

論文 97-34C-7-4

어레이 구조를 이용한 MPEG-2 비디오 인코더용 움직임 예측기 설계

(Design of A Motion Estimator for MPEG-2 Video Encoder Using Array Architecture)

沈載述*, 朴柱炫*, 周洛滋*, 金榮民*

(Jae Sul Shim, Ju Hyun Park, Rak Hyun Ju, and Young Min Kim)

요 약

본 논문에서는 MPEG-2 비디오의 부호화에 필수적인 움직임 예측기를 VHDL을 이용하여 설계하였다. 움직임 예측은 전체 영상 부호화 과정 중 50% 이상의 계산량을 필요로 하기 때문에 실시간 부호화의 걸림돌이 되고 있다. 따라서 본 논문에서는 실시간 부호화를 위해 어레이 구조를 이용한 고속 움직임 예측기를 설계하였다. 설계된 움직임 예측기는 예측 가능한 최대 프레임의 수가 초 당 37프레임으로 실시간 처리에 적합하다. 그리고 연산에 필요한 데이터를 메모리로부터 액세스하는 횟수가 기존의 구조보다 2배가 적으므로 상대적으로 느린 데이터 메모리를 사용할 수 있는 장점을 가지고 있으며, 입력된 데이터를 PE를 통해 전달시켜 사용함으로써 입력 핀의 수를 줄였다. 본 논문의 움직임 예측기는 MPEG-2에서 제공하는 여러 가지 움직임 예측 모드를 한 번의 실행으로 완벽하게 계산하며, 독립적인 제어 구조를 가지고 있으므로 단일 프로세서나 멀티미디어 칩의 서브 모듈로 사용할 수 있다.

Abstract

In this paper, we designed a motion estimator for MPEG-2 video coder using VHDL. Motion estimation is indispensable for encoding MPEG 2 video. Motion estimation takes over 50% computation power of video encoding. It prevents from real-time video processing. Designed processor is capable of encoding 37 frames per second and is suitable for real-time processing. The number of data accesses for computation is fewer than 2 times compared with that of old one. This makes slower memory module available. We minimize input pins to migrate input data through PEs. This processor can compute various motion estimation modes at one calculation that is supported by MPEG-2 video standard. Also independent control architecture makes this processor a single processor or a sub module in a multimedia chip.

I. 서 론

멀티미디어 관련 기술의 발전은 여러 분야에 걸쳐 변화를 유도하고 있다. 그 변화의 하나로 디지털 비디오 압축 방식인 MPEG 기술이 상용화됨에 따라 우리의 일상 생활이 윤택해지고 있다는 것을 꼽을 수 있

다. 특히 MPEG-1에 이어 MPEG-2는 디지털 TV 방송이나 HDTV, 대화형 TV, 디지털 비디오 디스크(DVD) 등 통신, 방송, 가전, 컴퓨터 분야에 광범위하게 응용될 수 있어 차세대 영상 산업으로 각광을 받고 있다.

그러나 MPEG-2는 CD-ROM에 기반을 둔 단방향성인 MPEG-1과는 달리 양방향 통신의 기능이 필수적이다. 그러므로 복호화 과정 뿐 만이 아니라 부호화 과정에서도 실시간성을 만족시켜야 한다.

영상 신호는 그 데이터량이 방대한 것이 특징이지

* 正會員, 全南大學校 電子工學科

(Dept. of Electronics, Chonnam National Univ.)

接受日字:1997年5月29日, 수정완료일:1997年7月2日

만, 연속되는 두 영상 간에 중복되는 데이터가 많다는 것 또한 특징이라고 할 수 있다. 따라서 데이터를 부호화하여 전송할 때 많은 데이터를 그냥 보내지 않고, 연속된 두 영상 사이에 존재하는 시간적인 중복성을 제거한 차 영상만을 부호화하여 전송함으로써 높은 데이터 압축율을 얻을 수 있다. 이러한 시간적인 중복성을 제거하는 방법으로 가장 널리 이용되는 기법이 움직임 예측이다. 이는 연속되는 두 개의 영상 중 직전 영상(Previous picture)을 기준 영상(Reference picture)으로 지정한 후, 이를 현재 처리해야 할 영상과 비교하여 두 영상간의 움직임을 예측하는 기법으로 현재 영상(Current picture)의 일정 영역을 직전 영상 내에서 이동시켜 가장 잘 정합하는 부분을 찾는 과정으로 이루어진다.

MPEG-1 표준 규격에서는 프레임 영상(Frame picture)만이 존재하였기 때문에 움직임 예측은 한 번으로 끝난다. 그러나 MPEG-2 표준 규격에서는 프레임 영상 뿐만 아니라 필드 영상(Field picture)도 존재하므로, 가장 정합이 잘 되는 부분을 찾기 위해서는 현재의 매크로 블럭(Current Macroblock)과 기준 영상을 프레임과 필드 영상 각각에 대하여 수행해야 한다. 이는 MPEG-1의 움직임 예측과는 비교가 되지 않을 정도의 계산 능력과 속도가 필요함을 의미한다. 실제 비디오 영상을 부호화할 때 움직임 예측은 전체 부호화 과정 중 50% 이상^[2]의 연산량을 필요로 하며, 이는 부호화 과정에서 실시간성을 만족시키기 위한 최대의 걸림돌이라고 할 수 있다.

따라서 효율적인 움직임 예측을 하기 위한 연구가 국내외로 활발히 진행되고 있는데, 주로 연산량 자체를 줄이는 방법과 연산의 규칙성을 이용하는 방법이 사용되고 있다. 연산량 자체를 줄이기 위한 방법으로 블럭 정합의 수를 줄이는 3-단 검색(3-Step Search) 등의 방법^{[1] [3] [4]}을 사용하거나, 비교되는 블럭의 크기를 줄이는 Domain Decimation 기법^{[5] [6]}이 이용되고 있다.

블럭 정합의 연산은 비교적 단순하고 규칙성이 있으므로 어레이 구조가 가장 많이 사용되고 있으며, 기존의 연구는 이와 같은 구조를 이용한 단순 프레임 예측^{[1] [3] [4] [8] [9] [10] [11] [13]}을 수행하는 연구가 주류를 이루었다. 그러나 MPEG-2 국제 표준^[7]으로 비월 영상(Interlaced Image) 처리가 승인되면서 비월 영상에 대한 효율적인 처리를 위한 예측기^{[3] [5] [}

14] [15]가 필요하게 되었다.

따라서 본 논문에서는 어레이 구조를 이용한 움직임 예측기를 설계하였으며, MPEG-2 비월 영상에 대한 처리가 가능하다. 또한 데이터의 병목 현상을 줄이기 위해 외부 핀 수를 줄였으며, 초당 예측 가능한 프레임 수가 실시간에 처리가 가능한 구조이다.

II. 움직임 예측기 구조

움직임 예측은 현재 영상과 기준이 되는 영상을 비교하여 가장 유사한 블럭을 찾아내는 과정을 말한다. 영상 간의 비교에는 각 영상 픽셀 값의 차이 값을 모두 합한 값 중에서 가장 작은 값을 찾는 방법을 사용한다. 움직임 예측의 기본 연산은 차이 값의 계산과 덧셈 계산이며, 이 연산을 대량으로 수행한다. 더구나 하나의 데이터가 여러 곳에서 사용되기 때문에 어레이 구조가 가장 적당한 구조라고 판단된다. 어레이의 각 PE(Processing Element)는 두 픽셀의 차이 값을 계산하고 계산된 값을 누적한다. 움직임 예측기의 블럭도는 그림 1과 같다. 본 움직임 예측기는 크게 입력부, 연산부, 비교 및 출력부 등 세 부분으로 구성되어 있다.

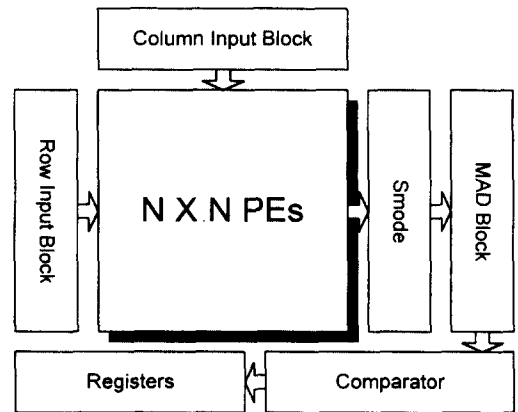


그림 1. 움직임 예측기 블럭도
Fig. 1. Block diagram of the motion estimator.

1. 예측 모드

움직임 예측은 예측의 방향에 따라 순방향 예측, 역방향 예측, 쌍방향 예측이 있는데, 이는 MPEG-1과 같다. MPEG-2에서는 비월 주사 영상이 새로 추가됨에 따라 이를 처리할 수 있도록 움직임 예측 모드가 다음 표 1과 같다.

표 1. 영상 구조에 따른 예측 모드
Table 1. Estimation modes according to the picture structures.

예측 모드	프레임 구조	필드 구조
필드 예측	MV 두 개	MV 한 개
16 x 8 필드 예측		MV 두 개
프레임 예측	MV 한 개	
듀얼 프레임 예측	MV 한 개와 DMV 한 개	MV 한 개와 DMV 한 개

그림 2와 그림 3은 프레임 구조와 필드 구조의 예측 모드를 그림으로 나타낸 것이다.

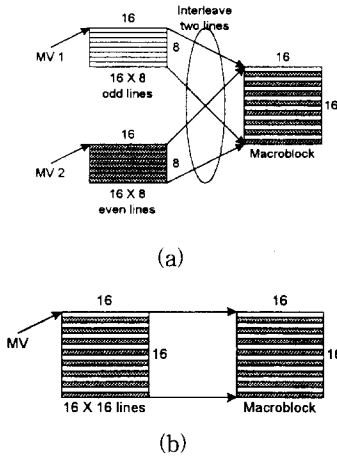


그림 2. 프레임 구조의 예측 모드 (a) 프레임 구조의 필드 예측 (b) 단순 프레임 예측
Fig. 2. Estimation modes for frame structure. (a) Field estimation of frame structure (b) Simple frame estimation

표 2는 예측 모드에 따라 예측 블럭(Predicted block)과 기준 블럭(Reference block)이 어떤 필드끼리 비교되는 지를 보이고 있다.

인코딩 과정에서는 영상의 구조에 따라 표 2.에서 나열한 예측 모드들을 모두 사용하여 정합이 가장 잘 되는 모드를 선택하기 때문에 MPEG-1에서의 프레임 예측에 비해 최대 4배의 연산량이 필요하다. 그런데 각 PE가 예측 블럭의 탐색 윈도우에 대한 하나의 편이점(Offset)의 MAD(Mean Absolute Difference)를 계산하므로, 표 2.에 나타난 모든 예측 모드의 값을 계산하지 않아도 된다. 즉, 프레임 예측은 프레임 구조의 필드 예측에서 탑 필드(top field)끼리의 값과 바텀 필드(bottom field)끼리의 값을 더하여 구할 수 있다. 필드 구조의 필드 예측 중 탑 필드의 값은 16 x

8 필드 예측의 위쪽 탑 필드(upper top field)와 아래쪽 탑 필드(lower top field)값을 더하여 구할 수 있고, 바텀 필드의 값은 역시 16 x 8 필드 예측의 위쪽 바텀 필드(upper bottom field)와 아래쪽 바텀 필드(lower bottom field)의 값을 더하여 구할 수 있다. 그러므로 PE에서 실제 계산량을 2배로 줄일 수 있다.

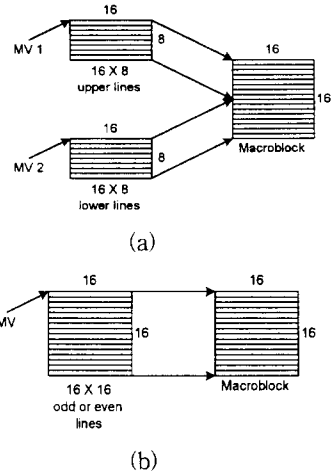


그림 3. 필드 구조의 예측 모드 (a) 16 x 8 필드 구조의 필드 예측 (b) 필드 구조의 필드 예측
Fig. 3. Estimation modes for field structure. (a) Field estimation of 16 x 8 field structure (b) Field estimation of field structure

표 2. 블럭 내 줄 간의 대응 관계
Table 2. Relationship between lines of blocks.

	대응 관계 (Predicted / Reference)	모드
프레임 구조	frame / frame ^x	1
	top field / top field*	2
	top field / bottom field*	3
	bottom field / top field*	4
	bottom field / bottom field*	5
필드 구조	field / top field [†]	1
	field / bottom field [†]	2
	upper field / upper top field [‡]	3
	upper field / upper bottom field [‡]	4
	lower field / lower top field [‡]	5
	lower field / lower bottom field [‡]	6

^x 프레임 예측 ^{*} 프레임 구조의 필드 예측
[†] 필드 구조의 필드 예측 [‡] 16 x 8 필드 예측

2. 연산부

연산부는 입력되는 픽셀의 차이 값을 계산하여 누적하는 모듈로 가장 기본적인 연산을 수행한다.

1) 차이 값 계산 모듈

그림 4는 연산부의 기본이 되는 차이 값 계산 모듈이다. 입력되는 픽셀 값 a, b는 부호 없는 8비트 정수로서 각각 예측 블록의 한 픽셀과 탐색 윈도우의 한 픽셀 값이다. 예측 과정에서 이 모듈로 입력되는 탐색 윈도우의 픽셀 값은 예측 블록에 대응되는 탐색 윈도우의 한 부분으로 예측 블록과 크기가 같은 부분의 픽셀 값이다. 어느 한 시간에서 입력되는 두 픽셀의 값은 비교되는 각 블록이 투영되었을 때 대응되는 픽셀 값이다. add 신호는 입력된 값이 누적될 것인지를 결정한다. 모든 픽셀의 값이 입력되면, 누적된 값은 out으로 출력된다. 하나의 계산 모듈은 예측 블록을 지정된 편이점만큼 탐색 윈도우 상에서 편이시켜 탐색 윈도우와 비교한 MAD 값을 계산한다. 4 x 4 개의 모듈을 사용하면 16개의 편이점에 대한 MAD 값을 얻을 수 있다.

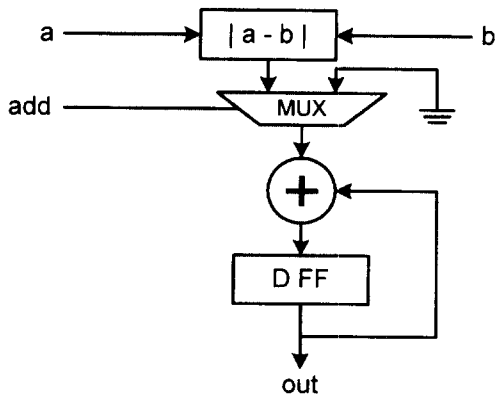


그림 4. 차이 값 계산 모듈
Fig. 4. Difference calculation module.

2) Processing Element (PE)

하나의 PE는 예측 블록을 탐색 윈도우의 한 편이점과 비교해서 MAD를 구한다. 다음 그림 5와 같이 차이 값 계산 블록을 두 개 사용한 PE를 구성한다.

C0는 현재 블록의 입력 값이고, R0는 탐색 블록의 입력 값이다. Cd와 Rd는 입력된 값을 다시 입력하지 않고 재사용하기 위해 각 PE를 거쳐 다시 입력되는 값이다. 두 개의 차이 값 계산 모듈은 처음 입력되는 값과 다시 입력되는 값에 대해 MAD를 계산한다. 다

중화기는 계산된 두 개의 MAD값 중 하나를 선택하여 오른쪽에 연결된 PE에 전달한다. MADi는 왼쪽에서 입력된 MAD 값이다. ctrl은 다중화기를 제어하기 위한 신호이다.

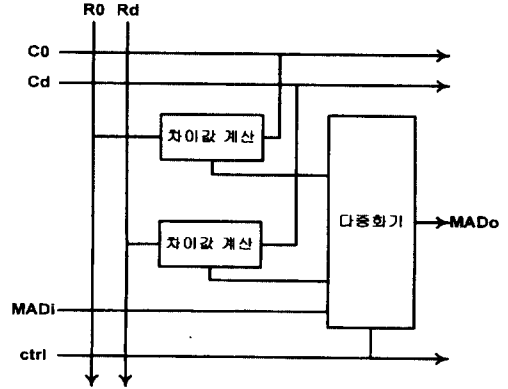


그림 5. Processing Element
Fig. 5. Processing Element.

3. 입력부

입력부에서는 N x N 셀에 입력 값을 전달한다. 예측 블록의 값은 행 입력 블록(Row Input Block)에 의해 입력되고, 탐색 윈도우 블록의 값은 열 입력 블록(Column Input Block)에 의해 입력된다.

1) 행 입력 블록

행 입력 블록은 예측 블록의 값을 입력한다. 프레임 영상 구조와 필드 영상 구조일 때 전혀 다른 회로를 사용하므로 이들을 분리하여 설명한다.

먼저 프레임 영상 구조의 행 입력 블록을 살펴본다. 그림 6은 프레임 영상 구조일 때 예측 모드를 계산하기 위한 데이터 입력도이다.

그림 6에서 정사각형은 PE를 나타내고, 직사각형은 각 블록의 한 행의 데이터를 나타낸다. 왼쪽의 데이터는 예측 블록의 데이터이고, 위쪽의 데이터는 탐색 블록의 데이터이다. 화살표는 입력되는 값이 재입력되는 모양을 나타낸다.

그림 6의 데이터 입력도를 가지고 표 2.에서 프레임 영상 구조일 때의 예측 모드 2, 3, 4, 5를 계산한다. PE 1에서는 예측 모드 2, 3의 MAD를 계산한다. 먼저 예측 블록의 탐색 필드 - 1행, 3행, ..., 2N-1행과 탐색 블록의 탐색 필드 - 1행, 3행, ..., 2N-1행을 서로 비교하여 MAD를 계산하는 데 이 값은 예측 모드 2의 값이다. PE 1에 탐색 블록의 바텀 필드 - 2행, 4

행, ... , 2N행이 입력될 때, 예측 블록의 탑 필드가 입력되게 하려면 PE 7에 입력된 예측 블록의 데이터 값을 PE 1의 입력으로 전달한다. PE 1에 탐색 블록의 바텀 필드가 입력될 때만 계산하면, 예측 모드 3의 MAD를 계산할 수 있다. PE 5, 6, 7은 수평 방향으로 편이된 예측 모드 2, 3의 MAD를 계산한다. PE 2에서는 예측 모드 4, 5의 MAD를 계산한다. PE 2에 예측 블록의 바텀 필드가 입력될 때 탐색 블록의 탑 필드 - 1행, 3행, ... , 2N-1행의 값이 입력되게 하려면 PE 4에 입력된 탐색 블록의 데이터 값을 PE 1을 통해 PE 2에 전달한다. 이 때 계산된 값은 예측 모드 4의 값이다. 예측 모드 5의 값은 각 블록의 바텀 필드가 입력될 때만 차이값을 계산하여 구한다. PE 3은 수직 방향으로 1 편이된 예측 모드 2, 3의 MAD를 계산하고, PE 4 역시 1 편이된 예측 모드 4, 5의 MAD를 계산한다.

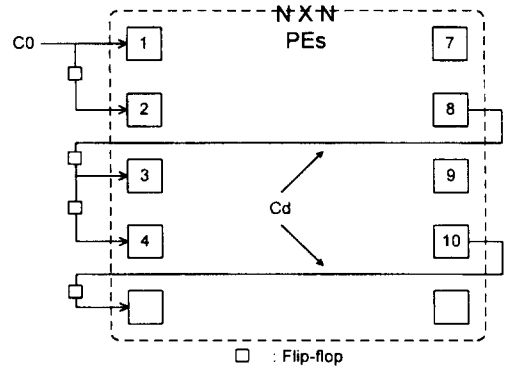


그림 7. 프레임 영상 구조의 행 입력부
Fig. 7. Row input block of the frame structure.

두 번째로 필드 영상 구조일 때 행 입력 블록도는 그림 8과 같다. 그림 8은 필드 영상 구조일 때 표 2의 예측 모드 3, 4, 5, 6을 계산하기 위한 데이터 입력도이다. 그림에서 H는 하위 블록을 가리킨다. 필드 영상 구조일 때는 프레임 영상 구조와 전혀 다른 방법으로 데이터가 입력된다. 즉, 예측 블록은 N열이 입력될 시간 동안 1행의 값이 입력되고, 2N-1행에 위치한 PE에는 상위 블록의 값이 2N행에 위치한 블록에는 하위 블록의 값이 입력된다. 그리고 탐색 블록은 두 행씩 번갈아 가며 상위 블록의 값과 하위 블록의 값이 입력된다.

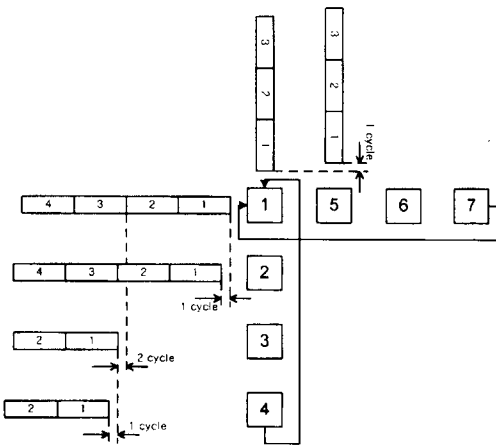


그림 6. 프레임 영상 구조의 데이터 입력도
Fig. 6. Data input diagram of the frame structure.

그림 6의 데이터 입력도를 만족하는 행 입력부는 그림 7과 같다. 그림 7에서 C0는 예측 블록의 입력 값이고, Cd는 PE 8에 입력되는 Cd를 출력시킨 값을 가리킨다. PE 2의 입력을 얻기 위해 PE 1의 입력을 플립플롭을 사용하여 한 클럭 지연시킨다. PE 8에 입력되는 Cd의 값은 PE 2에 처음 입력되는 값이 2행의 시간만큼 늦춰진 값이므로 여기에 플립플롭을 더 연결하여 PE 3에 필요한 입력 데이터를 만든다. 이와 같은 방법으로 나머지 PE에 대해 입력 값을 만든다.

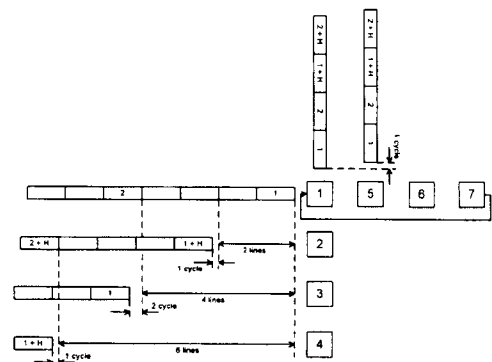


그림 8. 필드 영상 구조의 데이터 입력도
Fig. 8. Data input diagram of the field structure.

PE 1에서는 예측 모드 3과 4의 MAD 값을 계산한다. 예측 블록의 1행과 탐색 블록의 1행을 비교하여 예측 모드 3의 MAD값 일부를 계산하고, 예측 블록의

1행이 1행의 시간만큼 지연되어 입력될 때 탐색 블록의 2행이 입력되면 이를 계산하여 예측 모드 4의 MAD값 일부를 계산할 수 있다. PE 2에서는 예측 모드 5와 6의 MAD를 계산하는데, 예측 블록은 하위 블록의 값이 입력되고 탐색 블록 역시 하위 블록의 값이 입력된다. 각 하위의 1행을 계산하여 예측 모드 5의 MAD 일부의 값을 계산하고, 예측 블록의 값이 1행의 시간만큼 지연되어 입력될 때 2행이 입력되어 예측 모드 6의 MAD 일부의 값을 계산한다. 즉, 탐색 블록의 입력 값은 예측 모드 3, 4, 5, 6의 값을 모두 계산할 수 있도록 상위 2행과 하위 2행이 번갈아 입력된다.

그림 9는 그림 8의 데이터 입력도를 만족하는 행 입력부이다.

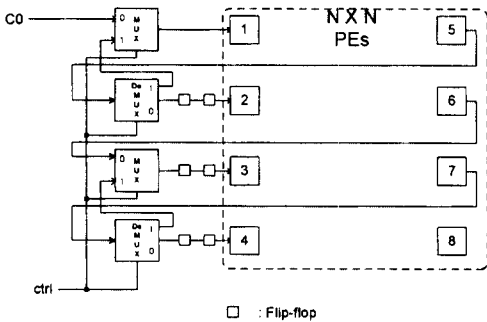


그림 9. 필드 영상 구조의 행 입력부
Fig. 9. Row input block of field structure.

C0에 입력되는 데이터는 그림 8의 예측 블록의 상위 블록과 하위 블록의 두 입력 값을 그림 10과 같이 중첩 시킨 것이다. 다중화기와 역다중화기는 제어신호에 의해 조정되어 입력된 데이터를 각 PE에 전달한다. PE 5의 출력 값이 1행일 때는 예측 모드 4의 계산을 위해서 PE 1에 전달되며, PE 5의 출력 값이 '1+H'행일 때는 예측 모드 5의 계산을 위해서 PE 2에 전달되어야 한다.

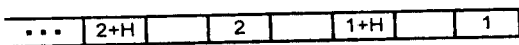


그림 10. 필드 영상 구조의 행 입력부의 데이터 형식
Fig. 10. Data format of row input block of field structure

행 입력 블록은 그림 7과 그림 9의 회로를 연결하여 구성되어 있으며 외부 신호인 fr/fib 에 의해 프레

임 구조를 쓸 것인가 지 필드 구조를 쓸 것인가를 결정한다.

2) 열 입력 블록

열 입력 블록에서는 탐색 윈도우의 데이터가 입력된다. 탐색 윈도우의 크기를 7 x 7로 놓고 각 픽셀의 값을 순차적으로 매겨 이를 탐색 윈도우의 데이터로 사용한다. 그림 11은 입력된 데이터와 그들이 전달되는 모습을 나타내고 있다. 한 행 단위로 데이터가 전달되며, 각 데이터 값은 두 클럭만큼 지연되는 것을 알 수 있다. 이 입력도를 만족시키기 위해서 열 입력 블록은 두 개의 입력 핀을 갖는다. 하나는 홀수 행과 다른 하나는 짝수 행의 데이터이다. 그림 12는 그림 13의 열 입력부에 입력되는 데이터와 이를 제어하는 다중화기의 신호이다.

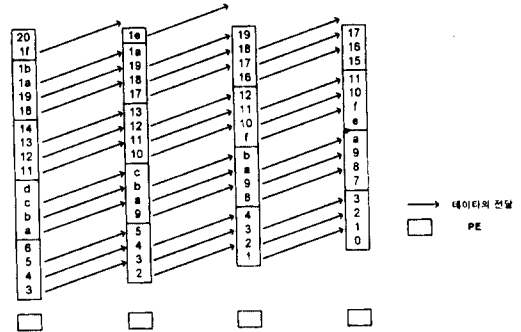


그림 11. 탐색 윈도우 블록의 데이터 입력도
Fig. 11. Data input diagram of search window block.

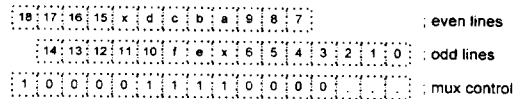


그림 12. 열 입력 블록의 입력 값과 제어 신호
Fig. 12. Input data and its control signal of column input block.

필드 영상 구조일 때는 상위 블록의 두 행이 입력된 후에 하위 블록의 두 행이 입력되고, 다시 상위 블록의 값이 입력되는 과정을 반복하며, 프레임 영상 구조일 때와 같은 모양으로 입력된다.

4. Smode

NxN PEs 에서는 각 영상 구조에서 다음과 같은 예측 모드의 값을 계산한다.

- 프레임 영상 구조일 때 계산하는 예측 모드

- 2n-1행에 위치한 PE : 예측 모드 2, 3
 - 2n행에 위치한 PE : 예측 모드 4, 5
 - 필드 영상 구조일 때 계산하는 예측 모드
 - 2n-1행에 위치한 PE : 예측 모드 3, 4
 - 2n행에 위치한 PE : 예측 모드 5, 6
- 계산되지 않은 나머지 예측 모드는 Smode에서 계산된다.

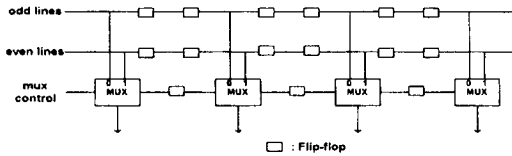


그림 13. 열 입력 블럭부
Fig. 13. Diagram of the column input block.

1) Smode에서 덧셈의 규칙

프레임 영상 구조에서 예측 모드 1은 예측 모드 2와 5를 더해 계산한다. 예측 모드 2는 블럭의 홀수 행을 비교한 것이고, 예측 모드 5는 블럭의 짝수 행을 비교한 것이므로 이 둘을 더하면 예측 모드 1을 계산할 수 있다. 그림 6의 PE 1과 PE 2의 출력을 더하면 y방향 편이가 0인 예측 모드 1의 값이 된다. 그런데 y방향 편이가 1인 예측 모드 1의 값을 계산하고자 할 때 PE 2의 출력은 예측 모드 1의 관점에서 볼 때 홀수 행의 비교가 되고, PE 3의 출력은 짝수 행이 된다. 다시 y방향 편이가 2인 예측 모드 1의 관점에서 보면 PE 3의 출력은 홀수 행의 비교가 되며, PE 4의 출력은 짝수 행의 비교가 된다. 그러므로 전체를 통합하면 각 PE의 출력을 연이어 더한 것이 된다. 이를 그림으로 나타낸 것이 그림 14이다.

필드 영상 구조일 때 예측 모드 1과 2의 계산은 다음과 같다. 예측 모드 1은 탐색 윈도우 블럭의 홀수 필드, 탑 필드를 비교하는 것이기 때문에 홀수 필드의 상위 8행의 비교인 예측 모드 3과, 홀수 필드의 하위 8행의 비교인 예측 모드 5를 더하여 구한다. 예측 모드 2는 탐색 윈도우 블럭의 짝수 필드, 바텀 필드를 비교하는 것이기 때문에 짝수 필드의 상위 8행의 비교인 예측 모드 5와, 짝수 필드의 하위 8행의 비교인 예측 모드 6을 더하여 구한다.

5. MAD 블럭

각 PE에서는 한 편이점에 대한 각 예측 모드의 MAD 값이 계산된다. 계산된 MAD 값은 PE의

MAD 다중화기를 통해 오른쪽에 인접한 PE로 전달된다. 움직임 예측에서 목표는 가장 작은 MAD값과 이것의 편이 좌표를 구하는데 있다. 그러므로 PE에서 출력된 MAD 값을 한 경로를 통해 전달해야 한다. 영상 구조에 따른 예측 모드마다 하나의 경로를 둔다. 각 경로마다 하나의 예측 모드의 MAD가 전달된다. 각 PE가 계산하는 예측 모드의 값이 다르므로 같은 예측 모드를 계산하는 PE의 출력을 묶어서 하나의 경로로 구성한다. 경로를 구성할 때 사용하는 회로는 그림 15와 같다.

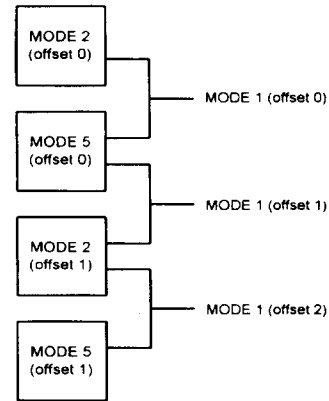


그림 14. 프레임 영상 구조에서 예측 모드 1의 계산
Fig. 14. Calculation of estimation mode 1 in the frame structure.

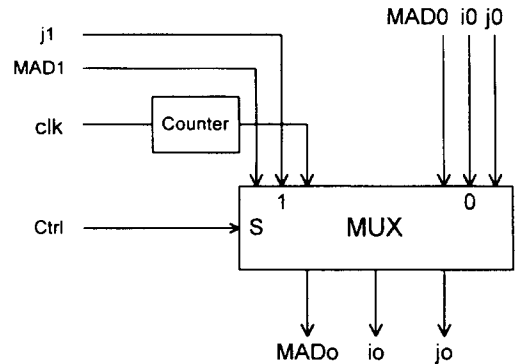


그림 15. 경로 구성 회로
Fig. 15. Path configuration circuit.

계산된 MAD는 PE를 통해 오른쪽으로 이동하고, 경로를 구성하기 위해 아래쪽으로 MAD가 전달된다. 그림 15에서 MAD1은 PE를 통해 오른쪽으로 전달된 MAD이고, MAD0는 경로 구성 회로에 의해 위에서 전달된 MAD이다. i와 j는 각각 x방향 편이와 y방

향 편이를 나타낸다. y방향 편이는 경로 구성 회로의 y방향 위치에 따라 회로적으로 정의된다. 그러나 x방향 편이는 PE의 x위치와 관계하므로 계수기를 두어 x방향의 편이를 계산한다. MAD, i, j는 한 단위로 움직인다.

경로를 구성할 때 중복되는 경로가 존재하므로 이를 중복하여 사용한다. 표 3.은 경로 구성 회로를 이용하여 MAD 블럭을 구성한 것이다. 표에서 프레임 영상 구조의 예측 모드를 프레임 모드로 필드 영상 구조의 예측 모드를 필드 모드로 나타냈다.

표 3. MAD 블럭의 구성
Table 3. Configuration of MAD block.

모드	배치 위치	y방향 편이
프레임 모드2 필드 모드 3	홀수 행	0, 2, 4,...
프레임 모드3 필드 모드 4	홀수 행	1, 3, 5,...
프레임 모드5 필드 모드 5	짝수 행	프레임 예측 : 1, 3, ... 필드 예측 : 0, 2, ...
프레임 모드4 필드 모드 6	짝수 행	프레임 예측 : 0, 2, ... 필드 예측 : 1, 3, ...
프레임 모드 1	모든 행	0, 1, 2, ...
필드 모드 1	짝수 행	0, 2, ...
필드 모드 2	짝수 행	1, 3, ...

6. 비교 및 출력부

비교부에서는 MAD 블럭에서 출력된 MAD 중 가장 작은 MAD를 찾는다. 이것은 비교기를 이용하여 구성하였으며 MAD를 비교할 때 작은 값을 저장하고 이를 입력된 MAD와 비교한다. 최종적으로 구해진 MAD와 편이점은 각 예측 모드에서 예측 블럭이 탐색 윈도우에서 가장 정합이 잘 되는 점의 좌표와 정합의 차이가 된다.

Ⅲ. 성능 비교 및 분석

설계된 움직임 예측 프로세서는 프레임 예측, 필드 예측, 16 x 8 필드 예측 모드를 계산할 수 있으며, 예측 블럭의 크기가 NxN 일 때 탐색 윈도우의 크기와 탐색 범위는 (2N-1)x(2N-1), ±N/2이다.

본 논문에서 제안한 구조의 움직임 예측기는 예측

블럭의 크기가 16x16 이고, 탐색 윈도우의 크기는 31x31으로 탐색 범위는 ±7이다. 영상의 크기는 720x480으로 매크로 블럭의 개수가 1,350개이다. 표 4.는 본 예측기의 성능을 영상의 구조가 프레임 구조 일 때와 필드 구조일 때로 나누어 보여주고 있다. 블럭당 클럭 수는 예측 블럭과 탐색 윈도우의 데이터가 예측기에 입력되어 각 PE에서 처리되어 비교 및 저장할 때까지 걸리는 시간을 나타낸다. 움직임 예측기는 50MHz로 동작하므로 프레임당 클럭수와 프레임의 매크로 블럭의 개수로 초당 계산가능한 최대 프레임을 계산할 수 있다.

표 4. 제안된 구조의 성능
Table 4. Performance proposed.

영상 구조	#클럭 수		최대 프레임
	/블럭	/프레임	
프레임 구조	564	761,400	65
필드 구조	997	1,345,950	37

성능 측정 결과 연산량이 많은 필드 구조의 영상에서 초당 37프레임을 예측할 수 있으므로 7프레임의 여유를 가지고 있다.

표 5는 기존에 제안된 움직임 예측 구조를 가지고 필드 구조의 영상을 예측하고자 할 때 필요한 매크로 블럭 당 데이터 액세스 횟수이다. 기존의 구조에서는 필드 구조의 영상을 예측하기 위해 데이터를 여러 번 액세스해야 하지만, 제안한 구조에서는 입력된 데이터를 재사용하기 때문에 데이터의 액세스 횟수가 기존의 구조보다 2배가 적다.

표 5. 블럭 당 데이터 액세스 수
Table 5. Data accesses per block.

영상 구조	블럭 당 데이터 액세스	초당 전체 데이터 액세스
Chen et al ^[10]	3,651	147,865,500
Yeo and Hu ^[11]	2,304	93,312,000
Proposed	1,217	49,288,500

Ⅳ. 결론

본 논문에서는 동영상 압축 표준인 MPEG 2비디오를 지원하는 움직임 예측 프로세서를 VHDL을 이용하여 설계하였다. MPEG 2는 방송 등의 분야에서 응용할

수 있도록 비월 주사방식을 사용하는 TV의 영상을 그대로 사용할 수 있기 때문에, 이로 인해 프레임 예측 뿐만이 아니라 필드 예측 방법이 있다. 각 예측 방법에서 다시 5개와 6개의 서로 다른 예측 모드가 있으므로, 기존의 단순한 프레임간의 움직임 예측만이 아닌 복합적인 예측 방법이 필요하다. 또한 방송, 통신 등에 쓰일 수 있도록 실시간으로 부호화를 수행해야 하기 때문에 초 당 예측 가능한 프레임의 수가 최소 30 프레임 이상이어야 한다.

이와 같은 요구를 만족하기 위해서 본 논문에서는 대량의 반복적인 데이터 처리에 적합한 어레이 구조를 제안하였다. 여러 가지 예측 모드를 한 번에 계산하기 위해서 연산 기본 블럭을 두고, 이를 정해진 규칙으로 배열하였다. 예측 모드의 일부분은 다른 예측 모드의 조합으로 이루어지는 성질을 이용하여 이를 통합하는 구조를 사용함으로써 계산이 집중되는 현상을 없앴다.

설계된 프로세서는 입력 핀의 수가 예측 블럭의 데이터와 탐색 윈도우 블럭의 데이터를 위한 16개뿐이며, 초 당 액세스 횟수가 기존의 구조보다 2배가 적기 때문에, 데이터의 입출력으로 인한 데이터 버스의 병목 현상을 없앴다. 720 x 480 의 기본 화면을 움직임 예측했을 때 초 당 가능한 최대 프레임의 수가 37프레임으로 실시간 부호화가 가능하다. 설계된 프로세서의 구조는 독립적인 제어 구조를 사용하고 있어 단일 프로세서로 설계되거나 멀티미디어 프로세서의 서브 모듈로 사용될 수 있다.

앞으로는 매크로블럭의 움직임 벡터에 대한 시간적이고 공간적인 통계에 바탕을 두어 탐색 윈도우를 설정함으로써 동일한 탐색 윈도우의 크기를 가지고도 효율적인 움직임 벡터 생성을 할 수 있도록 하며, 프로세서의 면적을 줄이기 위해 연산 블럭의 크기를 감소시키고, 프로세서의 연산 능력을 향상시켜 HDTV 등에 쓰일 수 있도록 하는 연구가 필요하다.

참 고 문 헌

- [1] P. Lakamsani et al., "An enhanced three step search motion estimation method and its VLSI architecture", Proc. Of Circuits and Systes, 1996.
- [2] T. Akiyama et al., "MPEG2 Video Codec Using Image Compression DSP", IEEE Trans. on Consumer Electronics, vol. 40, no. 3, August 1994.
- [3] K. Suguri et al., "A real-time motion estimation and compensation LSI with wide-search range for MPEG 2 video encoding", IEEE International Solid-state Circuits Conf., 1996.
- [4] E.S. Kim et al., "A new 3-step search block matching algorithm for video coding", Proc. of ITC-CSCC, July, 1996.
- [5] G. Fujita et al., "Single chip MPEG 2 MP@ML motion estimator", Proc. of ITC-CSCC, July, 1996.
- [6] K.W. Cheng and S.C. Chan, "Fast block matching algorithm for motion estimation", Proc. Of Circuits and Systems, 1996.
- [7] ISO/IEC JTC 1/SC 29/WG 11, "Generic coding of moving pictures and associated audio", ISO/IEC 13818-2 IS, 1994.
- [8] C.H. Hsieh and T.D. Lin, "VLSI architecture for block matcing motion estimation algorithm", IEEE Trans. Circuits Syst. Video Technology, vol. 2, June 1992, pp. 169-175.
- [9] SGS-Thomson, "Motion estimatino Processor", STi3220, Data Sheet, Jan 1994.
- [10] C.L. Wang, K.M. Chen, J.M. Hsiung, "A high-throughput, flexible VLSI architecture for motion estimation", in Confer. Proc. ICASSP-95 vol. 5, 1995.
- [11] H. Yeo, Y.H. Hu, "A novel modular systolic array architecture for full-search block matching motion estimation", in Confer. Proc. ICASSP-95 vol. 5, 1995.
- [12] S.Y. Kung, "VLSI Array processors", Prentice Hall, Englewood Cliffs, New Jersey 07632, 1988, pp. 8-9.
- [13] S. Dutta, W. Wolf, "A flexible parallel architecture adapted to block-matching motion-estimation algorithm", IEEE Trans. Circuits Syst. Video Technology, vol. 6, 1996.
- [14] A. Pirson et al., "A programmable motion estimation processor for full search block matching", in Proc. of Visual Comm. and Image Proc. '95, vol 2., 1995.

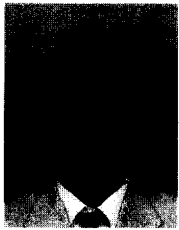
[15] 고재형 외 4, "MPEG-2 Codec을 위한 움직임 추정기의 구현", 전자공학회 추계학술대회 논문

집 제 18권 제 2호, 1995년

저 자 소 개

沈 載 述(正會員)

1973년 12월 19일생. 1995년 2월 전남대학교 전자공학과 졸업. 1997년 2월 전남대학교 대학원 전자공학과 졸업(공학석사). 1997년 3월 ~ 현재 전남대학교 대학원 전자공학과 박사과정 재학중. 주관심분야는 영상압축 프로세서설계, 프로세서 개발 환경 설계 등임.



周 洛 洵(正會員)

1970년 10월 27일생. 1993년 2월 전남대학교 전남대학교 전자공학과 졸업. 1995년 2월 전남대학교 대학원 전자공학과 졸업(공학석사). 1995년 3월 ~ 현재 전남대학교 대학원 전자공학과 박사과정 재학중. 주관심분야

는 영상압축 프로세서 설계 등임.

朴 柱 炫(正會員)

1969년 7월 13일생. 1993년 2월 전남대학교 전자공학과 졸업. 1995년 2월 전남대학교 대학원 전자공학과 졸업(공학석사). 1995년 3월 ~ 현재 전남대학교 대학원 전자공학과 박사과정 재학중. 주관심분야는 저속 영상 코딩, DSP 프로세서 설계 등임.

金 榮 民(正會員) 第33卷 B編 第1號 參照

현재 전남대학교 전자공학과 교수