

논문 2011-48SD-2-5

병렬처리 기반의 H.264/AVC 인코더를 위한 저 메모리 대역폭 움직임 예측 코어설계

(A Low Memory Bandwidth Motion Estimation Core for H.264/AVC
Encoder Based on Parallel Current MB Processing)

김 시 혜*, 최 준 림**

(Shi-hye Kim and Jun-rim Choi)

요 약

본 논문에서는 H.264/AVC 인코더를 위한 하드웨어 지향 알고리즘의 정화소 및 부화소 움직임 예측 코어를 제안한다. 정화소 움직임 엔진의 경우 참조블록은 병렬 처리 내의 연속된 현재 블록들에 공유되어 데이터 재사용율을 높이고 오프칩 대역폭을 줄인다. 부화소 움직임 엔진의 경우 두 단계의 순차적 보간 신호 생성 대신 불필요한 후보 위치들 대신 1/2과 1/4 화소정밀도 신호를 병렬 기법으로 생성하여 처리량을 두배로 높인다. 또한 제안하는 H.264 움직임 예측 코어는 Chartered 0.18 μ m CMOS 1P5M 공정의 MPW(Multi-Project Wafer)를 통해 칩으로 제작되었으며 높은 처리량으로 HDTV 720p 30fps를 실시간 지원한다.

Abstract

In this paper, we present integer and fractional motion estimation IP for H.264/AVC encoder by hardware-oriented algorithm. In integer motion engine, the reference block is used to share for consecutive current macro blocks in parallel processing which exploits data reusability and reduces off-chip bandwidth. In fractional motion engine, instead of two-step sequential refinement, half and quarter pel are processed in parallel manner in order to discard unnecessary candidate positions and double throughput. The H.264/AVC motion estimation chip is fabricated on a MPW(Multi-Project Wafer) chip using the chartered 0.18 μ m standard CMOS 1P5M technology and achieves high throughput supporting HDTV 720p 30 fps.

Keywords : H.264/AVC Encoder, Integer Motion Estimation, Parallel processing

I. 서 론

H.264/AVC는 ITU-T 와 ISO/IEC가 공동으로 개발한 디지털 비디오 코딩의 표준으로 비트율 최적화를 통해 기존의 동영상 압축 표준인 MPEG-4, H.263, MPEG-2 에 비해 높은 압축 성능을 지닌다^[1]. 이러한 뛰어난 성능은 새로운 화면간 예측 기법에 기인한다.

화면간 예측 성능을 향상시키기 위해 1/4 화소 정밀도 움직임 예측과 가변블록 크기의 움직임 보상, 복수 참조픽처로부터 예측이 채택되었다. H.264의 움직임 예측은 두 부분으로 구성되어있다. 첫 번째로 정화소 움직임 벡터를 찾기 위한 정화소 움직임 예측과 이를 토대로 1/4 화소 정밀도 신호를 찾기 위한 부화소 움직임 예측이 그것이다. 부화소 움직임 예측은 첫 번째 단계로 41개 블록들의 정화소 위치로부터 주변에 1/2화소정밀도 예측 신호를 생성하고 그 다음 과정으로 최적의 1/2화소정밀도 신호 주변에 1/4 화소정밀도 예측 신호를 생성한다. 이로 인해 정화소와 부화소 움직임 예측은 PSNR +4dB 증가로 비트율-왜곡 최적화에 기여한다^[2]. 그러나 움직임 예측 엔진은 특히 HDTV 부호화를 위해

* 학생회원, ** 정회원-교신저자, 경북대학교 전자전기 컴퓨터학부

(Kyungpook National University, School of Electrical Engineering and Computer Science)

※ 본 연구는 IT-SoC 핵심인력양성사업과 IDEC의 지원으로 이루어졌으므로 이에 감사드립니다.

접수일자: 2010년11월15일, 수정완료일: 2011년1월25일

엄청난 계산의 복잡도를 요구한다. 따라서 HDTV의 실시간 부호화를 위해 움직임 예측의 속도 향상은 필수적이다.

기존의 연구에서 병렬 움직임 예측구조, 고속 움직임 예측 알고리즘, 데이터 재사용을 위한 다양한 방식 등의 기술을 제안하고 있다. 예를들어 참조문헌 [3]은 연속된 8개의 참조 후보 계산을 위한 8개의 처리 요소들이 사용된다. [4~5]에 정의된 데이터 재사용 방식은 현재 탐색영역과 인접한 탐색 영역 사이에 이루어진다(level C). 부화소 움직임 예측을 위해 [3]의 저자는 정화소 후보 주변의 9개의 부화소 위치 탐색을 위해 9개의 병렬 처리 장치들을 제안하였다. 또한 [6]에서 흥미로운 것은 한 블록과 이것의 상위 계층 블록 사이에 같은 움직임 벡터가 발견된다는 것이다. 이는 둘 사이에 중복된 영역이 1/2과 1/4 단계에서 각각 재사용 될 수 있음을 의미한다. 계산의 복잡도 감소를 위해 [6]은 4x4, 4x8, 8x4와 같은 작은 블록 처리 모드를 제거하였다. 이는 실시간 HDTV 1080p에 적합한 높은 처리량 증가를 불러왔으나 PSNR과 비트율이 크게 감소된다는 단점이 있다. 이웃한 부화소 위치 후보들 간에 높은 상관관계를 이용하여 [7]에서는 부화소 위치 후보의 숫자를 참조 모델 소프트웨어인 JM^[8] 과 같은 17개 후보 대신 8개 또는 9개로 줄였다. 이를 통해 [7]은 계산 복잡도를 참조 소프트웨어에 비해 약 50% 줄일 수 있었다. 그러나 모든 데이터와 보간 절차는 변경되지 않는다. 이것은 여전히 불필요한 부화소 위치에 대한 여분값을 지니고 있음을 의미한다.

Level C의 데이터 재사용 가능성을 활용하고 off-chip 대역폭을 줄이기 위해 본 논문에서는 연속된 현재 매크로 블록들에 공유되는 참조 블록에 대한 병렬 프로세싱을 제안한다. 부화소 움직임 예측은 종래의 접근법인 두 단계 알고리즘을 대신하여 병렬기법을 통해 21개의 부화소 위치 중 최상의 후보를 찾는다. 제안하는 부화소 움직임 예측 알고리즘은 처리시간을 반으로 줄이고 처리량을 배가시킨다. 또한 하드웨어 구현이 아주 간단하다.

본 논문의 나머지 구성은 다음과 같다. II장에서 현재 매크로블록에 공유되는 참조블록의 데이터 재사용 기법 알고리즘과 이의 병렬 구조를 설명하고, III장에서는 병렬 부화소 엔진을 제안한다. IV장에서는 합성 결과 및 기존 디자인과의 비교 분석을 제시한다. 마지막으로 V장에서는 본 논문의 결론을 맺는다.

II. 정화소 움직임 예측 구조

1. 기본 개념

H.264 인코딩 하드웨어는 계산의 복잡도와 IME 대역폭 문제로 인하여 데이터 재사용 및 병렬 구조가 현재의 추세이다. [9]의 연구에서 볼 수 있듯이 움직임 예측을 위해 각각의 현재 블록은 개별적인 탐색영역을 가지고 있으며 이 탐색영역은 이웃한 현재 블록의 탐색영역과 상당부분 겹쳐있다(level C).

그림 1은 연속된 현재블록 A~E에서 공유되는 탐색영역을 나타낸 그림이다. 그림에서와 같이 현재블록 A는 R0~R4의 영역을 탐색하며 현재블록 B는 R1~R5를 탐색한다. 마찬가지로 현재블록 C, D, E는 각각 R2~R6, R3~R7, R4~R8을 탐색하게 된다. 이처럼 A~E 중앙의 참조영역 R4는 연속된 현재블록 A, B, C, D, E에 동시에 참조된다는 것을 알 수 있다. 이와 같이 참조 데이터는 메모리로부터 각 열마다 한번만 로딩되며 연속된 현재블록들 간에 동시에 병렬로 처리됨으로써 메모리 접근율을 감소시킨다.

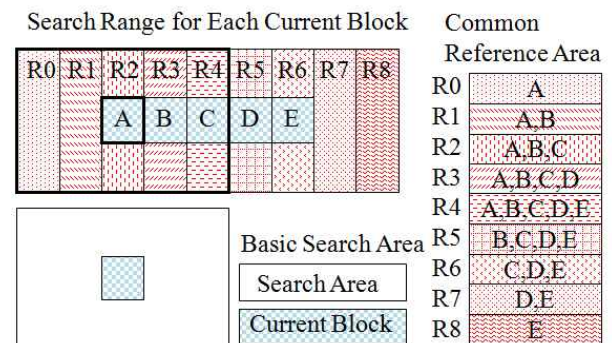


그림 1. 연속된 현재블록들 사이에 공유되는 참조영역
Fig. 1. Current block & shared reference area.

2. 병렬 정화소 움직임 예측 구조

그림 2는 연속된 현재 매크로블록들에 대한 병렬 SAD(Sum of Absolute Differences) 엔진을 보여준다. 참조버퍼로 부터의 한 개의 참조블록은 여러개의 현재 블록들간에 공유되며 동시에 SAD 연산을 하게 된다. 연속된 현재 매크로블록들 간에 중복되는 탐색영역은 재사용 된다. 따라서 제안하는 알고리즘은 [4]의 FSA(Full Search Algorithm)에 비해 낮은 메모리 접근율을 가진다.

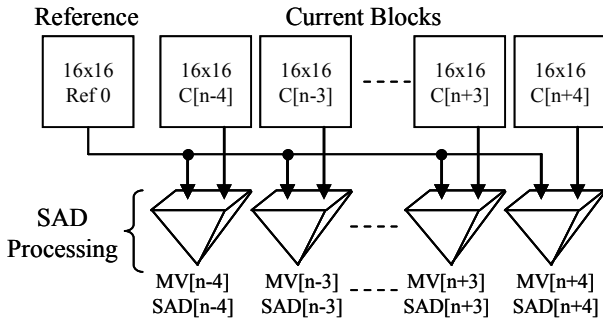


그림 2. 알고리즘 기반의 코어 구조
Fig. 2. Algorithm based the architecture core.

3. 현재와 참조블록의 버퍼 스케줄

그림 3의 (a)는 현재블록 A를 중심으로 본 현재블록과 참조블록의 버퍼 스케줄이다. 프레임의 가장 왼쪽 첫 번째 참조영역이 R0라고 하였을 때, R0는 세 개의 현재블록에 동시에 연산된다. 각 열마다 SAD 연산을 수행하다가 참조버퍼에 R1의 영역이 채워지게 되면 참조블록은 A를 비롯한 4개의 현재블록과 동시에 SAD 연산을 수행하게 된다. 이어서 R2로 이동하였을 때 5개의 현재블록과 동시에 연산을 수행하게 되며 이러한 십자 모양의 패턴은 프레임의 우측 끝까지 유지되고 이 십자 패턴이 프레임의 우측 끝에 도달하면 시작할 때와 마찬가지로 동시에 연산되는 현재블록의 개수는 5개에서 4개 3개로 줄어들게 된다. 현재블록 A는 R0~R4까지의 모든 열에 대해 SAD 연산을 수행하고 R4영역의 연산이 끝남과 동시에 현재블록 A에 대한 MV(Motion

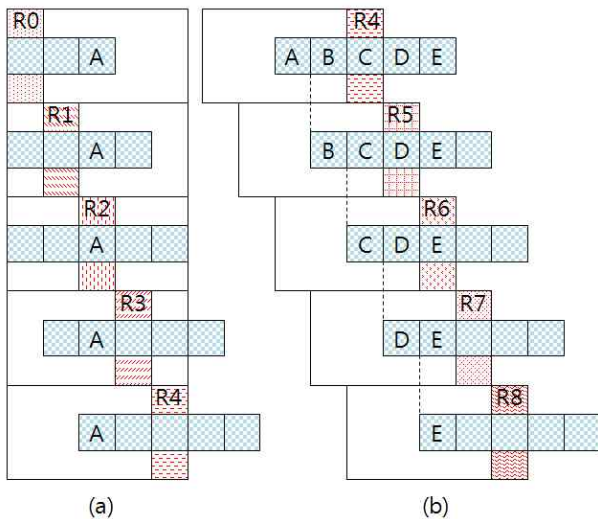


그림 3. (a) 현재와 참조블록의 버퍼 스케줄
(b) 파이프라이닝 순서
Fig. 3. (a) Buffer schedule of Current & reference block
(b) Pipelining order.

vector)를 얻는다.

그림 3의 (b)는 이러한 구조에 의한 파이프라이닝 동작을 나타내는 그림이다. R4 참조영역에 대한 연산이 끝났을 때 현재블록 A에 대한 MV를 얻게 되고, R5 참조영역의 연산이 끝나면 B의 MV를 얻는다. 이처럼 5개의 병렬 SAD엔진에 의해 MV 생성 주기가 1/5로 짧아져 움직임 예측 속도가 증가한다.

III. 부화소 움직임 예측 구조

최적의 부화소 위치를 찾기 위해 참조 모델 소프트웨어 [8]에서는 두 단계 알고리즘을 채택하고 있다. 첫 번째로 여덟 개의 1/2 화소 후보가 최적의 정화소 위치 주변에 정렬된다. 이후 최적의 1/2 화소 위치 주변의 8개의 1/4 화소 후보들 중 최적의 1/4 화소를 찾는다. 이때, 정화소 후보는 정화소 움직임 예측이 끝났을 때 SATD(Sum of absolute transformed differences)로 재계산되어야 한다. 결과적으로 17개의 후보가 최적의 후보를 찾기 위해 사용되며 그림 4 (a)에서 이 알고리즘을 그림으로 나타내었다.

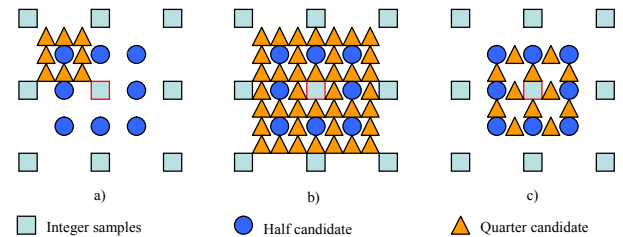


그림 4. (a) Two-step 알고리즘 (b) 49개의 후보 위치
(c) FIHQ 알고리즘
Fig. 4. (a) Two-step algorithm. (b) 49 candidate positions. (c) FIHQ algorithm.

1. Flattened IHQ 알고리즘

참조 모델 소프트웨어의 두 단계 알고리즘은 순차 처리임이 명확하다. 그림 4 (b)는 부화소 움직임 예측 참조 소프트웨어에서 탐색되는 49개의 후보들이다. 그러나 실험에 의해 85%의 후보들이 정화소와 1/2화소 위치 사이 공간, 특히 정화소와 부화소의 두 축에 흩어져 있는 것으로 나타났다. 이 발견을 적용하여 본 장에서는 Flattened Integer-Half-Quarter(FIHQ) 알고리즘을 제안한다. 49개의 후보 대신 1개의 정수 위치, 8개의 1/2화소 정밀도 위치와 1/8화소 정밀도 위치로 이루어진 오직 21개의 후보들이 가장 일치하는 후보를 찾기 위해 탐색된다. 결과적으로 이 IHQ 위치들은 병렬 기법으

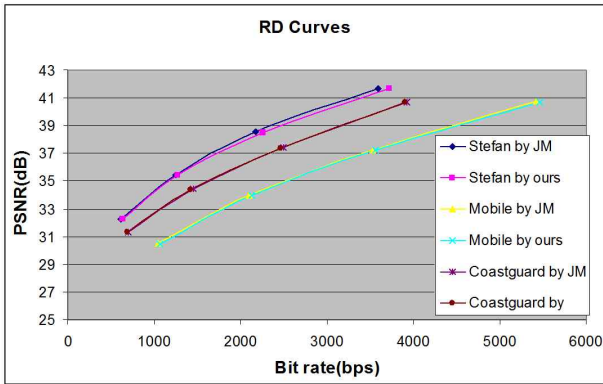


그림 5. JM과의 코딩 품질 비교
Fig. 5. Coding quality comparisons with JM.

로 수행이 가능하다. 이렇게 함으로써 두단계 알고리즘에 비해 사이클 계수는 반으로 감소하고 처리량은 두 배로 증가한다. 이 FIHQ 알고리즘을 그림 4 (c)에 나타내었다. 제안하는 FIHQ 알고리즘을 JM 참조 소프트웨어를 이용하여 몇가지 표준 시퀀스 테스트를 수행하였다. 느리거나 빠른 서로 다른 움직임 레벨을 갖는 Costguard, Mobile, Stefan의 CIF(352x288) 30 fps 동영상 선택되었다. IPPP 시퀀스 형태를 갖는 90 프레임의 Stefan과 100 프레임의 Costguare, Mobile 영상이 울-왜곡 최적화 없이 인코딩되었다. 그림 5는 FIHQ 알고리즘의 시뮬레이션 결과이다. 참조 소프트웨어와의 비교에서 최고 0.02dB의 PSNR 저하를 보이고 있으며 (Stefan) 이는 오차범위 이내의 수준이다.

2. 부화소 움직임 예측 구조

그림 6은 병렬 부화소 움직임 예측 구조를 나타내는 그림이다. 정화소 움직임 예측 모듈로부터 하나의 16x16 매크로블록에 대응되는 41개의 Motion Vector (MV)들이 생성된다. 이때, 부화소 움직임 예측은 FIHQ 알고리즘을 사용하여 가장 일치하는 화소를 찾기

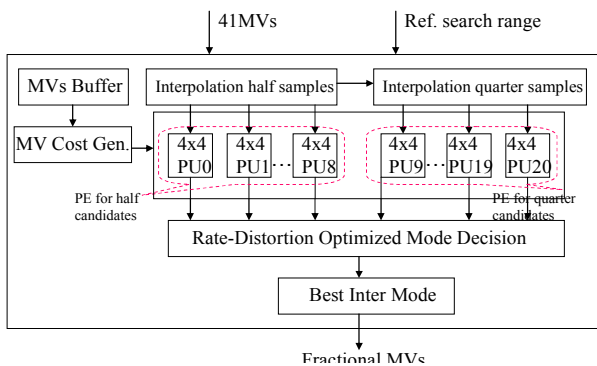


그림 6. 부화소 움직임 예측 구조
Fig. 6. FME architecture.

위해 이 MV들을 정렬한다. 21개의 모든 IHQ 위치들은 동시에 탐색된다. 제안하는 구조는 기존의 고속 알고리즘 [7] 보다 간단한 하드웨어 구조를 지닌다. 게다가 [2],[7]의 디자인과 비교했을 때 FIHQ 알고리즘으로 인해 약 절반의 사이클 계수를 가진다. 21개의 처리 단위는 그들 가운데 최적의 위치를 찾는 계산을 담당하고, 각 4x4 처리 단위는 각 후보의 잔차 발생과 하다마드 변환을 수행한다.

알고리즘의 구현을 위해 보간 부분은 [3]과 유사하며 또한 메모리 요소는 병렬 처리에 포함된다. 여섯 사이클의 지연 시간 후 1/2의 수직 및 대각선 화소들이 생성된다. 1/4 화소 정밀도 신호들은 이중 선형 필터를 통해 한 클럭 사이클 이후 바로 생성된다.

3. 정화소와 부화소 움직임 예측의 파이프라인 구조

정화소와 부화소 움직임 예측은 대량의 계산 복잡성을 포함하는 매우 긴 경로를 가지므로 파이프라인으로 처리 해야 한다. 그러나 같은 매크로블록 단계에서 정화소와 부화소 움직임 예측을 파이프라인 처리 하는 것은 매우 낮은 활용도를 갖게 된다. 그러므로 부화소 움직임 예측은 처리량 향상을 위해 정화소 움직임 예측 이후의 매크로블록에 대하여 파이프라인 처리 되었다.

그림 7은 정화소와 부화소 움직임 예측의 파이프라인 구조를 보여준다. 각각의 현재 움직임 예측은 완전히 겹쳐지지 않는 개별적인 탐색영역을 지니고 있기 때문에 N개의 현재 매크로블록들이 정화소 움직임 예측 부분에서 병렬 처리되고 있을지라도 41개의 MV로 이루어진 N개의 그룹들은 부화소 움직임 예측의 현재 매크로블록을 위해 한 개의 그룹씩 순차적으로 입력된다. 부화소 움직임 예측의 입력 제어는 파이프라이닝을 위해 41개의 MV 버퍼 집합이 필요하다. 마지막 모듈은 Lagrangian 모드 결정이다. 이 모듈은 오직 1/4 화소

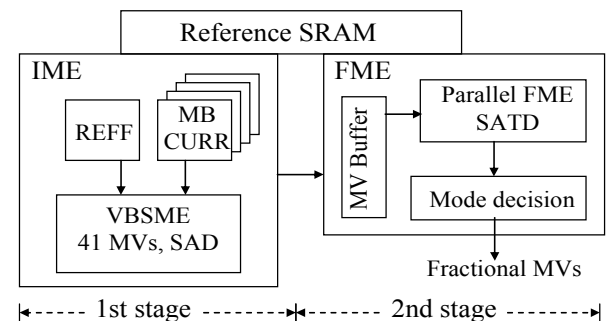


그림 7. 정화소와 부화소 움직임예측 파이프라인 구조
Fig. 7. IME/FME pipeline architecture.

정밀도에서 7 종류의 블록 크기에 대한 연산을 끝마친 이후에 수행된다. 이는 PSNR을 1,0 ~ 1,5 dB 증가시켜 코딩 성능을 향상시킨다.

IV. 실험 결과

1. 메모리 접근율 비교

제안하는 알고리즘은 정확소와 부화소 움직임 예측 구조에 의해 기존의 디자인보다 낮은 메모리 접근율을 가진다.

표 1은 제안하는 구조와 FSA(Full Search Algorithm) 기본 알고리즘 및 FSA를 기반으로 한 최근의 데이터 재사용 알고리즘과의 메모리 접근율을 비교한 그림이다. 해상도가 높아질수록 제안하는 움직임 예측 구조가 뛰어난 메모리 접근율을 지님을 알 수 있다. FSA는 데이터 재사용 없이 Full HD 인코딩을 위해 32GByte/s의 메모리 접근율을 가지며 [3]은 1.9GByte/s의 메모리 접근율을 요구한다. 이들 알고리즘과 비교하여 본 논문에서 제안하는 구조는 Full HD에서 0.3GByte/s로 [3]의 데이터 재사용 알고리즘에 비해 6 배 이상의 메모리 접근률 감소를 보인다.

표 1. 움직임 예측을 위한 해상도별 메모리접근율 비교

Table 1. Memory access rate for motion estimation on each video resolutions.

Video Resolution	Full Search	Chen[3]	Ours
QCIF	59.4MB/sec	23.2MB/sec	2.9MB/sec
CIF	23760MB/sec	92.81MB/sec	11.6MB/sec
D1	79.1GB/sec	0.31GB/sec	0.04GB/sec
720p	210.9GB/sec	0.82GB/sec	0.1GB/sec
1080p	478.1GB/sec	1.86GB/sec	0.23GB/sec

2. 병렬 움직임 예측 코어 제작

정확소와 부화소 움직임 예측 코어는 Chartered 0.18μm CMOS 1.8V/3.3V 1-poly 5-metal 공정을 사용한 MPW(Multi-Project Wafer)를 통해 칩으로 제작되었다. 그림 8의 (a)는 제작된 움직임 예측 코어 사진이며, (b)는 움직임 예측 코어 칩 패키지 사진이다. 자세한 내용은 표 1에 정리하였다. 움직임 예측 코어는 메모리 성분을 포함하여 907k 로직 게이트로 합성되었다.

설계된 움직임 예측 코어는 94MHz에서 HDTV 720p 30fps를 실시간 부호화 가능하다. 제안하는 알고리즘의

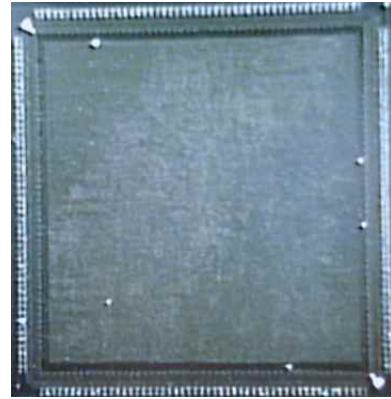


그림 8. 제작된 H.264 움직임 예측 코어
Fig. 8. Picture of IME/FME core and chip.

표 2. 움직임 예측 칩의 특성
Table 2. Chip specification of motion estimation.

Design	움직임 예측 코어
Max Spec.	1280 × 720 × 30p
Search range	64 × 64
Inter modes	4×4, 4×8, 8×4, 8×8, 8×16, 16×8, 16×16
Technology	Charard 0.18 μm CMOS 1P5M
Logic gates(total)	907k gates
IME	783k (including memory elements)
FME	124k
Clock Freq.	94 MHz

병렬 구조와 메모리 접근율 감소를 위한 버퍼들로 인해 게이트 수는 증가되었다. 제작된 칩을 테스트하기 위해 Agilent 16700B tester를 사용하였다. Post-P&R 시뮬레이션 결과로서 추출된 벡터 파일을 Agilent 16700B tester를 통해 나오는 칩의 출력 결과와 비교하고 분석함으로써, 칩 테스트 과정이 수행되었다.

3. 처리량 비교

표 3은 기존 [10~12]의 디자인과 제안하는 정수 움직임 예측 디자인을 비교한 표이다. 제안하는 디자인은 가장 낮은 Latency를 가지며 43.2MHz에서 720p HD 영상을 초당 30프레임 움직임 예측이 가능하다.

표 4는 기존 [2, 7]의 디자인과 제안하는 부화소 움직

표 3. 기존 정확소 움직임 예측 디자인들과의 비교
Table 3. Comparison of ime with previous design.

	Yap[10]	Kim[11]	Su[12]	Ours
Max.resolution	4CIF@30fps	4CIF@30fps	720p@30fps	720p@30fps
Latency(cycle)	4096	1024	972	400
Operating Freq.(MHz)	294 for 4CIF@30fps	100 for 4CIF@30fps	105 for 4CIF@30fps	43.2 for 4CIF@30fps

표 4. 기존 부화소 움직임 예측 디자인들과의 비교
Table 4. Comparison of fme with previous designs.

	Chen[2]	Wang[7]	Ours
Max.resolution	720x576@30fps	720x576@30fps	720p@30fps
Latency(cycle)	1648	2000	862
Operating Freq.	100 MHz	100 MHz	94 MHz
Throughput	49 kMBs/s	50 kMBs/s	109 kMBs/s

임 예측 디자인의 비교를 보여준다. 제안하는 구조는 FIHQ 알고리즘으로 인해 [2, 7]과 비교하여 단지 43%와 52%의 latency를 가진다. 따라서 제안하는 구조는 [2, 7]에 비해 두 배의 처리량을 가지며 94MHz에서 720p HDTV 30fps를 지원 가능하다.

V. 결 론

본 논문에서 우리는 정화소와 부화소 움직임 예측을 위한 완전한 병렬 구조를 제안하고 설계하였다. 정화소 움직임 예측의 현재 매크로블록 병렬 처리는 FSA에 비해 100배 이상의 메모리 접근을 감소시켰다. FIHQ 구조는 1/2와 1/4 화소 정밀도 후보를 위해 단지 한 단계의 보간을 필요로 하며 처리량은 2배 증가되었다. 파이프라인 처리된 정화소와 부화소 움직임 예측 구조는 기존의 디자인에 비해 향상된 처리량을 달성하여 HDTV 720p 30fps를 94MHz로 실시간 부호화 지원이 가능하다. 제안하는 구조의 움직임 예측 코어는 Chartered 0.18 μ m 공정의 MPW를 통해 칩으로 구현되었으며, Agilent 16700B tester를 이용하여 제작된 칩을 검증하였다.

감사의 글

본 연구는 IT-SoC 핵심인력양성사업과 IDEC의 지원으로 이루어졌으므로 이에 감사드립니다.

참 고 문 헌

[1] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7-28, First Quarter, 2004.
 [2] T. C. Chen, Y. W. Huang, and L. G. Chen,

"Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC," *ICASSP*, vol. 5, pp. 9-12, May. 2004.
 [3] T. C. Chen, S. Y. Chien, Y. W. Huang, C. H. Tsai, C. Y. Chen, Y. W. Chen, and L. G. Chen, "Analysis and architecture design of an HDTV. 720p 30frames/s H.264/AVC encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol.16, no.6, pp.677-679, June 2006.
 [4] J. C. Tuan, T. S. Chang, and C. W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61-72, Jan. 2002.
 [5] D. X. Li, W. Zheng, and M. Zhang, "Architecture design for H.264/AVC integer motion estimation with minimum memory bandwidth," *IEEE Trans. Consumer Electron.*, vol. 53, no. 3, pp. 1053-1060, Aug. 2007.
 [6] Z. Liu, Y. Song, M. Shao, S. Li, L. Li, S. Ishiwata, M. Nakagawa, S. Goto, and T. Ikenaga, "HDTV 1080p H.264/AVC encoder chip design and performance analysis," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 2, pp. 594-606, Feb. 2009.
 [7] Y. J. Wang, C. C. Cheng, and T. S. Chang, "A fast algorithm and its VLSI architecture for fractional motion estimation for H.264/MPEG-4 AVC coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 578-583, May. 2007.
 [8] JVT H.264/AVC Reference Software JM 12.2.
 [9] J. S. Youn, J. R. Choi, and S. S. Han, "Parallel integer motion estimation method by using reference blocks shared for HD video coding," *2009 International Conference on Electronic Computer Technology*, pp. 577-581, Feb. 2009.
 [10] S. Y. Yap and J. V. McCanny, "A VLSI architecture for variable block size video motion estimation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 7, pp. 384-389, Jul. 2004.
 [11] M. Kim, I. Hwang, and S. I. Chae, "A fast vlsi architecture for full search variable block size motion estimation in MPEG-4 AVC/H.264," in *Proc. Asia South Pacific Design Autom. Conf.*, vol. 1, pp. 631-634, Jan. 2005.
 [12] C. L. Su, W. S. Wang, U. L. Chen, Y. C. Wang, C. W. Chen, J. I. Guo, and S. Y. Tseng, "A low complexity high quality integer motion estimation architecture design for H.264/AVC," in *Proc IEEE Asia Pacific Conf Circuits Syst.*, pp. 398-401, Dec. 2006.

— 저 자 소 개 —



김 시 혜(학생회원)
 2009년 경북대학교 전자전기컴퓨터학부 학사 졸업.
 2011년 경북대학교 전자전기컴퓨터학부 석사 재학.
 <주관심분야 : H.264/AVC, 임베디드 시스템, VLSI 및 SoC 설계>



최 준 림(정회원)-교신저자
 1986년 연세대학교 전기공학과 학사 졸업.
 1988년 (미)Cornell Univ. 전자전기공학과 석사 졸업.
 1991년 (미)Minnesota Univ. 전자전기공학과 박사 졸업.
 1991년~1997년 LG전자기술원 책임연구원.
 1997년~현재 경북대학교 교수.
 2000년~현재 경북대 IDEC 지역센터장.
 <주관심분야 : System On Chip, 마이크로 센서, 디지털 시스템 설계>