

논문 2007-44CI-5-2

실시간 H.264/AVC 처리를 위한 ASIP설계

(ASIP Design for Real-Time Processing of H.264)

김진수*, 선우명훈*

(Jin Soo Kim and Myung Hoon Sunwoo)

요약

본 논문에서는 ASIP (Application Specific Instruction-set Processor) 기반의 실시간 H.264/AVC 구현 가능한 VSIP (Video Specific Instruction-set Processor) 을 제안한다. 제안한 VSIP은 H.264/AVC의 화면 내 예측, 디블록킹 필터, 정수 변환 등 새로운 기능들을 효율적으로 지원하기 위한 전용의 하드웨어 구조와 명령어를 가지고 있다. 또한 화면 간 예측 및 엔트로피 코딩과 같이 연산량이 많은 부분은 하드웨어 가속기로 만들어 연산 처리 속도 및 효율을 높였다. VSIP은 H.264/AVC에 적합한 하드웨어 구조와 명령어를 통해 기존의 디지털 신호처리 프로세서보다 작은 크기를 가지며, 메모리 접근 횟수를 줄여 전력 소비를 감소시켰다. 제안한 VSIP을 이용하여 실시간 영상 신호처리를 할 수 있으며, 다양한 프로파일과 표준을 지원할 수 있다.

Abstract

This paper presents an ASIP (Application Specific Instruction Set Processor) for implementation of H.264/AVC, called VSIP (Video Specific Instruction-set Processor). The proposed VSIP has novel instructions and optimized hardware architectures for specific applications, such as intra prediction, in-loop deblocking filter, integer transform, etc. Moreover, VSIP has hardware accelerators for computation intensive parts in video signal processing, such as inter prediction and entropy coding. The VSIP has much smaller area and can dramatically reduce the number of memory access compared with commercial DSP chips, which result in low power consumption. The proposed VSIP can efficiently perform in real-time video processing and it can support various profiles and standards.

Keywords: ASIP (Application Specific Instruction-set Processor), HW/SW co-design, H.264/AVC, low power design, hardware accelerator

I. 서론

최근 통신 및 멀티미디어 관련 규격들의 요구사항이 고도화되면서 주문형 반도체(Application Specific Integrated Circuit) 기반의 시스템으로는 개발 기간의 장기화, 설계비용 증대, 설계 변경의 어려움 등의 문제로 빠르게 변화하는 기술 및 시장에 대처할 수 없다. 반면 디지털 신호처리 프로세서(Digital Signal Processor) 기반의 시스템들은 광범위한 응용을 위해 설계되어 다양한 분야에 쓰일 수 있으나 주문형 반도체보다 상대적으로 높은 가격과 전력 소모 그리고 낮은 성능을 가진

다. 그에 비해 ASIP (Application Specific Instruction-set Processor) 은 디지털 신호처리 프로세서의 장점인 다양한 응용에 사용 가능한 적용성과 주문형 반도체의 장점인 최적 설계에 따른 저전력, 및 고성능을 동시에 지니고 있어, 다양하고 빠르게 변화하고 있는 멀티미디어 통신 시스템의 개발에 적합하다^[1~5].

멀티미디어 관련 기술 표준은 MPEG-2, MPEG-4, H.261, H.263 등이 있으며 영상의 손실을 최소화하며 압축효율을 높이는 연구가 이루어지고 있다. 멀티미디어 표준을 주도해온 단체인 MPEG(Motion Picture Expert Group)과 ITU(International Telecommunication Union) 는 기존의 멀티미디어 표준보다 더 큰 압축 효율을 갖는 표준을 제정하고자 공동의 표준 연구 기관인 JVT (Joint Video Team) 를 설립하였으며 이 단체는 2003년

* 정회원, 아주대학교 전자공학과
(Department of Electrical Engineering, Ajou University)

접수일자: 2007년6월26일, 수정완료일: 2007년9월4일

12월 새로운 멀티미디어 표준인 H.264/AVC를 확정하였다^[6]. 새로 제정된 H.264/AVC는 현재 가장 많이 사용되고 있는 MPEG-4 simple profile에 비하여 2배 이상 압축 효율이 좋지만, 부호화기의 경우 10배, 복호화기의 경우 2배 이상 연산 복잡도가 높아 하드웨어 구현에 어려움이 많다^[7].

또한 H.264/AVC를 이동 통신 기기에 구현하기 위해서는 높은 성능 가지면서 적은 전력을 소모해야 한다. 또한 시스템의 업그레이드가 간편해야 한다. ASIP 기반의 H.264/AVC 설계는 높은 성능과 낮은 전력 소모 및 쉬운 시스템 업그레이드를 제공할 수 있기 때문에 이동 통신 기기에 구현하기 적합한 설계 방법이다. 본 논문에서는 ASIP 기반의 실시간 H.264/AVC 복호화 가능한 영상 신호처리 전용 프로세서인 VSIP (Video Specific Instruction-set Processor) 을 제안하였다.

VSIP은 H.264/AVC의 연산량 분석을 통해 하드웨어로 처리할 부분과 소프트웨어로 처리할 부분으로 나누었으며, 화면 내 예측, 디블록킹 필터 및 정수 변환 등의 새로운 기능들을 적합한 하드웨어 구조 및 명령어들을 추가해 보다 효율적인 연산을 수행할 수 있다.

VSIP은 QCIF 크기의 영상을 H.264/AVC baseline profile에 맞추어 실시간으로 복호화 할 수 있으며, Xilinx XC2v6000 FPGA를 이용해 검증하였다.

본 논문의 II장에서는 H.264/AVC 특징과 기존 디지털 신호처리 프로세서의 멀티미디어에 관련된 명령어들을 소개하고, III장에서는 H.264/AVC를 효율적으로 지원하기 위한 VSIP의 하드웨어 가속기와 새로운 명령어들을 제안한다. IV장에서는 성능 비교에 대하여 서술하고, 마지막 V장에서는 결론에 대하여 논한다.

II. H.264/AVC 특징 및 기존 멀티미디어 명령어 분석

이 장에서는 기존의 다양한 H.264/AVC 특징을 분석하고, 기존 디지털 신호처리 프로세서의 멀티미디어 관련 명령어를 살펴본다.

1. H.264/AVC의 특징

H.264/AVC는 부호화된 영상의 화질 향상을 위해 여러 장의 참조 영상, 다양한 크기의 블록 그리고 1/4 픽셀 정확도의 움직임 추정 및 움직임 보상 등 새로운 기능들을 적용하였다. 또한 첫 번째 영상을 화면 내 예측이라는 새로운 방법을 사용하여 압축 후 전송한다. 위

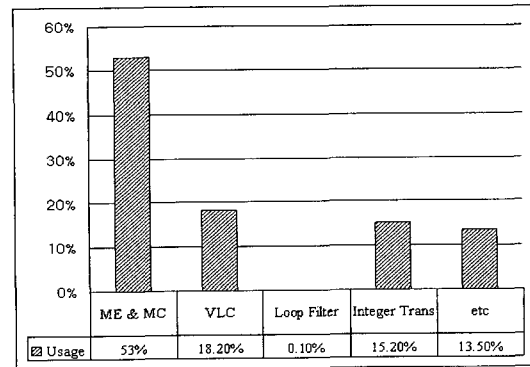


그림 1. H.264/AVC baseline profile의 연산량 분석

Fig. 1. Complexity analysis of the H.264/AVC baseline profile.

와 같은 기능들은 압축률과 화질 향상에 기여하지만, 메모리 접근 횟수와 연산량을 크게 증가시켜 높은 전력 소모를 발생시킨다.

일반적으로 움직임 추정 및 보상은 블록 기반의 탐색 방법이 사용된다. 블록 기반의 탐색은 연산이 간편하다는 장점이 있지만, 블록화 현상 (blocking artifact) 을 발생시켜 영상의 화질을 떨어뜨리는 원인이 된다. H.264/AVC에서는 이러한 블록화 현상의 제거를 위해 디블록킹 필터를 사용한다. 그리고 H.264/AVC baseline profile의 경우 엔트로피 코딩으로 EGC(Exponential Golomb Coding)와 CAVLC(Context Adaptive Variable Length Coding) 을 채택하였다. EGC는 규칙적인 구조를 가지는 엔트로피 코드이며^[8], CAVLC는 4 x 4 블록의 오차 데이터를 압축하는데 사용된다^[9~11].

그림 1은 H.264/AVC baseline profile의 연산량을 나타내었다^[12]. 움직임 추정과 보상 및 엔트로피 코딩은 전체 연산량의 53%와 18.20%를 차지한다. 특히 움직임 추정과 보상 연산은 빈번한 메모리 접근을 발생시켜 전력 소비가 많다. 그리고 엔트로피 코딩은 비트 단위의 연산이 필요하므로 기존 디지털 신호처리 프로세서로 구현하려면 비효율적이다. 따라서 위 연산에 적합한 하드웨어 구조가 필요하다. VSIP은 그림 1의 연산량 분석을 바탕으로 복잡도와 전력 소비를 줄이기 위한 하드웨어 구조와 명령어들을 제안하였다^[5].

그림 2는 디지털 신호처리 프로세서, 주문형 반도체와 VSIP이 하나의 매크로 블록을 처리하는데 걸리는 연산 시간 및 방법을 보여준다.

그림 2(a)는 디지털 신호처리 프로세서로 구현했을 때의 연산 시간으로 모든 연산을 순차적으로 실행해야 하므로 연산시간이 가장 길다. 그림 2(b)는 주문형 반도체의 연산 시간으로 연산을 병렬로 수행 할 수 있어 가

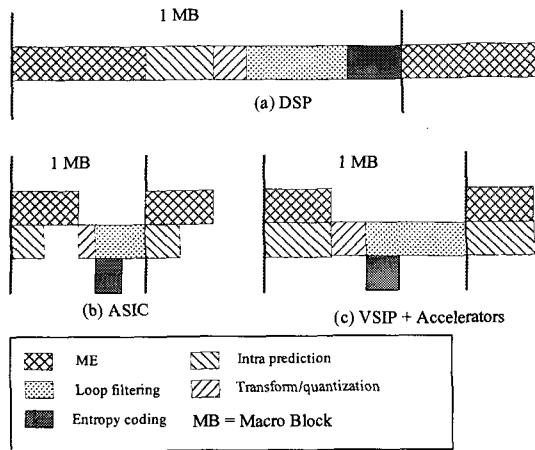


그림 2. 다양한 구현 방법의 연산 비교
 Fig. 2. Computation comparisons of various implementations.

장 연산시간이 짧다. 그림 2(c)는 VSIP 을 이용한 경우로 주문형 반도체보다 연산 시간이 길지만, 디지털 신호처리 프로세서보다는 연산 시간이 짧으면서 다양한 표준을 지원할 수 있다는 장점이 있다.

2. 기존 디지털 신호처리 프로세서의 멀티미디어 관련 명령어

기존의 DSP 들은 두 개의 레지스터 사이의 packed operation을 지원하는 다양한 명령어를 제공하고 있다. 이러한 packed operation은 DCT (Discrete Cosine Transform), IDCT (Inverse Discrete Cosine Transform), 움직임 추정, 움직임 보상 등 다양한 영상 신호처리에 사용되고 있다. Texas Instruments의 TMS320c6x는 멀티미디어 데이터의 연산을 위해 SUBABS4, AVGx, DOTPU4 등의 명령어를 지원하고 있다^[13]. SUBABS4는 레지스터의 절대값의 차를 계산하는 명령어이며, AVG4는 두 개의 레지스터의 평균값을 계산하는 명령어이다. 그리고 DOTPU4 명령어는 4개의 8비트 값들의 곱셈 연산을 하며, 4 클럭 사이클이 필요하다.

TMS320c55x는 DCT 연산을 위한 코프로세서를 가지고 있으며, QCIF 크기의 영상 30fps 처리하기 위해 2.8 MIPS를 필요로 하며, TMS320c6x는 8개의 연산 유닛을 가지고 있으며, QCIF 영상 30fps 처리하기 위해 1.1 MIPS가 필요하다^[14].

엔트로피 코딩은 입력 비트 스트림에서 연속되는 1의 개수에 따라서 코드 워드 테이블을 참조하는데, packed 비교 연산이 필요하다. TMS320c64x는 엔트로피 연산을 위해 LMBD와 CMPEQ/GT/LT 명령어를 지

원하며 Analog Device의 Blackfin 은 ONES 명령어를 지원하고 있다^[14~15].

III. 제안한 명령어들과 하드웨어 가속기

이 장에서는 H.264/AVC를 효과적으로 지원하기 위한 새로운 명령어들과 하드웨어 가속기에 대해서 제안한다.

1. H.264/AVC 구현

그림 3은 VSIP의 전체적인 구조도이며, 디지털 신호처리 프로세서와 하드웨어 가속기 두 부분으로 이루어져 있다. 디지털 신호처리 프로세서 부분은 프로그램 카운터 유닛과 데이터 프로세싱 유닛 그리고 주소 유닛으로 구성되어 있다. 하드웨어 가속기 부분은 화면 내 예측 가속기와 엔트로피 코딩 가속기로 이루어져 있으며, 디지털 신호처리 프로세서와 동시에 연산을 수행할 수 있다.

VSIP은 prefetch, fetch, decode, execute1, execute2, execute3의 총 6 단계의 파이프라인으로 이루어져 있다. 그리고 35개의 산술 명령어와 11개의 논리 및 쉬프트 명령어, 6개의 프로그램 제어 명령어, 4개의 이동 명령어 그리고 16개의 H.264/AVC를 위한 전용 명령어들로 이루어져 있다.

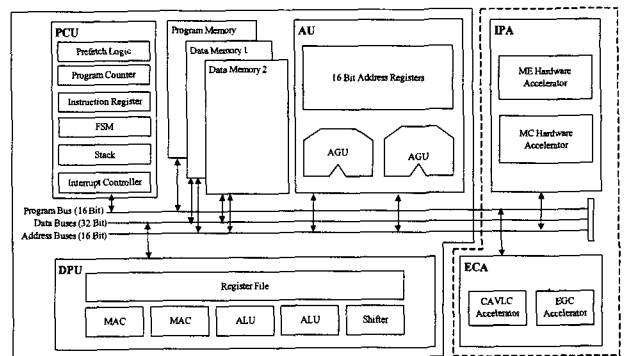


그림 3. 제안한 VSIP 하드웨어 구조
 Fig. 3. Proposed VSIP architecture.

2. 디블록킹 필터와 화면 내 예측을 위한 명령어 제안

앞서 말했듯이 디블록킹 필터는 블록화 현상의 제거에 사용된다. 그림 4는 두 개의 이웃한 4 x 4 블록의 픽셀들이며, p0~p3과 q0~q3의 픽셀 값은 디블록킹 필터를 거친 후 새로운 값으로 재정의된다. 디블록킹 필터의 연산 수식^[6]은 5종류로 분류할 수 있다.

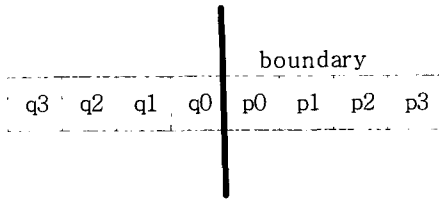


그림 4. 블록 경계
Fig. 4. Block boundary.

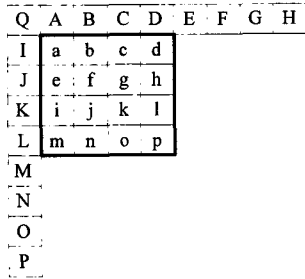


그림 5. 4 x 4 화면 내 예측
Fig. 5. 4 x 4 intra prediction.

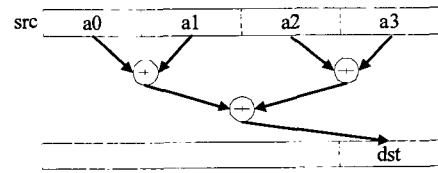
- (1) $p2 + p1 + p0$
- (2) $p2 + 2 \times p1 + 2 \times p0$
- (3) $2 \times p3 + 3 \times p2 + p1 + p0$
- (4) $2 \times p1 + p0$
- (5) $(p0 + q0 + 1) \gg 1$

화면 내 예측은 영상 내 중복성을 제거하여 압축을 수행한다. 그림 5은 4 x 4 블록에 대한 화면 내 예측 방법이다. a~p 픽셀들은 A~Q의 픽셀 값을 이용해 수식 (6)에 따라 계산되며, (6)은 화면 내 예측 방법에 관한 수식이다.

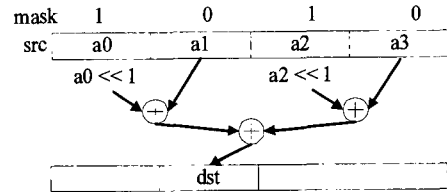
$$\begin{aligned} (A + 2 \times B + C + 2) \gg 2 \\ (A + B + 1) \gg 1 \\ (A + 3 \times B + 2) \gg 2 \end{aligned} \tag{6}$$

위의 디블록킹 필터와 화면 내 예측 연산들은 레지스터 사이의 연산뿐만 아니라 레지스터 내의 연산을 필요로 한다. 하지만 기존 디지털 신호처리 프로세서는 두 개의 레지스터 사이의 연산만을 지원하고 있어 기존의 명령어로 위의 연산들을 수행하기 위해서는 많은 연산 사이클이 필요하다. 따라서 레지스터 내의 연산을 수행할 수 있는 명령어를 제안하였다.

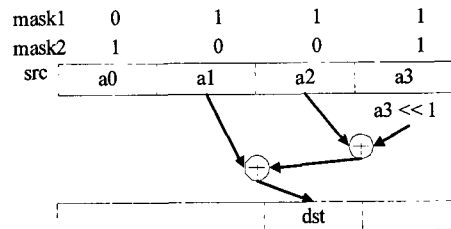
그림 6은 제안한 세 개의 레지스터 내 덧셈 명령어 (HADD) 이다. 그림 6 (a)은 레지스터 내의 8비트 데이터 4개를 더하는 연산이고, 그림 6 (b)은 마스크를 사용



(a) $dst = HADD(src)$



(b) $dst = HADD(src;mask1)$



(c) $dst = HADD(src;mask1;mask2)$

그림 6. 제안한 레지스터 내 덧셈 연산 명령어
Fig. 6. Proposed instructions for packed additions within a register.

해서 원하는 8비트 데이터를 쉬프트 한 후 더하는 연산이다. 그림 6 (c)은 두 개의 마스크를 사용하며, 하나는 쉬프트 연산을 위해 하나는 연산을 수행할 8비트 데이터 값을 선택하는데 사용된다. 제안한 HADD 명령어는 레지스터 내 연산이 가능하므로 효과적으로 디블록킹 필터와 화면 내 예측 연산을 수행 할 수 있다.

3. 정수 변환을 위한 명령어 제안

4 x 4 정수 변환은 레지스터 내 연산을 필요로 하며, 일정한 연산 흐름을 가지고 있다. 순방향 및 역방향 4 x 4 정수 변환을 위해서 새롭게 fTRAN과 iTRAN 명령어를 제안했다.

그림 7(a)는 fTRAN 연산의 흐름을 보여준다. fTRAN 연산은 레지스터 파일에서 32비트 데이터를 읽어 연산을 수행한다. fTRAN은 하나의 행에 대해서만 연산을 수행하므로, 1D 변환을 수행하기 위해서는 총 4 번을 수행해야 한다. 그 후 TRAN 명령어를 이용해서 전치한 후 다시 4번의 fTRAN 명령어를 수행하면 2D 변환이 수행된다. iTRAN는 1D 역변환 명령어이며,

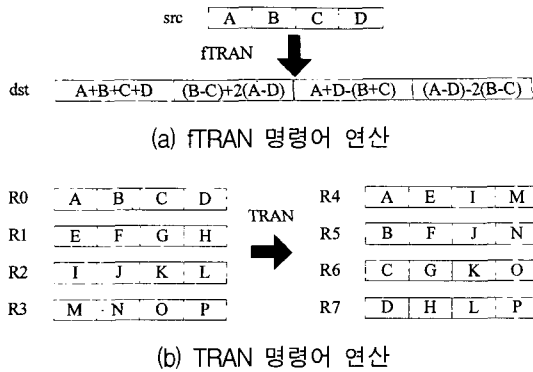


그림 7. fTRAN과 TRAN 명령어의 연산흐름
Fig. 7. Operation flow of fTRAN and TRAN instruction.

fTRAN과 비슷한 방법으로 연산을 수행된다. 그림 7 (b)는 TRAN 명령어의 연산이며, 4 x 4 매트릭스를 전치한다.

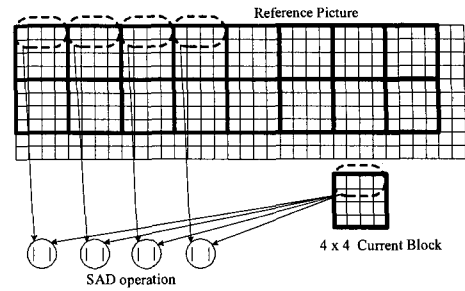
4 x 4 정수 변환은 제안한 명령어를 사용하면 19 클럭이 필요하며, 데이터 읽는 부분이 추가되면 총 23 사이클이 소모된다. TMS320c55x 명령어를 이용해 정수 변환을 수행하면 1078 클럭 이상이 소모되는 것^[16]에 비해 제안한 fTRAN과 iTRAN 명령어를 이용하면 23 클럭이 필요하므로, 기존의 디지털 신호처리 프로세서보다 효율적으로 정수 변환 연산을 수행할 수 있다.

4. 화면 간 예측을 위한 하드웨어 가속기

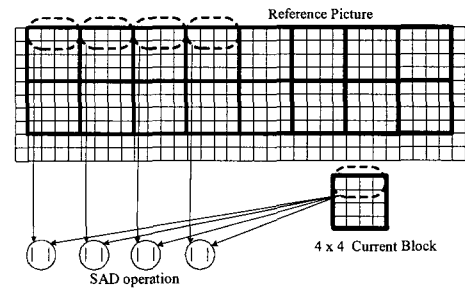
제안한 움직임 보상 하드웨어 가속기는 H.264/AVC의 1/4 픽셀 정확도를 지원한다. 움직임 예측과 보상은 빈번한 메모리 접근을 발생시켜 성능 저하와 높은 전력 소모를 발생시킨다. 슬라이딩 윈도우 방법의 움직임 추정을 이용하면 위의 문제들을 줄일 수 있다^[17]. 그림 8은 제안한 움직임 추정 연산 흐름을 보여준다.

제안한 움직임 추정 구조는 [+8, -7]의 탐색 영역을 가지고 있다. [+8, -7]의 탐색 영역 내에는 16개의 4 x 4 블록이 존재한다. 첫 번째 클럭 사이클에서 그림 8(a)과 같이 블록의 첫 행의 SAD 값이 4개 블록에 대해서 동시에 계산된다. 그 후 탐색 영역이 오른쪽으로 이동하고 그림 8(b)과 같이 SAD 연산을 수행한다. 탐색 영역 내 모든 블록의 첫째 행 SAD 값들은 4 클럭이 지난 후에 얻을 수 있으며, 16개의 SAD 값들은 버퍼에 저장된다. 블록 내 두 번째 행의 4개의 픽셀 데이터도 같은 방법으로 계산되어 16개의 SAD 값들이 누적되어 버퍼에 저장된다. 위와 같은 방법으로 16 클럭 후 16개 4 x 4 블록에 대한 SAD 값들을 얻게 된다.

그림 9는 일반적인 구조 [18]에서의 움직임 추정 연

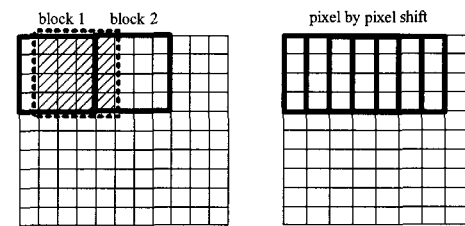


(a) 첫 번째 클럭에서의 움직임 추정 연산



(b) 두 번째 클럭에서의 움직임 추정 연산

그림 8. 제안한 움직임 추정 연산 방법
Fig. 8. Operation flow of the proposed motion estimation.



(a) 기존 연산 흐름 (b) 제안한 연산 흐름

그림 9. 움직임 추정 연산 흐름
Fig. 9. ME computation flow.

산을 보여준다.

그림 9(a)의 블록 1의 SAD 값들을 얻은 후에 점선으로 된 블록의 픽셀들의 SAD 값들을 계산한다. 하지만 4 x 4 블록이 그림 9 (b)처럼 픽셀 단위로 이동한다면, 그림 9(a)의 점선으로 된 블록의 데이터를 다시 계산할 필요 없이 해당하는 영역의 계산된 데이터를 가져다 사용하면 된다. 따라서 메모리 접근 횟수를 줄여 전력 소비를 줄일 수 있다.

5. 엔트로피 코딩을 위한 가속기

H.264/AVC는 엔트로피 코딩으로 EGC와 CAVLC를 사용한다. CAVLC 부호화기는 입력 데이터에 따라 lookup 테이블에서 코드워드의 길이와 값을 읽어온다. 따라서 효율적인 메모리 주소 생성기가 필요하다.

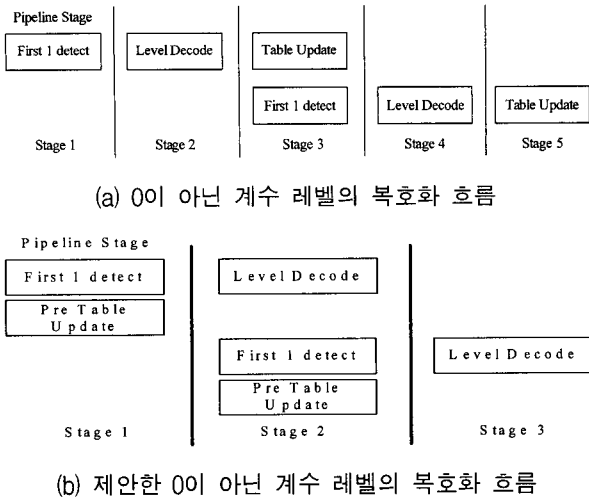


그림 10. 0이 아닌 계수 레벨 복호화 연산 흐름
 Fig. 10. Comparison of flows for the level of the nonzero coefficient decoding

0이 아닌 계수 레벨의 복호화 과정이 그림 10(a)에 나와 있다. 레벨 복호화 과정은 lookup 테이블 업데이트 과정이 끝나기 전에는 수행할 수 없으므로 두 개의 심볼을 복호화 하기 위해서는 다섯 단계의 파이프라인이 필요하다. 그림 10(b)는 0이 아닌 계수를 복호화 하기 위해 제안한 사전 테이블 업데이트 방법이다. 두 개의 심볼을 복호화 하기 위해서 3 단계의 파이프라인이 필요하므로 2 클럭을 줄일 수 있다.

IV. 성능 비교 및 구현

VSIP과 하드웨어 가속기를 이용해서 H.264/AVC를 구현하였다. 제안한 VSIP은 Verilog HDL을 이용해서 설계하였고 그림 11의 iPROBETM의 XilinxTM VirtexII FPGA를 이용해서 검증하였다.

영상의 기본 블록들에 대해 화면 내 예측과 디블록킹 필터를 적용하여 실험하였다. 제안한 전용 명령어를

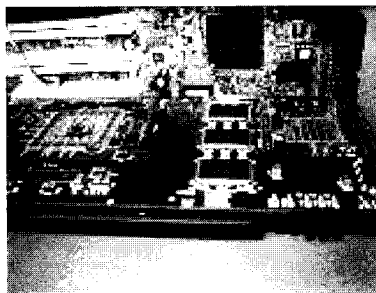


그림 11. FPGA 에뮬레이션
 Fig. 11. FPGA Emulation.

표 1. 4 x 4 블록의 정수 변환의 성능 비교
 Table 1. Performance comparisons of 4 x 4 integer transform.

	TMS320c55x (SW)[16]	TMS320c55x (HW)[16]	TMS320c64x [14]	Proposed VSIP
MIPS	12.8	2.8	1.1	1.1

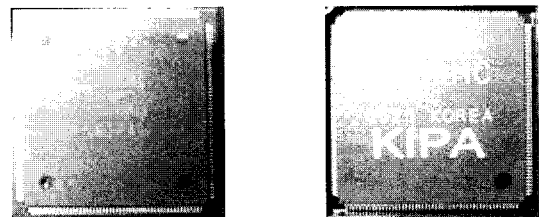
가진 VSIP과 기존의 명령어를 가지고 있는 TMS320c64x를 가지고 부호화 해보았다. 제안한 하드웨어는 TMS320c6x보다 연산에 필요한 클럭을 40% 감소시킬 수 있으며, 디블록킹 필터는 20~25% 감소시켰다.

4 x 4 블록의 정수 변환경우는 제안한 명령어로 수행하기 위해서는 23 클럭이 걸려, 30 프레임의 QCIF 영상을 처리하기 위해서는 1,092,960 클럭이 필요하다.

표 1에 기존의 디지털 신호처리 프로세서^{[14][16]}와 제안한 VSIP를 이용해서 30 프레임의 영상을 정수 변환을 하는데 필요한 명령어 수를 비교하였다. VSIP은 TMS320c55x의 명령어와 TMS320c55x의 코프로세서를 이용해서 정수 변환을 수행했을 때보다 효율적임을 볼 수 있다. TMS320c64x는 VLIW 구조임에 비해, VSIP은 두 개의 32비트 덧셈기만 필요로 한다.

VSIP의 명령어들은 연산의 복잡도와 중복성 그리고 낭비를 최소화하게 제안되었으며, 기존의 디지털 신호처리 프로세서보다 적은 연산 사이클을 필요로 한다. 따라서 VSIP은 메모리 접근 횟수를 줄여, 전력 소비를 줄일 수 있다.

그림 12의 하드웨어 가속기는 SynopsysTM의 Astro를 이용해 MagnaChip HSI 0.25μm 표준 셀 라이브러리로 구현하였다. 움직임 추정과 움직임 보상 가속기에 대해 표 2에 정리하였다. 칩은 ITSOC MPW를 통해서 제작되었으며, 메모리가 제외된 움직임 추정 칩은 10K의 게이트 수와 150MHz의 속도로 동작한다.



(a) 움직임 추정 가속기 (b) 움직임 보상 가속기

그림 12. 움직임 추정 및 보상 하드웨어 가속기
 Fig. 12. Chips of proposed hardware accelerator for ME and MC.

표 2. 움직임 추정 및 보상 가속기

Table 2. Chip summary of proposed hardware accelerator for ME and MC.

	ME Hardware accelerator	MC hardware accelerator
Process technology	0.25um 1p4m	0.25um 1p4m
Logic gate count	40,000	10,000
Maximum frequency	100MHz	150MHz
On Chip Memory size	-	32Kb

표 3. 움직임 추정 가속기의 성능 비교

Table 3. Performance comparisons of the hardware ME architectures.

	Clock cycles / frame	Search range	Supported block size	Gate counts
Ref [18]	405,603	[-16,+15]	VBS	154K
Ref [19]	406,077	[-8, +7]	VBS	61K
Proposed	431,244	[-8, +7]	VBS	40K

기존 하드웨어 구조^[19-20]와 제안한 하드웨어와의 성능을 표 3에 비교하였다. 제안한 움직임 추정 가속기는 [19]과 [20]에 비해 필요한 연산 사이클은 비슷하면서도 제안한 하드웨어의 게이트 수가 매우 작다는 것을 알 수 있다.

V. 결 론

본 논문에서는 ASIP 기술을 이용해 실시간 H.264/AVC 복호화 가능한 VSIP을 제안하였다. VSIP은 H.264/AVC에 최적화된 하드웨어 구조와 전용 명령어를 가지고 있으며, 움직임 추정과 보상 그리고 엔트로피 코딩을 지원하기 위한 하드웨어 가속기를 가지고 있다. VSIP은 기존의 디지털 신호처리 프로세서보다 적은 연산 사이클이 필요하며, 메모리 접근 횟수를 감소시켜 전력 소비를 줄였다. 또한 VSIP과 하드웨어 가속기를 이용하여 실시간 비디오 신호처리를 할 수 있으며, 다양한 프로파일과 표준을 지원할 수 있다.

참 고 문 헌

- [1] Jae S. Lee, Young S. Jeon, and Myung H. Sunwoo, "Design of new DSP instructions and

their hardware architecture for high-speed FFT," in Proc. IEEE Workshop on Signal Processing Syst., pp. 80-90, Sept. 2001.

- [2] J. Glossner, J. Moreno, M. Moudgill, J. Derby, E. Hokenek, D. Meltzer, U. Shavadron, and M. Ware, "Trends in compilable DSP architecture," in Proc. IEEE Workshop on Signal Processing Syst., pp. 181-199, 2000.
- [3] Jeong H. Lee, Jong H. Moon, Kyung L. Heo, Myung H. Sunwoo, Sung K. Oh, and In H. Kim, "Implementation of Application Specific DSP for OFDM Systems," in Proc. IEEE Int. Symp. Circuit Syst., May 2004.
- [4] Suk Hyun Yoon, Jong Ha Moon, and Myung Hoon Sunwoo, "Efficient DSP Architecture for High-Quality Audio Algorithms," in Proc. IEEE Int. Symp. Circuits Syst., May 2005.
- [5] Sung Dae Kim, Jeong Hoo Lee, Chung Jin Hyun, and Myung Hoon Sunwoo, "ASIP approach for implementation of H.264/AVC," in Proc. Asia South Pacific Design Automation Conf., Jan 2006.
- [6] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 (E) AVC). July, 2004.
- [7] J. Ostermann, T. Wedi, et al., "Video coding with H.264/AVC: tools, performance, and complexity," IEEE Circuits and Systems Magazine, vol. 4, pp. 7-28, 2004.
- [8] Wu Di, Gao Wen, Hu Mingzeng and Ji Zhenzhou, "An Exp-Golomb encoder and decoder architecture for JVT/AVS," in Proc. 5th International Conference on ASIC, vol. 2, pp. 910-913, Oct. 2003.
- [9] Gisle Bjontgaard and Karl Lillcvold, "Context-adaptive VLC (CAVLC) coding of coefficients," Doc. JVT-028, JVT of ISO/IEC MPEG & ITU-T VCEG 3rd Meeting, Virginia, USA, May. 2002.
- [10] Hsiu-Cheng Chang, Chien-Chang Lin, and Jiun-In Guo, "A Novel Low-Cost High-Performance VLSI Architecture for MPEG-4 AVC/H.264 CAVLC Decoding," in Proc. IEEE Int. Symp. Circuits Syst., May 2005.
- [11] Yeong-Kang Lai, Chih-Chung Chou, and Yu-Chieh Chung, "A simple and cost effective video encoder with memory-reducing CAVLC," in Proc. IEEE Int. Symp. Circuits Syst., May 2005.
- [12] Woong IL Choi, Byeungwoo Jeon and Jechang Jeong, "Fast motion estimation with modified

diamond search for variable motion block sizes," in Proc. International Conference on Image Processing, vol. 3, pp. 14-17, Sept. 2003.

[13] TMS320C6000 CPU and Instruction Set Reference Guide, Texas Instruments Inc., Dallas, TX, 2000.

[14] TMS320C64x Image/Video Processing Library, Texas Instruments Inc., Dallas, TX, 2003.

[15] Blackfin™ DSP Instruction Set Reference, Analog Device Inc., Norwood, Mass. 2002.

[16] TMS320C55x Hardware Extensions for Image/Video Applications Programmer's Reference, Texas Instruments Inc., Dallas, TX, 2002.

[17] Thomas Wiegand, Xiaozheng Zhang, and Bernd Girod, "Long-Term Memory Motion-Compensated

Prediction," Trans. Circuit Syst. Video Technol., vol. 9, no. 1, pp. 70-84, Feb. 1999.

[18] Iain E. G. Richardson, Video Codec Design: Developing Image and Video Compression Systems, Wiley, 2002.

[19] Min H. Kim, In G. Hwang and Soo I. Chae, "A Fast VLSI Architecture for Full-Search Variable Block Size Motion Estimation in MPEG-4 AVC/H.264," in Proc. of Asia and South Pacific Design Automation Conference (ASP-DAC 2005), Shanghai, China, Jan 2005.

[20] Swee Yeow Yap and John V. McCanny, "A VLSI Architecture for Variable Block Size Video Motion Estimation," Trans. Circuit Syst. Video Technol., vol. 51, no. 7, July 2004.

— 저 자 소 개 —



김진수(정회원)
 2005년 아주대학교 전자공학부
 학사 졸업.
 2005년~현재 아주대학교
 전자공학과 석사 과정.
 <주관심분야 : 영상 처리, 화질 개
 선, 영상압축>



선우명훈(정회원)
 1980년 서강대학교 전자공학과
 학사 졸업.
 1982년 한국과학기술원 전자공학
 석사 졸업.
 1982년~1985년 한국전자통신
 연구소 (ETRI) 연구원.
 1985년~1990년 Univ. of Texas at Austin
 전자공학 박사.
 1990년~1992년 Motorola, DSP Chip Division
 (미국).
 1992년~1996년 아주대학교 전기전자공학부
 조교수.
 1996년~2001년 아주대학교 전자공학부 부교수.
 2001년~현재 아주대학교 전자공학부 교수.
 300편 이상의 논문, 37개 특허 출원 및
 등록. IEEE Workshop on Signal
 Processing Systems (SIPS) 2005 Best
 Student Paper Award 포함 20회 이상의
 논문상 수상. IEEE SIPS 2003 Technical
 Program Chair 역임.
 2007년 현재 IEEE SIPS, Cool Chips, Design
 Automation and Test in Europe
 (DATE), IEEE International ASIC/SOC
 Conference, Asian-Pacific Conference
 on CAS (APC-CAS), Asian-Solid State
 Circuits Conference (A-SSCC),
 International SOC Design Conference
 (ISOCC), International Symposium on
 VLSI Design, Automation and Test
 (VLSI-DAT) 등의 국제 학회 committee
 위원.
 <주관심분야 : VLSI 및 Parallel Architecture, 통
 신 멀티미디어용 DSP 칩 및 ASIC 설계>