### 요약문

본 논문은 전탐색 블럭매칭 움직임추정 VLSI 구조에서 클럭당 두연산(하나는 클럭의 상향에지, 하나는 하향에지에서 동작)을 수행하는 PE(Processing Element)를 교번적으로 결선, 클럭의 상향에지는 물론 하향에지에서도 동작하도록 하는 방식으로 클럭 사이클수를 줄이는 VLSI 구조를 제안한다. 기존 구조에 그대로 적용되는 본 방법은 공급 데이터폭이 2 배, PE의 HW 복잡도가 1.5 배, 절대차 합 연산의 복잡도가 2 배로 늘어나 전체 하드웨어가 복잡해지나, PE 수를 2 배로 하여 클럭 사이클수를 줄이는 방법에 비해서는 매우 효율적이다.

본 제안 구조는 계층적 움직임 추정 알고리즘을 사용한 MPEG-2 움직임 추정기 개발의 설계에 적용하여 기능과 HW 복잡도를 확인하였다.

### I. 서론

비디오데이터의 프레임간 상관성을 제거하기 위한 움직임 추정 알고리즘에는 pel-recursive, block-matching, feature-based 기법들이 있으나 block-matching algorithm(블럭정합기법)이 하드웨어 구현상의 이유로 가장 널리 사용되고 있다. 실제 94 년 국제표준으로 제정된 MPEG-2 에서도 블럭정합기법을 비디오 부호화에 채택하였다[1]. 블럭정합기법 역시 많은 연산을 포함하며 이에 따라 지금까지 움직임 추정 연산량을 줄이기 위한 알고리즘 연구[2]와 하드웨어를 단순화하기 위한

VLSI 구조 연구[3]가 많이 이루어져 왔다.

$$SAD(u,v) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |r(i,j) - s(u+i, v+j)|, \quad -d \leq u, v \leq +d, \quad (1)$$

$$V = (u, v) \mid \min SAD(u, v) \quad (2)$$

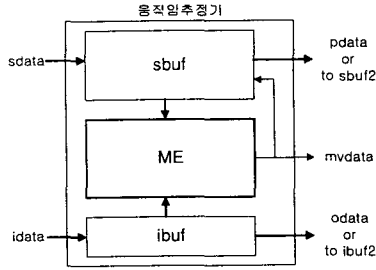
식 1 과 식 2 는 전탐색블럭정합알고리즘(full search block matching algorithm: FBMA) 수식으로서 탐색범위내 (-d ~ +d)의 모든 탐색블럭 중 절대차합(sum of absolute difference : SAD)이 최소인 블럭을 선택한다. 여기서는 하드웨어 단순성으로 SAD 를 정합적으로 사용했다. 이 FBMA 는 최적의 성능을 나타내는 만큼 많은 연산량을 포함하나 연산의 규칙성과 단순성으로 인해 아직까지 많이 사용되고 있다. 본 논문의 구조는 기본적으로 FBMA 의 VLSI 구조에 적용되며 이는 FBMA 를 활용한 계층 탐색 알고리즘 구조 등에도 그대로 적용될 것이다

움직임추정기의 하드웨어 복잡도는 PE(processing element)의 수와 복잡도, 내부 버퍼의 크기 및 구조, 기준블럭데이터 및 탐색영역데이터의 공급 방법등으로 결정되며 이들은 주어진 클럭 사이클수와 밀접한 관련이 있다. 사이클수가 작으면 단위연산(기준블럭·픽셀과 탐색블럭 픽셀간 절대차 계산) 소자인 PE 의 수는 늘려야 하며 또한/혹은 내부버퍼의 구조 및 데이터 공급의 복잡도 증가를 감수하고 PE 의 연산효율을 높여야 한다.

본 논문은 단위연산 대신 두연산(하나는 클럭의 상향에지에서 동작, 하나는 하향에지에서 동작)을 수행하는 PE 를 교번적으로 결선하여 클럭의 상향에지는 물론 하향에지에서도 동작시켜 마치 기존 구조가 클럭의 양(both) 에지에 따라 shift 하면서 동작하는 것처럼 하는 방식으로 클럭을 반으로 줄인 구조를 제시한다.

## II. 움직임 추정 VLSI 구조

그림 1은 움직임추정기의 기본 구성도로서 탐색영역데이터(sdata)와 기준블럭데이터(idata)를 입력으로 받아 움직임벡터(mvdata)와 예측블럭데이터(pdata), 그리고 idata의 지연데이터인 odata를 출력한다.

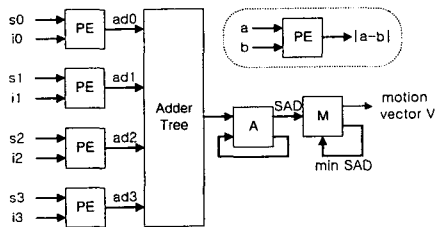


(그림 1) 움직임추정기의 기본 구성도

그림 1에서 sbuf는 이전 블록에서 사용되었던 sdata를 저장, 현재 블록에서는 새로운 sdata만 입력함으로써 sdata의 입력 데이터율을 줄여주며 ME의 구조에 따른 다양한 데이터 요구에 쉽게 대응하도록 해 준다. 한편 ibuf는 sbuf와 같이 데이터를 완충 역할도 하지만 idata를 지연시켜 pdata와 동시에 odata로 출력시켜 주기 위해 필요하다. ME블록은 실제 움직임추정 연산이 이루어지는 곳으로 VLSI 구조에 따라 탐색데이터나 기준데이터를 클럭당 한 데이터씩 혹은 여러 데이터씩, 그리고 동일한 데이터를 한번만 혹은 여러 번 요구한다.

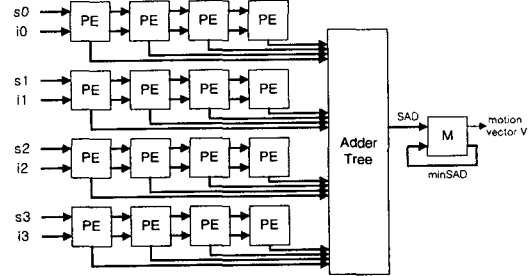
ME 구조는 PE들의 배열로서 [탐색블럭수\*기준블럭데이터수]만큼의 연산을 수행해서 최적의 블럭을 찾아주는 하드웨어 구성이다. 보통 클럭 사이클수는 전체연산수보다 작아 여러개의 PE를 두어 병렬 처리한다.

그림 2는 1차원 배열의 PE열 구조이다. 이 구조의 장점은 PE의 연산 효율이 100%라는 것이다. 그러나 이 구조는 sdata 및 idata를 클럭당 PE수만큼 데이터를 공급해 줌에 따라 ibuf와 sbuf의 버퍼구조 및 공급회로가 복잡해진다. 따라서 PE수가 많은 경우 적절치 않다.



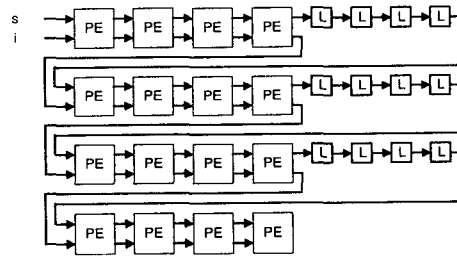
(그림 2) 1차원 배열의 PE열 구조(N=4)

그림 3은 idata와 sdata의 대역폭을 늘리지 않고 PE의 수를 늘려주는 2차원 PE열 구조이다. 네 클럭동안 idata와 sdata가 PE의 내부 래치로 로딩된다. 이후 idata는 그대로 PE에 래치되어 있고 sdata만 shift right하면서 절대차연산을 수행한다. 이 구조의 단점은 로딩이라는 클럭낭비가 있으며 PE열로의 데이터 공급폭이 2차원 PE열의 수직수로 여전히 크다는 것이다.



(그림 3) 2차원 배열의 PE열 구조(N=4)

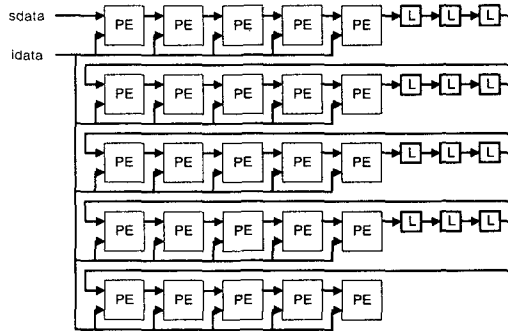
그림 4는 데이터의 공급을 단순화시킨 2차원 구조로서 NxN개의 PE와 (2d)x(N-1)개의 래치를 두고 있다. idata는 NxN 클럭동안 입력되어 각 PE에 로딩되고 sdata는 한번씩 마지막 sdata가 입력됨과 동시에 움직임추정 연산이 완료된다. 이 구조는 데이터 입력 구조가 단순하나 많은 래치와 로딩클럭을 필요로 하는 단점이 있다.



(그림 4) 2차원 배열의 PE열 구조(N=4, d=2)

그림 5는 그림 4와 비슷하나 PE가 절대차합 연산까지 수행하여 각 탐색블럭의 SAD 연산을 담당하는 것이 차이이다. 모든 sdata가 입력되는 순간 모든 탐색블럭의 SAD가 구해지나 그림 5 구조는 각각의 PE에 들어있는 SAD를 하나씩 뽑아내어 최적의 탐색블럭을 찾는 클럭 사이클이 소모된다. 반면 그림 4의 구조에서는 클럭당 하나의 탐색블럭의 SAD가 구해지면서 동시에 최적 탐색블럭 비교가 이루어진다. 그림 5의 구조에서의 PE 수

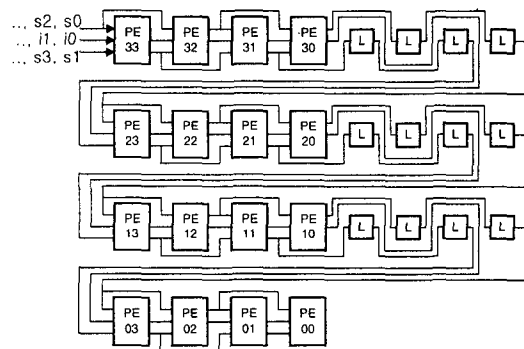
는 탐색블럭수와 관계되며 래치의 수는 수평기준블럭데 이타수와 수직탐색블럭수로 정해진다. 따라서 그림 5 구조는 MPEG-2 에서 정수 화소 단위의 움직임 추정 후 에 이루어지는 반화소 단위 움직임 추정과 같이 탐색블 락의 수가 적은 ME 에 적절하다.



(그림 5) 2 채널 배열 구조(N=4, d=2, PE 수=탐색블럭수)

### III. 제안한 움직임 추정 VLSI 구조

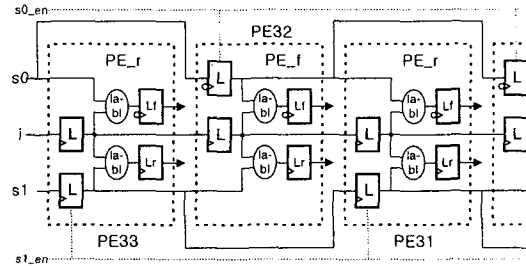
그림 2 나 3 은 클럭의 연산 효율은 좋으나 데이터 공급이 복잡하며 그림 4 와 5 는 데이터공급이 단순하나 소요 사이클수가 많다는 단점이 있다. 본 논문은 이들 구조에 PE 의 구조와 결선을 변경함으로써 소요 사이클 수를 반으로 줄인 구조로서 그림 6 은 그림 4 에 제안방 법을 적용한 예이다. PE 수와 래치(L)의 수, 그리고 배치 는 그림 4 와 동일하며 단지 sdata 가 두개씩 입력되고 그것이 교번적으로 전파되는 것이 차이다.



(그림 6) 그림 4 에 제안방법을 적용한 예(N=4, d=2)

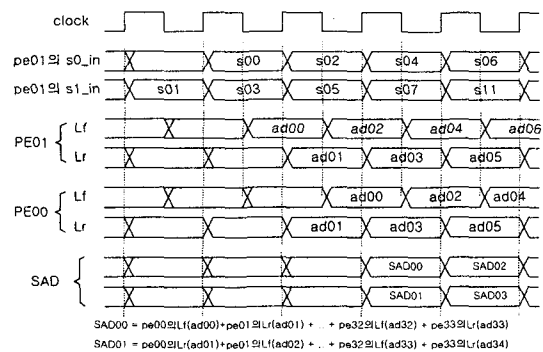
그림 7 은 제안방법의 구체적인 PE 의 내부구조와 PE 열 결선이다. PE 에는 sdata 를 클럭의 상향에지에 래 치하는 PE\_r 과 하향에지에 래치하는 PE\_f 두 종류가

있으며 이들 모두 절대차 결과의 하나는 하향에지에서 하나는 상향에지에서 래치한다. 그림 6 의 PE 외부 래치 (L)도 L\_f와 L\_r가 있어 PE\_f의 sdata 는 L\_f으로, PE\_r 의 sdata 는 L\_r으로 전파한다. idata 는 i\_en, PE\_f와 L\_f 는 s0\_en, PE\_r과 L\_r은 s1\_en에 따라 래치된다.



(그림 7) PE 의 내부 구조 및 PE 열 결선

idata 는 클럭당 한 데이터씩 16 클럭동안 입력되어 각 PE 에 로딩되고, sdata 는 클럭당 두 데이터씩 입력되어 2 칸씩 이동된다. 이때 sdata 의 s0, s2, s4, ...는 클럭의 하 향에지에서 래치되고, s1, s3, s5, ...는 상향에지에서 래치 된다. idata 가 모두 loading 되고 sdata 가 PE00 까지 채워 지면 그때부터 클럭당 두 탐색블럭에 대한 절대차합 (SAD)이 구해지는데 홀수번째 PE 의 Lr 절대차값과 짝 수번째 PE 의 Lf 절대차값이 더해져서 첫째 탐색블럭에 대한 SAD 가 구해지고 홀수번째 PE 의 Lf 값과 짝수번 제 PE 의 Lr 값이 더해져서 둘째 탐색블럭에 대한 SAD 가 구해진다. 다음 클럭에서 셋째와 넷째 탐색블럭에 대한 SAD 가 구해진다. 이 후 중간에 loading 만 일어나 는 클럭구간이 존재하나 최종 sdata 가 입력되면서 최종 SAD 가 구해지고 움직임 추정 연산이 완료된다. 그림 8 은 이상의 동작에 대한 자세한 타이밍이다.



(그림 8) 제안구조의 동작 타이밍

#### IV. 구현 결과 및 검토

본 논문에서는 그림 6의 제안 구조를 ATM 통신용 MPEG-2 비디오인코더의 움직임추정기 개발에 적용, VHDL로 설계하여 기능 및 복잡도(게이트수)를 확인하였다. 움직임추정기의 규격은 다음과 같다:

- 탐색범위 : 수평방향: -16 ~ +15, 수직방향: -8 ~ +7
- 추정모드 : 프레임픽처부호화, 필드픽처부호화
- 예측모드 : 프레임간/필드간, 16x16MC/16x8MC
- 속도 : NTSC 비디오부호화(1,350x29.97 blocks/sec)
- 동작 주파수 : 27MHz(660cycles/block)
- 추정알고리즘 : 2 단계 추정
  - . 단계 1 : idata 및 sdata 수평 2:1 샘플 후 전탐색
  - . 단계 2 : 16x16 블록의 수평방향으로 -1 ~ +1 탐색
- 출력데이터 : mvdata, pdata, odata

여기서 본 제안구조는 단계 1 탐색(ME1), 단계 2 탐색(ME2), 움직임보상(MC)로 구성된 움직임추정기에서 ME1에 적용되었다.

Mentor 툴상에서 VHDL로 설계하여 C언어 프로그램 결과와 비교하여 기능을 확인하였으며 Synopsys 툴상에서 현대 0.35um 표준셀 라이브러리를 사용하여 게이트합성하였다. 표 1은 다수(16개)의 데이터를 공급해서 사이클수를 줄인 그림 3의 설계와 단일 데이터 공급 구조인 그림 4에 제안 방법을 적용한 설계에 대한 움직임추정기 전체(ME1+ME2+MC) 하드웨어 복잡도(SRAM과 로직)와 단계 1 탐색(ME1)만의 하드웨어 복잡도를 보여준다. 두 설계 모두 16(V)x8(H)개의 PE를 가지며 제안구조만 16(H)x15(V)개의 래치를 가지고 있다.

<표 1> 움직임추정 구조에 따른 복잡도 비교

ME 구조	그림 3 ME 구조		제안 ME 구조	
	SRAM	게이트	SRAM	게이트
ME1+ME2+MC	37K(18개)	100K	36K(6개)	94K
ME1 only	29K(14개)	76,884	27K(2개)	87,251

표 2는 두 설계의 단계 1 탐색에서의 소요 사이클수이다.

<표 2> 클럭 사이클수 비교

ME 구조	그림 3 ME 구조	제안 ME 구조
프레임픽처모드	384 클럭(24x16)	276 클럭(12x23)
필드픽처모드	384 클럭(24x16)	552 클럭(12x46)

표 1과 표 2는 주어진 클럭사이클수(660 사이클)를 충족하는 그림 3 구조와 필드픽처모드의 경우 1,104 사이클 소요로 사이클 요구조건을 만족하지 못한 그림 4 구

조에 제안방법을 적용하여 사이클조건을 만족시킨 구조와의 비교이다. 제안구조에서 필드픽처모드의 경우 8(H)x16(V)의 기준블록크기에 대한 수평 -8~+7, 수직 -8~+7의 전체탐색데이터수는 24(H)x46(V)이며 클럭당 2개씩 공급되므로 12x46=552 사이클이 된다.

그림 3 구조와 제안구조가 적용된 ME1 설계에서 제안구조의 게이트수가 좀 더 많으나 SRAM의 수에서는 많은 차이를 보인다. 그림 3의 구조는 클럭당 많은 데이터를 공급해 줘야 함에 따라 그만큼 SRAM을 병렬로 두고 있기 때문이다. 움직임추정기 전체 복잡도에서는 오히려 제안 구조가 낮는데 이는 ME2와 MC가 적극적으로 설계되었기 때문이며 그림 3 구조의 ME2와 MC도 동일하게 설계되면 ME1에서의 복잡도 차이가 그대로 유지될 것이다.

#### V. 결론

본 논문에서는 전탐색 블록매칭 움직임추정 VLSI 구조에서 PE의 두개 연산 처리(하나는 클럭의 상향에서 동작, 하나는 하향에서 동작, 하드웨어 복잡도는 기존의 1.5배), PE 열의 교번 결선, 클럭의 양(상향 및 하향)에서 동작시키는 방식으로 소요 사이클수를 반으로 줄여주는 구조를 제안했다.

본 제안 구조는 두배의 공급 데이터폭, 1.5배의 PE 복잡도, 2배의 SAD 계산회로 및 비교회로 복잡도로 전체 하드웨어가 그 만큼 증대되나, 기존 구조를 그대로 유지하면서 1.5배 정도의 하드웨어 복잡도로 두배의 클럭 사이클수 감소를 가져다 준다. 본 제안 구조는 실제 MPEG-2 비디오인코더의 움직임추정기에 적용하여 그 기능과 복잡도를 확인하였다.

#### 참고문헌

- [1] ISO/IEC, "Information Technology - Coding of Moving Pictures and Associated Audio," International Standard ISO/IEC 13818-2, Apr. 1996.
- [2] Bede Liu and Andre Zaccarin, "New Fast Algorithm for the Estimation of Block Motion Vectors", IEEE Trans. Circuits Syst. for Video Tech., vol.3, no.2, pp. 148-157, Apr. 1993.
- [3] Yeu-Shen Jehng, et al, "An Efficient and Simple VLSI Tree Architecture for Motion Estimation Algorithms", IEEE Trans. Signal Processing, vol. 41, no. 2, Feb. 1993.