Video SoC를 위한 고성능 ME/MC IP의 설계

서영호* · 최현준** · 김동욱**

Design of High-Performance ME/MC IP for Video SoC

Young-Ho Seo* · Hyun-Jun Choi* · Dong-Wook Kim**

논문은 2008년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

요 약

본 논문은 비디오 압축을 고성능으로 수행하기 위한 움직임 예측(motion estimation, ME) 및 보상(compensation, MC) 알고리즘의 VLSI 구조를 제안하고 하드웨어로 구현하였다. 움직임 예측을 계산하기 위해서는 일반적으로 SAD 결과를 이용하게 되는데 이를 위하여 새로운 연산방법을 제안하였다. 제안한 SAD 연산 방법으로 인해 연산의 효율성이 증대되고 메모리의 사용을 줄임으로써 ME/MC의 성능을 높였다. 제안한 ME/MC 하드웨어는 TSMC 90nm HVT CMOS 공정으로 구현하였다. 구현된 하드웨어는 약 33만 게이트를 점유하였고, 143MHz의 클록 주파수에서 안정적으로 동작하였다.

ABSTRACT

This paper proposed a new VLSI architecture of motion estimation (ME) and compensation (MC) for efficient video compression and implemented it to hardware. ME is generally calculated using SAD result. So we proposed a new arithmetic method for SAD. The proposed SAD calculation method increases arithmetic efficiency and decreases external memory usage. Finally it increases performance of ME/MC. The proposed ME/MC hardware was implemented to ASIC with TSMC 90nm HVT CMOS library. The implemented hardware occupies about 330K gates and stably operates the clock frequency of 143MHz.

키워드

motion estimation, compensation, video compression, VLSI, hardware design

Ⅰ. 서론

현재 멀티미디어 기술은 정보산업 뿐만 아니라 경제, 문화 등 사회 각처에서 활용되고 있을 정도로 우리 생활의 중요한 요소로 자리 잡고 있다. 멀티미디어의 핵심은 영상과 음향이다. 특히 영상 데이터는 그 양이 방대하여 이들을 원 데이터로 그대로 저장 및 전송하는 것은 불가능하다. 이러한 문제점을 극복하기 위해서 영상/비디오 압축에 대한 연구 및 기술개발은 계속 진행되어 왔다. 아울러 유/무선 네트워크 대역폭도 방대해져서 이를 통한 멀티미디어 데이터의 처리를 위한 연구는 더욱 활발해지고 있다. 또한 영상/비디오의 압축 및 이들의 전송방식

* 광운대학교 교양학부
** 광운대학교 전자재료공학과

접수일자 2008. 05. 08

에 관한 국제적인 표준화 작업도 계속 진행되어 왔다. 대표적인 예로 ITU-T의 JPEG, MPEG 및 H.26X 계열과 3GPP, 3GPP2등을 들 수 있다.

H.261/3/4 혹은 MPEG-1/2/4와 같은 동영상 압축표준에서 ME는 시간적인 중복성을 제거하는데 중요한 역할을 한다. 이와 관련해서 H.261 및 MPEG-1/2와는 달리 H.263/4 혹은 MPEG-4에서는 다양한 크기의 블록별 움직임 벡터(움직임 벡터)를 지원함으로써 압축 시 압축률을 더욱 증대시킬 수 있다. 그러나 이를 H/W로 구현하면 큰 버퍼가 소요되고 연산량이 많다. 또한 제어가 복잡하므로 항상 면적, 타이밍, 제어, 그리고 전력 등의 관점에서 많은 고려가 필요하다.

동영상 압축과정에 필수적인 움직임 예측 과정은 연산량이 방대하고 메모리 호출의 횟수도 많다. 예를 들어, ITU H.261 권고의 움직임추정에 필요한 계산량은 약 1.2 GOPS (giga operation per second)이다. 또한 소비되는 전력은 전체 동영상 압축과정의 약 50%를 차지하는 것으로 보고되고 있다[6]. 따라서 동영상 실시간 처리가 필요한 휴대 정보단말기 구현을 위하여 저전력 움직임추정 VLSI 설계가 필수적이다.

움직임 예측 VLSI에서 가장 많이 사용되는 움직임 예측 방식은 하드웨어 구현이 간단한 전역탐색 블록정합(full search block 정합) 방식이다[1][2]. 움직임벡터 예측을 위해서 이전 혹은 이후 프레임의 탐색영역내의 블럭들을 탐색하여 현재 프레임의 참조블럭과 가장 잘 유사도가 높은 블럭의 위치를 탐색한다. 이와 같은 방식은 블럭 내의 모든 화소들을 이용하기 때문에 복원된 프레임의 화질이 우수하지만 연산량이 너무 많다는 단점이 있다. 단점을 해결하기 위하여 고속 블럭정합 방식들이 제안되었다[3][4][5]. 고속 블록정합 방식은 예측 시 모든 픽셀을 이용하지 않기 때문에 전역탐색 방식에 비해서 예측 초차가 크고, 복원된 영상의 화질이 떨어진다는 단점이 있다. 따라서 전역탐색 블럭정합 방식의 고성능이면서 전력을 작게 소비하는 방식 및 구조에 대한 연구가 필수적이다[7]. 전역탐색 블럭정합 방식의 많은 연산량으로 인하여 소비전력이 증가되는데 이를 감소시키기 위하여 외부 메모리 호출횟수를 감소시키는 방법[8]이 제안되었고, 움직임 예측 시 불필요한 연산을 제거시키는 방법[9]과 이를 1D 하드웨어 배열로 설계한 연구가 있었다[10]. 또한 픽셀을 표현하는 비트수를 줄여 움직임추정 하드웨어의 복잡도를 감소시키고 처리속도를 증가시키는 구조가 제안되었으나 복원된 프레임의 화질이 저하된다는 단점이 있었다[11].

본 논문에서는 효율적인 ME/MC 프로세서의 구조를 제안하고 이를 하드웨어로 구현하고자 한다. 본 논문은 다음과 같이 구성된다. 2장에서는 제안한 알고리즘에 대해서 설명하고 3장에서는 제안한 VLSI 구조에 대해서 나타낸다. 4장에서는 구현결과를 보이고 5장에서 결론을 맺는다.

Ⅱ. 제안한 알고리즘

본 장에서는 하드웨어의 구조를 제안하기에 앞서서 효율적인 하드웨어의 구조를 설계하기 위해 몇 가지 알고리즘을 제안하고자 한다.

2.1. 프레임 경계 확장

프레임 밖에서 안으로 이동하는 객체가 있다고 가정하자. 움직임 예측시 프레임 경계 내부에서 최적의 매크로 블록 또는 블록을 찾는다면 움직임 벡터 외에 빗금친 부분에 대한 residual 정보를 추가로 전송해야 한다. 만일 프레임을 확장했고 탐색창도 확장되었다고 가정하자. 경계 외부에서 최적의 정합을 찾을 수 있고 이를 이용할 수 있다면 residual에 대한 정보는 필요 없다. 이런 방식으로 움직임 벡터가 프레임 경계 외부까지 확장 했을 때 이를 UMV(Unrestricted 움직임 벡터) 라고 한다. 그림 1(b)에서 이를 나타내었다.

대부분의 프레임은 객체를 중앙에 두고, 배경이 대부분이다. 연속적인 프레임에서 배경에 해당하는 경계 부근의 화소를 확장해서 이전 프레임과 비교하면 이들은 매우 유사하다. 따라서 UMV를 이용하면 압축률을 증가시킬 수 있다. UMV를 위해서 탐색구간을 프레임 외부로 확장하는데 있어서 확장은 새로운 화소를 만드는 것이 아니라, 경계 내의 화소를 그대로 확장하는 개념이므로 UMV로 인한 화질 개선 및 움직임 예측의 성능개선은 없다. 확장 시 경계에 있는 화소를 아래와 같은 3가지 절차에 의해서 경계 외부로 확장 한다. 그림 2 에서 이를 나타내었다.

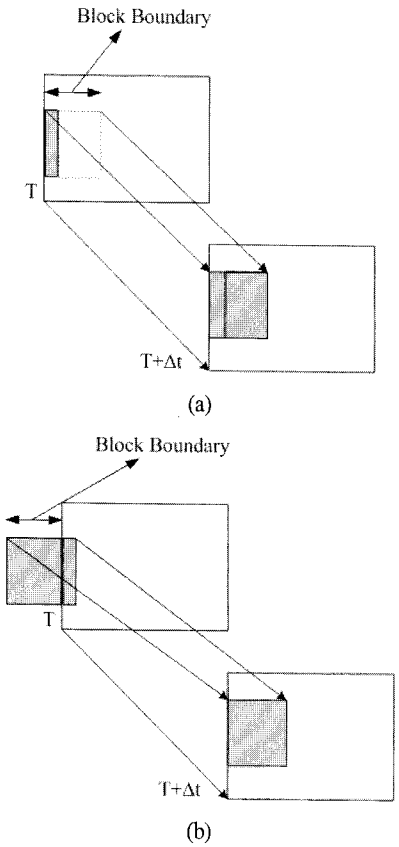

그림 1. 경계확장 (a) 제한된 움직임 벡터
(b) 제한되지 않은 움직임 벡터
Fig. 1. Boundary padding (a) restricted motion vector
(b) unrestricted motion vector

① Horizontal 확장
② Vertical 확장
③ Edge Corner 확장

그림 2에 의하면 보간된 화소가 프레임 외부로 확장되므로 보간 후에 확장을 해야 한다. 또한 mpeg 표준 문서에서는 확장된 화소들은 보간시 사용되지 않는다고 규정하고 있다. 하드웨어 설계 시 확장을 먼저 진행하는 것이 훨씬 수월하다.

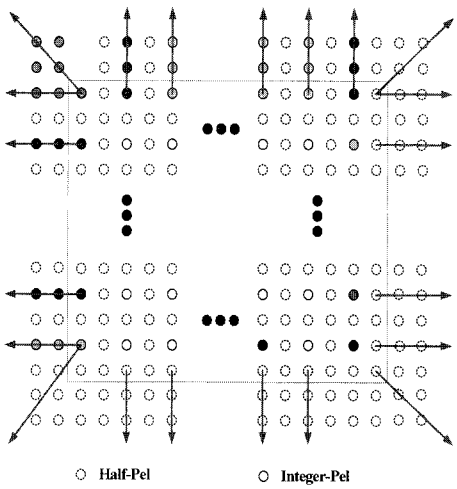

그림 2. 프레임 경계의 확장
Fig. 2. Boundary padding

2.2. 블록 정합 알고리즘

2.2.1 BMA

매크로블록 또는 블록 간의 유사도 비교는 대부분 블록 정합 알고리즘(BMA)를 이용한다.

BMA를 그림 3에 나타내었다. 탐색구간 내의 각각의 참조 화소를 기준점으로 현재 M×N 블록을 정합 시킨다. 그 다음 현재 pel들과 참조 pel들과의 유사도를 비교한다. Half-pel을 위한 BMA는 현재 매크로 블록 또는 블록의 M×N을 참조 프레임에 M×N으로 정합시키지 않는다. 참조 프레임에는 2M×2N으로 정합시켜야 한다. 즉 2pel 간격을 두고 현재 매크로 블록 또는 블록의 pel을 정합시켜야 한다. 유사도의 측정을 위해서는 식 (1)의 연산량이 작은 SAD가 사용된다. 그림 4에서 BMA시 SAD 추출 방법을 나타내었다.

$$SAD = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} |c_{ij} - d_{ij}| \quad (1)$$

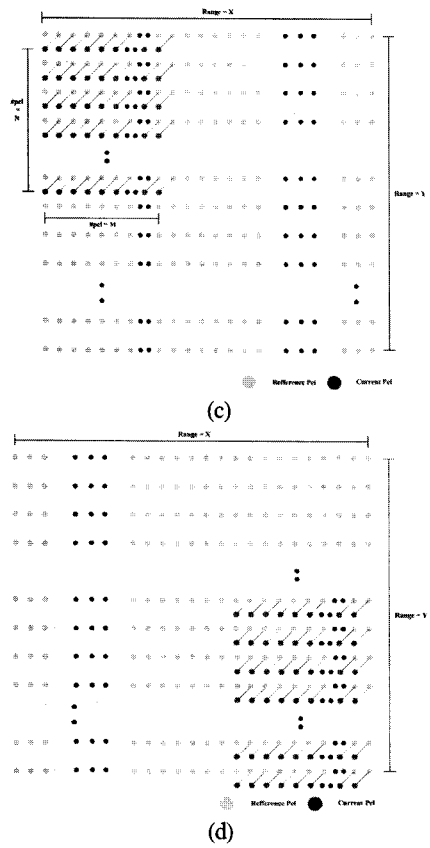

그림 3. 블록 정합 알고리즘 (a) (M×N) 번째 BMA (b) (M×N) 번째 SAD

Fig. 3. Block matching algorithm (a) (M×N)'st BMA (b) (M×N)'st SAD

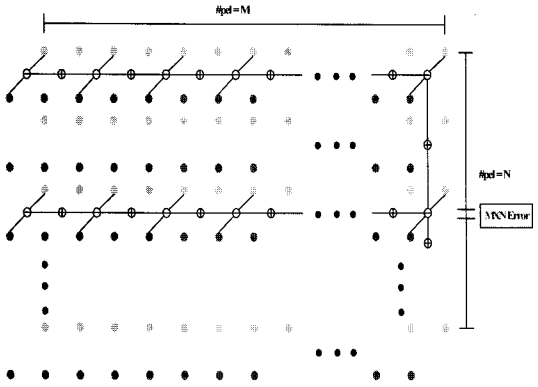

그림 4. BMA를 이용한 SAD 추출(Mp=Np=2)

Fig 4. SAD calculation using BMA(Mp=Np=2)

2.2.1 BMA 효율성 개선

Motion 예측의 연산시간에 절대적으로 영향을 주는 요소는 현재의 M×N 블록을 참조의 그것과 유사도를 비교하는 BMA이다. 만약 모든 M×N 블록의 모든 pel을 비교하지 않고 일정 간격 Mp×Np을 두어 SAD를 추출한다면 BMA를 위한 연산시간은 1/(Mp×Np)으로 감소할 것이다. 이에 대한 Min SAD 값, 추출된 움직임 벡터 값의 일부분을 표 1에 비교하였다.

표 1은 움직임이 가장 많은 어느 매크로블록의 integer pel에 대한 예측 결과이다. 표 1에서 Mp=Mn=1일 경우에 MinSAD값이 제일 정확함을 알 수 있다. SAD 값은 매크로 블록의 모드 결정 및 1MV 혹은 4MV의 결정요소이다. 따라서 가능한한 정확해야 하므로 BMA에서 화소의 정합 간격을 조절하는 것은 바람직하지 않다.

표 1. Mp, Mn에 따른 예측 결과

Table 1. Estimation result according to Mp, Mn

Mp	Nn	Min SAD	mv_x	mv_y	연산시간
1	1	5866	19	1	100
1	2	2724	19	0	50
2	1	2875	19	0	50
2	2	1377	19	0	25
4	4	199	19	0	6.25
8	8	9	-5	-2	1.56

BMA의 목적은 탐색구간내에서 최소의 SAD를 찾는 것이다. 따라서 BMA 시 기존의 최소 SAD를 초과한다면 더 이상 남아있는 다른 화소에 대한 BMA를 수행할 필요가 없다. 이를 그림 5에 나타내었다. 동영상의 성격에 따라 다르겠지만 이런 방법을 사용한다면 BMA 연산시간을 상당부분 감소시킬 수 있다.

그림 5에서 이전 M×N 블록의 BMA결과 최소 SAD가 1000이었다고 가정하자. 다음 M×N 블록의 BMA 수행 시 3번째 화소 비교 시 SAD가 기존의 최소 SAD보다 크다.

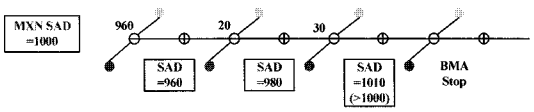

그림 5. BMA 종료 방법

Fig. 5. BMA termination method

식 1에 의해서 M×N 블록의 BMA는 점점 커지게 되어 있다. 따라서 이 M×N 블록은 기존의 최소 SAD보다 작은 SAD를 추출할 가능성이 없으므로 더 이상 BMA를 진행하는 것은 의미 없다. 예상대로 표 2에서 MinSAD보다 작을 때만 BMA를 수행한 결과 2배 이상의 연산시간을 감소시켰다.

표 2. BMA 종료 적용시 연산시간 비교
Table 2. Comparison of operational time in case of BMA termination

#프레임	#프레임	1	2	4	5	7	8
Cpu Time(s)	Full	5.80	5.53	5.67	5.69	5.74	5.63
	Partial	2.81	3.00	3.04	2.94	2.92	2.78

2.2.3 탐색 범위 및 탐색 순서

M×N 블록의 BMA의 연산시간이 고정되었다고 가정한다면 M×N 블록의 BMA를 수행해야 하는 횟수를 결정하는 탐색구간과 움직임 예측 블록의 연산시간은 비례한다. 탐색구간이 클수록 예측의 정확도는 증가하지만, 이를 위한 버퍼 크기 및 연산시간도 증가하게 된다. 일반적으로 탐색구간=32이다. 이 연구에서는 전역탐색 BMA만 고려한다.

전역탐색 BMA인 이상 모든 영역에 대해서 계산을 해야한다는 것은 바꿀 수 없다. 그러나 앞에서 언급한 BMA 종료 기법을 사용할 경우 그림 6(b)의 스캔 방법이 유리하다. 대단히 빠른 영상이 아니라면 움직이는 객체는 근처에 있게 마련이다. 객체가 근처로 이동한다면 근처에서 Min SAD값을 발견할 가능성이 크다. 반대로 멀어지면 유사도가 작아져서 SAD값이 커지므로 BMA 의 연산시간이 더욱 감소할 확률이 크다.

현재의 SAD값이 Min SAD와 같을 때에는 움직임 벡터가 작은 위치의 SAD값을 취해야 한다. 그림 6(b)의 예를 든다면, 10번에서 Min SAD가 발견되었고, 11의 SAD값이 이와 같다고 가정하자. 그렇다면 추출될 움직임 벡터를 고려해야 한다. mv_x(10)+mvy(10) > mv_x(11) +mvy(11) 이므로 11의 SAD와 움직임 벡터를 선택해야 한다. 반면 그림 6(a)에서는 이런 기법을 적용시키기가 까다롭다. 이러한 점들을 고려할 때 그림 6(b)의 방법이 장점이 많다. 그러나 바로 하드웨어 구현 시 제약이 많다는 단점을 갖는다. 따라서 하드웨어로 구현하고자 할 경우에는 그림 6(b)의 스캔 방법을 선택해야 한다.

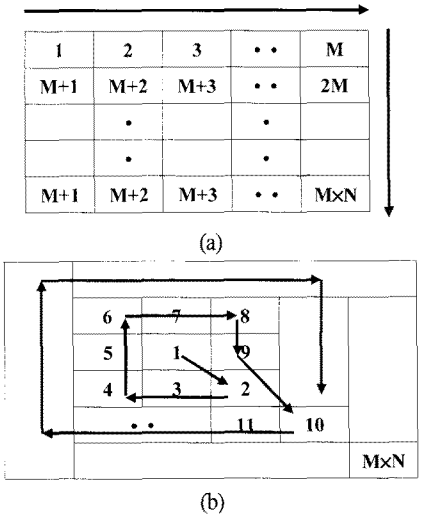

그림 6. 스캔 방식 (a) 래스터 (b) 회전 순서
Fig. 6. Scan method (a) raster (b) spiral order

2.3. Half-pel 보간

움직임 예측시 실제 화소와 가장 근사한 화소를 찾고 이들을 보간한다. 그러면 그림 7(a)는 (b)와 같이 가상의 pel이 만들어 진다. 이들에 대해서 한번 더 움직임 예측을 수행한다. Half-Pel로 인한 움직임 벡터는 float형이 된다. MPEG-4 표준에서는 확장된 화소는 보간에 사용되지 않는다고 했다. 그러나 사용해도 무관하다. 그림 8(a)는 확장후 보간을 하지 않은 그림이고, 그림 8(b)는 확장후에 보간을 실시하였다. 수식적인 연산과 실험결과에 따르면 이러한 방법이 원래의 방법과 동일한 결과를 가져온다는 것을 확인할 수 있다.

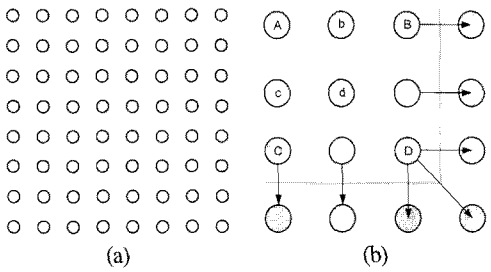

그림 7. 보간 방식 (a) integer-pel (b) half-pel
Fig. 7. Interpolation method (a) integer-pel (b) half-pel interpolation

2.4. SAD 연산 방법의 제안

표 3에는 제안한 SAD 계산 방법을 자세히 정리하였다. SAD 계산을 표 2와 같은 방법으로 수행하면 M×N 블록의 모든 SAD값을 256 cycle의 대기지연시간 이후에 연속적으로 추출할 수 있다.

표에서 curr_pel은 현재 화소의 위치이고 ref_pel은 참조 프레임의 화소 위치이다. SAD at MV는 해당 위치에서의 SAD 연산에 해당한다. 표에 나타난 방법을 따르면 기계적으로 SAD를 연속적으로 추출할 수 있다. 최종 추출된 SAD 결과는 M×N SAD에 해당한다.

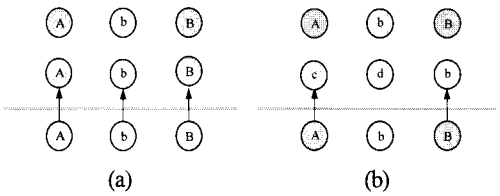

그림 8. 확장 방법 (a) 단순, (b) 보간 확장
Fig. 8. Padding method (a) simple Padding, (b) interpolation after Padding

III. 제안한 하드웨어 구조

본 장에서는 앞서서 제안된 알고리즘들을 바탕으로 하드웨어의 구조를 제안하고자 한다.

3.1. 전체 ME 블록의 구조

표 4에서 ME/MC 프로세서의 입출력 정보를 나타내었다. 출력으로 매크로 블록단위와 4MV 모드를 위한 블록 단위의 움직임 벡터와 SAD값이 출력된다. 출력된 SAD값은 1MV, 4MV의 기준이 되며 통상 1MV SAD < $\sum$ 4MV SAD 일 때 4MV를 선택한다. 이 외에도 SAD값은 매크로 블록의 모드 결정 시 사용된다.

그림 9에서 매크로 블록 단위의 ME/MC 프로세서를 나타내었다. 그림 9에서 참조 프레임과 탐색구간이 2개씩 사용된다. 이는 마이크로소프트사의 참조 소프트웨어에서 integer pel과 half 혹은 quarter pel의 참조를 각각 달리 사용하기 때문이다.

표 4. ME/MC 프로세서의 입출력 사양
Table 4. Input/output specification of ME/MC processor

Layer	I/O	Port Name	Description
V.O.P	In	vop_height, vop_width	VOP size
		vop_rounding_type	Interpolation시 사용될 rounding value
매크로블록		mb_x, mb_y	매크로 블록의 2D 시작 좌표
		int16_srchx, int16_srchy	16X16 mode의 integer pel 탐색구간
		int8_srchx, int8_srchy	8X8 mode의 integer pel 탐색구간
		flt16_srchx, flt16_srchy	16X16 mode의 half pel 탐색구간
		flt8_srchx, flt8_srchy	8X8 mode의 half pel 탐색구간
		intsrchwin[x16][y16]	입력 영상의 탐색구간 내의 pel
		fltsrchwin[x8][y8]	Reconstructed된 탐색구간 내의 pel
		current_mb[16][16]	현재 매크로블록
	Out	SAD16	16X16 mode의 SAD 값
		mv16_x, mv16_y	16X16 mode의 Sign을 포함한 움직임 벡터
		SAD8[4]	8X8 mode의 각 블록의 SAD 값
		mv8_x[4], mv8_y[4]	8X8 mode의 sign을 포함한 각 블록의 움직임 벡터

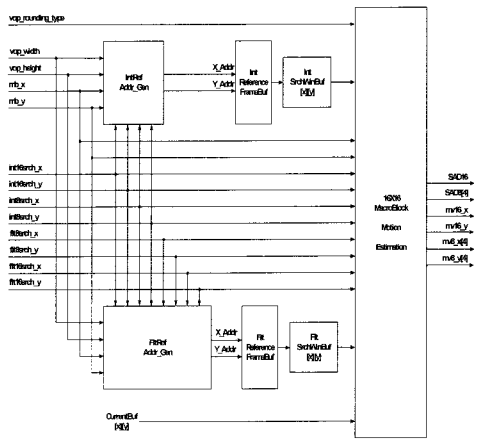

그림 9. ME/MC 프로세서의 구조
Fig. 9. Architecture of ME/MC processor

3.2. 경계확장 블록의 구조

그림 10에서 확장을 위한 블록의 구조를 나타내었다. 표준에서는 매크로 블록 단위인 16×16 의 배수로 확장

표 3. SDA 계산 방법
Table 3. SAD calculation method

phase	curr_pel	ref_pel	SAD at MV	curr_pel	ref_pel	SAD at MV	.....	curr_pel	ref_pel	SAD at MV	curr_pel	ref_pel	SAD at MV	MIN SAD
0	[0,0]	(0,0)	mv(0,0)+(0,0)											
1	[0,0]	(0,1)	mv(0,1)+(0,0)	[0,1]	(0,1)	mv(0,0)+(0,1)		don't care						
•			•											
•			•											
16	[0,0]	(1,0)	mv(1,0)+(0,0)	[0,1]	(0,16)	mv(0,15)+(0,1)	.....	[0,14]	(0,16)	mv(0,2)+(0,14)	[0,15]	(0,16)	mv(0,1)+(0,15)	
			•			•	•		•			•		don't care
			•			•	•		•			•		
	[15,0]	(15,0)	mv(0,0)+(15,0)	[15,1]	(15,1)	mv(0,0)+(15,1)	.....	[15,14]	(15,14)	mv(0,0)+(15,14)	[15,15]	(15,15)	mv(0,0)+(15,15)	
•			•			•	•		•			•		
•			•			•	•		•			•		
255	[0,0]	(15,15)	mv(15,15)+(0,0)	[0,1]	(15,15)	mv(15,14)+(0,1)	.....	[0,14]	(15,15)	mv(15,1)+(0,14)	[0,15]	(15,15)	mv(15,0)+(0,15)	
			•			•	•		•			•		don't care
			•			•	•		•			•		
	[15,0]	(15,15)	mv(0,15)+(15,0)	[15,1]	(15,15)	mv(0,14)+(15,1)	.....	[15,14]	(15,15)	mv(0,1)+(15,14)	[15,15]	(15,15)	mv(0,0)+(15,15)	SAD(0,0)
256	[0,0]	(0,16)	mv(0,16)+(0,0)	[0,1]	(15,16)	mv(15,15)+(0,1)	.....	[0,14]	(15,16)	mv(15,2)+(0,14)	[0,15]	(15,16)	mv(15,1)+(0,15)	
			•			•	•		•			•		don't care
			•			•	•		•			•		
	[15,0]	(16,0)	mv(1,0)+(15,0)	[15,1]	(15,16)	mv(0,15)+(15,1)	.....	[15,14]	(15,16)	mv(0,2)+(15,14)	[15,15]	(15,16)	mv(0,1)+(15,15)	SAD(0,1)
257	[0,0]	(0,17)	mv(0,17)+(0,0)	[0,1]	(0,17)	mv(0,16)+(0,1)	.....	[0,14]	(15,17)	mv(15,3)+(0,14)	[0,15]	(15,17)	mv(15,2)+(0,15)	
			•			•	•		•			•		don't care
			•			•	•		•			•		
	[15,0]	(16,1)	mv(1,1)+(15,0)	[15,1]	(16,1)	mv(1,0)+(15,1)	.....	[15,14]	(15,17)	mv(0,3)+(15,14)	[15,15]	(15,17)	mv(0,2)+(15,15)	SAD(0,2)
•			•			•	•		•			•		
•			•			•	•		•			•		

하도록 되어있다. 그러나 이는 동작을 구현하는데 있어서 아무 의미가 없다. 움직임 벡터가 지시하는 화소의 주소에서만 확장하면 되기 때문에 필요 없이 16×16의 배수로 확장할 필요가 없다. 확장은 하드웨어 구현 시 매우 까다롭다. 그러나 UMV를 위해서는 프레임 경계에서의 완벽한 확장이 선행되어야 한다. 확장처리를 잘 못하고 UMV에 대한 복원을 시도한다면 버퍼 주소 접근 시 오버플로우가 발생한다.

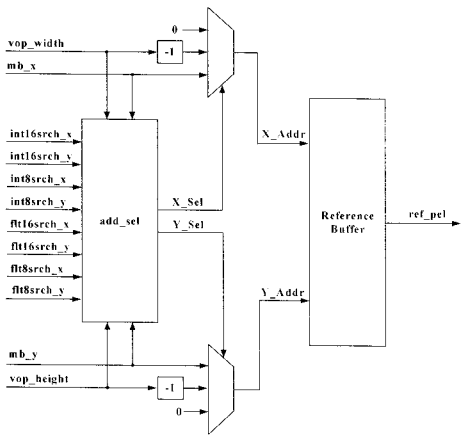

그림 10. 경계확장 블록의 구조
Fig. 10. Architecture of 경계 확장 블록

3.3. ME 블록의 구조

그림 11은 3-step ME 블록의 하드웨어 구조이다. 그림 12에서 그림 11의 움직임 벡터와 SAD값을 추출하는 부분의 하드웨어 구조를 나타내었다. 그림 11에서는 MV Gen 블록의 입력이 incremental X,Y 밖에 없다. 이는 그림을 간소화 하기 위함이다. 그러나 실제로 탐색창을 저장하는 버퍼의 처음 즉, 0번지부터 탐색구간 내의 화소들이 저장되지 않는다. Half-pel 등을 위해서 상하좌우에 1개 이상의 여유 pel이 저장되야 하기 때문이다. 마이크로소프트의 참조 소프트웨어에서는 16×16 integer MV 추출 후 이를 근거로 8×8 integer MV를 추출한다. 따라서 3-step 방식의 BMA를 채택하였다. 이런 이유에서라도 MV Gen 블록의 입력은 탐색구간 등이 추가 되어야 한다. 그림 12에서 SAD를 추출하는 블록에 대한 상세한 블록 다이어그램을 그림 13에 나타냈다. BMA를 실시하려면 탐색구간 버퍼 내에 참조 pel이 저장되어 있어야 한다. 그림 14에는 full search BMA의 타이밍 스케줄링 다이어그램을 나타내었다.

Ⅳ. 실험결과

제안된 영상 압축/복원 프로세서는 VHDL(VHSIC 하

드웨어 Description Language)을 이용하여 하드웨어로 설계하였다. 또한 설계된 하드웨어는 시스템내의 호스트 프로세서와 함께 실시간으로 수행하도록 하였다. 설계는 VHDL 하향식(top-down) 설계 기법을 통해서 이루어졌으며 특정 구현 대상에 국한하지 않는 범용적인 설계를 이루고자 오직 IEEE 표준 라이브러리만을 사용하였다. 각각의 모듈들은 RTL(registar transfer level)수준으로 설계되었고, 구조적 수준(structure-level)에서 서로 연결되었다. 설계된 하드웨어를 검증하기 위해 Synopsys의 Design ComplierTM로 논리 합성을 수행하였다.

구현된 회로는 TSMC의 90nm HVT CMOS 라이브러리를 이용하여 Synopsys에서 합성되었는데 NAND를 한 개의 게이트로 환산할 경우 구현된 하드웨어는 약 330K 게이트의 자원을 사용한다. Synopsys에서 합성된 회로의 타이밍 시뮬레이션은 NC Sim에서 이루어졌고, 타이밍 시뮬레이션 결과 약 143MHz의 클록 속도에서 안정적으로 동작함으로 확인하였다. 본 논문에서 설계된 회로는 컴퓨터와의 인터페이스 혹은 A/D 변환기와의 동작적인 연결성을 위해서 33Mhz 혹은 27Mhz의 동작조건을 만족하면 되기 때문에 타이밍 시뮬레이션을 통해 검증된 결과는 만족할 만하였다. 표 5에 하드웨어 자원에 대해서 자세히 정리하였다.

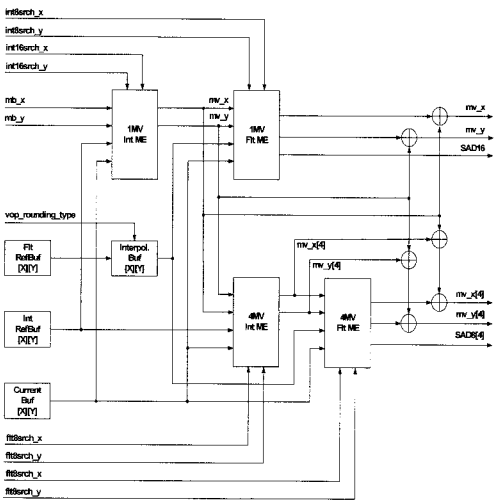

그림 11. ME 블록의 구조
Fig. 11. Architecture of ME 블록

합성된 회로에 입출력 포트를 위한 패드를 붙인 후 Apollo를 이용하여 P&R을 수행하였고 DRC 및 LVS 과정을 통해서 에러없이 결과를 추출하였다. 전체 chip 크기는 3860x3860mm이고 설계된 core의 utilization은 약 70%에 해당한다. 그림 15에 Apollo를 이용하여 P&R을 수행한 결과를 나타내었다.

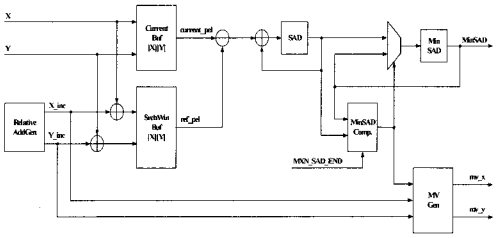

그림 12. MV, SAD 블록의 구조
Fig. 12. Architecture of MV, SAD block

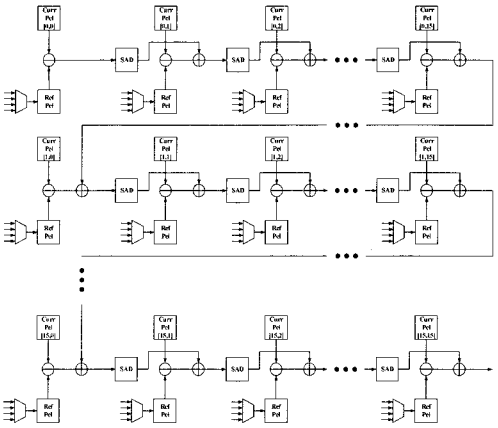

그림 13. SAD 추출 블록의 구조
Fig. 13. Architecture of SAD extraction block

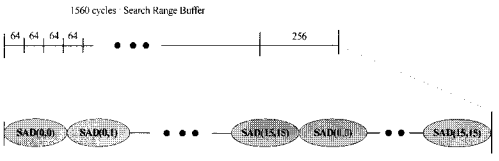

그림 14. BMA의 타이밍 스케쥴링 방법
Fig. 14. Timing scheduling method of BMA

표 5. 게이트 수
Table 5. Gate count

	탐색구간	
	-16~+15	-32~+31
Search 버퍼	78,166	171,121
Current F/F	15,155	15,155
Adder	11,520	46,080
Substractor	7,680	33,420
etc.	30,000	30,000
# of Gate	182,521	334,655

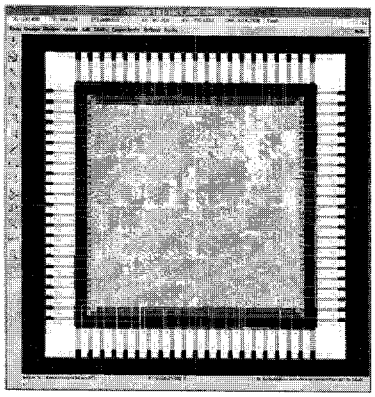

그림 15. Apollo를 이용한 P&R 결과
Fig. 15. P&R result using Apollo

V. 결론

본 논문은 비디오 SoC를 위한 움직임 예측 및 보상알고리즘의 VLSI 구조를 제안하고 하드웨어로 구현하였다. 움직임 예측을 계산하기 위해서는 일반적으로 SAD 결과를 이용하게 되는데 이를 위하여 새로운 연산방법을 제안하였다. 또한 경계 확장 방식, 반화소 확장, 및 보간 방식등을 제안하여 연산의 효율성을 높이고 메모리의 사용을 줄여서 ME/MC의 성능을 높였다. 제안한 ME/MC 하드웨어는 TSMC 90nm HVT CMOS 공정으로 구현하였다. 구현된 하드웨어는 약 33만 게이트를 점유하였고, 143MHz의 클록 주파수에서 안정적으로 동작하는 것을 확인할 수 있었다. 구현된 하드웨어는 다양한 비디오 하드웨어 분야에서 IP 솔루션으로 활용가능할 것으로 사료된다.

참고문헌

[1] S. Y. Yap and J.V. McCanny, "A VLSI Architecture for Variable block Size Video Motion estimation", IEEE Trans. on Circuits and Systems II: Express Briefs, Vol. 51, issue 7, pp. 384-389, Jul. 2004.

[2] Chen-Fu Lin, Jin-Jang Leou, "Adaptive Fast Full Search Motion estimation Algorithm for H.264," IEEE International Symposium on Circuits and Systems, pp.1493-1496, May 2005.

[3] S. Zhu and K. Ma, "A new diamond search algorithm for fast block matching", IEEE Trans. on Image Proc., Vol. 9, No. 2, pp.287-290, Feb. 2000.

[4] A. Tourapis, O. Au, and M. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation", IEEE Trans. on Circuits and Systems for Video Technology, pp. 934-947, Oct. 2000.

[5] Donghyung Kim, Jechang Jeong, Xiao Song and Chengke Wu, "A Fast Macroblock Mode Selection Algorithm In The H.264/AVC Standard," IWAIT 2005, pp.157-162, Jan. 2005

[6] K. Guttag et al. "a single-Chip Multiprocessor For Multimedia: The MVP," IEEE Computer Graphics and Applications, Nov, 1992. pp 53-64.

[7] Elgamel, M.A.; Shams, A.M.; Bayoumi, M.A.; "A comparative analysis for low power motion estimation VLSI architectures," 2000 IEEE Workshop on SiPS, 2000 Page(s): 149-158.

[8] Bo-Sung Kim; Jun-Dong Cho, "VLSI architecture for low power motion estimation using high data access reuse," ASICs, 1999. AP-ASIC '99. The First IEEE Asia Pacific Conference on 1999, Page(s): 162-165.

[9] V.Do, and K. Yun, "A Low-Power Architecture for Full-Search block-matching Motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, pp. 393-398, 1998.

[10] L. Sousa, and N. Roma, "Low-Power Array Architect ures for motion estimation," Proceeding of the IEEE i nternational Workshop on Multimedia Signal Processi ng, Copenhagen, MMSP'99, pp. 679-684, Copenhage n, Denmark, September 1999. Co-Chaired by Ed Depr

ettere and Bastiaan Kleijm.

[11] Z. He, and M. Liou, "Reduing hardware Complexity of Motion estimation Algorithms Using Truncated Pixels," IEEE International Symposium on Circuits and Systems 1997, ISCAS'97, pp. 2809-2817, Hong Kong, June 1997.

저자소개

서영호(Young-Ho Seo)

1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
2001년 2월 : 광운대학교 일반대학원 졸업(공학석사)
2000년 3월~2001년 12월 : 인티스닷컴(주) 연구원
2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)
2003년 6월~2004년 6월 : 한국전기연구원 연구원
2004년 12월~2005년 8월 : 유한대학 연구교수
2005년 9월~2008년 2월 : 한성대학교 전임강사
2008년 3월~현재 : 광운대학교 교양학부 조교수
※관심분야 : 2D/3D 영상 및 비디오 처리, 디지털 홀로그램, SoC 설계, 워터마킹/암호화

최현준(Hyun-Jun Choi)

2003년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
2005년 2월 : 광운대학교 일반대학원 졸업(공학석사)
2005년 3월~현재 : 광운대학교 일반대학원 박사과정
※관심분야 : 영상압축, 워터마킹, 암호학, FPGA/ASIC 설계, Design Methodology

김동욱(Dong-Wook Kim)

1983년 2월 : 한양대학교 전자공학과 졸업(공학사)
1985년 2월 : 한양대학교 대학원 졸업(공학석사)
1991년 9월 : Georgia공과대학 전기공학과 졸업(공학박사)
1992년 3월~현재 : 광운대학교 전자재료공학과정교수. 광운대학교 신기술 연구소 연구원
2000년 3월~2001년 12월 : 인티스닷컴(주) 연구원
※관심분야 : 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication