ASIP 기술을 활용한 H.264/AVC 고속 병렬 복호화기 설계

*지봉일, *심동규, **김경수, **박성모

*광운대학교 컴퓨터공학과, **한국전자통신연구원

*jibongil@naver.com, *dgsim@kw.ac.kr, **kimks0326@etri.re.kr, **smpark@etri.re.kr

Design of High-speed H.264/AVC Parallel Decoder Using ASIP Approach

*Ji, Bong-Il *Sim, Dong-Gyu **Kim, Kyung-Su **Park, Seong-Mo

*Dept. Computer Engineering, Kwangwoon University, **Electronics and Telecommunications Research Institute

요약

본 논문에서는 고해상도 동영상의 실시간 복호화를 위하여 Application Specific Instruction-set Processor (ASIP)기술을 이용하여 H.264/AVC 고속 병렬 복호화기를 설계하였다. 우선, 하드웨어에 최적화된 구조로 복호화기를 설계하고 LISA로 기술한 멀티미디어 전용 명령어를 명령어 집합에 추가하였다. 이렇게 설계한 고속 H.264/AVC 복호화기는 사이클 기반 시뮬레이터에서 성능을 측정한 결과 기존 대비 약 35%의 복호화 사이클 감소를 보였다. 추가적인 성능 향상을 위해, 앞서 설계한 고속 복호화기를 여러 개 사용하여 병렬 H.264/AVC 복호화기를 설계하였다. 병렬 복호화기는 여러 매크로블록을 동시에 복호화 처리함으로써 복호화기의 성능을 대폭 향상시켰다. 병렬 복호화기는 고속 복호화기 대비 약 75%의 복호화 사이클이 감소하였다. 이에 고해상도 동영상의 실시간 복호화를 위한 H.264/AVC 고속 병렬 복호화기의 설계 방법을 제시하고자 한다.

1. 서론

최근 반도체 개발 기술인 ASIP (Application Specific Instruction-set Processor)는 ASIC (Application Specific Integrated Circuit)의 장점인 고속처리 능력과 저 전력 동작 그리고 DSP (Digital Signal Processor)의 재설계 유연성을 동시에 만족하는 설계 방법으로 중요성이 대두되고 있다. 예를 들어, ASIP 기술을 사용하면 기존에 개발된 시스템에 기능이 추가될 경우 하드웨어를 재설계해야 하는 ASIC과 달리 시스템의 코드를 수정하여 재사용함으로써 시장의 변화에 빠르게 대처할 수 있게 된다.

H.264/AVC는 ITU-T와 ISO/IEC의 JVT (Joint Video Team)가 2003년 5월에 발표한 최신 동영상 압축 표준으로 문맥기반 적응적 이진 산술 부호화 (Context Adaptive Binary Arithmetic Coding), 다양한 블록 크기의 움직임 예측, 복수 개의 참조 영상, 디블록킹 필터 등의 기술을 사용함으로써 기존 코덱 대비 우수한 압축 성능을 제공한다.[1] 반면 복잡도는 크게 증가하였다. 예를 들어, H.264/AVC 압축 표준은 MPEG-4 대비 부호화기는 두 배 이상의 압축 성능을 제공하지만 약 10배 이상의 복잡도가 증가하였고 복호화기는 하드웨어 복잡도로 약 두 배 이상의 계산 복잡도가 증가하였다.[4]

따라서 최근 IPTV (Internet Protocol Television)와 같은 상용 방송 시스템에서 서비스하는 HD(1280x720), Full HD(1920x1080)의 고해상도 영상을 실시간으로 복호화 하여 시청하기 위해서는 고속으로 동작하는 복호화기가 요구된다.

2. ASIP를 이용한 고속 H.264/AVC 복호화기 설계

가. ReferenceC 설계

ASIP 기술을 사용한 고속(High performance) H.264/AVC 병렬 복호화기를 설계하기 위하여 그림 1의 ASIP 개발 단계를 따랐다.[2]

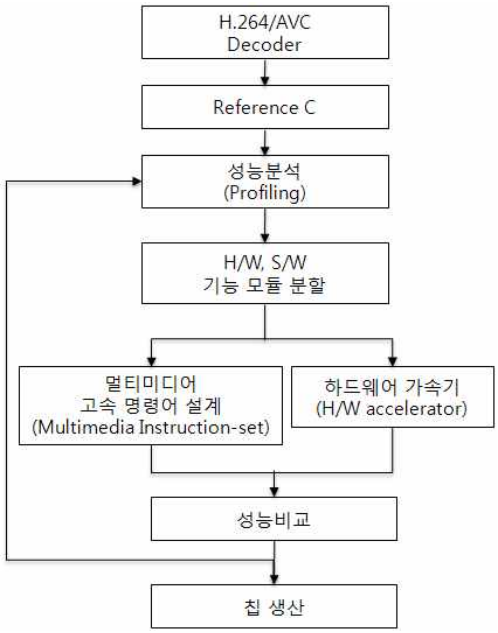

그림 1. ASIP 기술을 사용한 H.264/AVC 설계 과정

우선, ASIP 기술을 적용할 애플리케이션을 선택한다. 본 논문에서는 H.264/AVC High-profile 레퍼런스 소프트웨어인 JM (Joint Model)15.1 복호화기를 사용하여 설계를 진행하였다.

다음으로 ReferenceC 단계는 범용 PC (Personal Computer)환경에서 동작하도록 설계된 기존 복호화기 구조를 임베디드 하드웨어 플랫폼에서 효율적으로 동작할 수 있는 구조로 변경하는 것이다. 예를 들어, 동적 메모리 할당 명령어 제거, 구조체 변수 제거, 변수 데이터 폭(data width)최적화, 그리고 이후의 병렬화 설계를 고려하여 복호화 과정이 매크로블록 단위로 진행되도록 구조를 변경한다. 아래는 ReferenceC 단계를 거친 후 복호화기의 의사코드(Pseudocode)이다.

```
Header_Parsing( );              // SPS, PPS, SEI 파싱
SetParam( );                    // 복호 환경 파라미터 설정

While ( FramesToBeDecoded( ) != NULL ) {

// 매크로블럭 단위 복호화 루프
for( MbY = 0; MbY < HEIGHT; MbY+=16 ) {
    for ( MbX = 0; MbX < WIDTH; MbX += 16 ) {
        VLD_IQ ( );      // CAVLC, CABAC
                         // 역양자화
        IDCT( );         // 4x4, 8x8 단위 역변환
        IPRED( );        // 인트라 예측
        MEM_MC( );       // 움직임 보상
        MC( );           // 인터폴레이션
        RECON( );        // 화소 복원
        DB( );           // 디블록킹 필터링
    }
  }
  Update_ReferencePictureBuffer( ); // 참조 영상 갱신
}
```

위 의사코드에서 볼 수 있듯이, 제안하는 고속 H.264/AVC 복호화기의 복호화 루프는 매크로블록을 기본 단위(Unit)로 하여 수행되며 각 기능 모듈이 명확히 구분된 상태임을 알 수 있다. 이러한 프로그램 구조는 하드웨어 구현에 매우 유용할 수 있다. 특히, DMA (Direct Memory Access)를 이용할 경우, 여러 기능 모듈 중 선택적으로 DMA와 병렬로 동작시킬 수 있어 외부 메모리 사용에 따른 성능저하를 유연성 있게 줄일 수 있게 된다. 아래 그림 2는 모듈화 과정이 적용된 ReferenceC의 블록도이다.

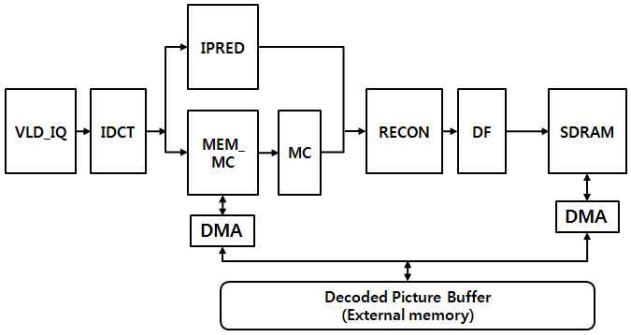

그림 2. ReferenceC 블록도

외부 메모리는 복호화된 화소 데이터를 저장하는 공간으로 매크로블록 단위로 복호화된 화소 데이타는 SDRAM 모듈에서 DMA를 사용하여 외부 메모리로 전송한다. 한편, 외부 메모리의 화소 데이터가 참조 데이터로 사용되는 경우는 마찬가지로 DMA를 통해 외부 메모리에서 MEM_MC 모듈로 전송 받는다.

나. 프로파일링을 통한 H/W 모듈 설계와 전용 명령어 설계

ReferenceC 다음 단계는 프로파일링(Profiling) 과정으로 각 기능(functional) 모듈별 소요 사이클 즉, 복잡도를 측정하고 복호화 실행 과정에서 호출된 명령어(Instruction)의 빈도수를 파악한다. 프로파일링은 Coware(社)의 Processor Designer에서 제공하는 사이클 기반 프로파일링 기능을 사용하여 측정하였다. 기능 모듈의 복잡도 통계는 H/W로 설계할 모듈을 결정하는데 사용된다. 아래 표 1은 ReferenceC의 프로파일링 결과이다. 표 1에서 확인할 수 있듯이, 비트스트림 파싱(parsing)을 하는 VLD_IQ 모듈의 복잡도가 높은 것을 알 수 있다. 또한 비트스트림 파싱 과정은 루프와 분기가 많이 발생하므로 S/W최적화로는 해결이 어렵다. 따라서 본 설계에서는 VLD_IQ 기능 모듈을 위한 전용 H/W를 사용하였다.

표 1. ReferenceC의 프로파일링 결과

모듈명	복잡도 (cycle)	비율(%)
VLD_IQ	18,840,095	26.77
IDCT	6,399,796	9.09
IPRED	2,810,044	3.99
MEM_MC	2,883,418	4.1
MC	10,796,383	15.34
RECON	4,210,272	5.98
DB	11,640,991	16.54
Etc. (dump)	10,312,415	18.19
Total	70,372,308	100

한편, 복호화 과정에서 호출된 명령어의 빈도수 측정 결과는 표 2와 같다.

표 2. 복호화 과정에서 호출된 명령어의 빈도수 측정 결과

명령어	호출 빈도수(call)	비율(%)
clip	301,412	66.5
memcpy	37,956	8.3
memset	21,864	4.8
etc.(abs, min)	91,828	20.4
Total	453,060	100

표 2에서 알 수 있듯이, clip 명령어가 가장 빈번히 호출이 된다. 따라서 이에 해당하는 전용 명령어를 LISA (Language for Instruction-Set Architecture)로 기술하여 추가하였다. 그 밖에 멀티미디어 애플리케이션에서 주로 사용 가능한 명령어도 함께 추가하였다. 아래 표 3은 추가된 멀티미디어 전용 명령어 목록이다.

표 3. 추가된 멀티미디어 전용 명령어

명령어	용도	포맷
lclip_ckf	클리핑	lclip_ckf(src, low, high)
labs_ckf	절댓값	labs_ckf(src1, src2)
lmin_ckf	최솟값	lmin_ckf(src1, src2)
lmax_ckf	최댓값	lmax_ckf(src1, src2)

그 외 라이브러리 함수를 사용함에 따른 사이클 소모가 큰 명령어는 S/W최적화 과정에서 SIMD (Single Instruction Multiple Data)형태로 직접 값을 복사 및 이동하도록 최적화 하였다. 아래 표 4는 멀티미디어 전용 명령어 추가와 최적화가 적용된 ReferenceC 즉, ASIP 기술을 이용하여 설계한 고속 H.264/AVC의 프로파일링 결과이다.

표 4. ASIP 기술을 사용한 고속 H.264/AVC의 프로파일링 결과

모듈명	복잡도 (cycle)	비율(%)
VLD_IQ	7,477,275	16.07
IDCT	1,963,619	4.22
IPRED	2,211,885	4.75
MEM_MC	2,760,127	5.93
MC	8,067,088	17.34
RECON	1,571,427	3.38
DB	9,853,032	21.18
Etc. (dump)	12,626,782	27.13
Total	46,531,235	100

표 4에서 알 수 있듯이, 멀티미디어 전용 명령어를 추가하고 소프트웨어 최적화한 고속 H.264/AVC 복호화기의 성능은 초기 복호화기 성능(표 1) 대비 동작 사이클이 약 33.8% 감소하였다.

3. 고속 H.264/AVC 병렬 복호화기 설계

앞서, ASIP 기술을 사용하여 설계한 고속 H.264/AVC 복호화기는 ReferenceC 대비 33.8%의 속도 향상이 있지만 HD(1280x720), Full HD (1920x1080)와 같은 고해상도의 영상을 실시간으로 복호화 하기에는 충분하지 않다. 따라서 본 논문에서는 앞서 설계한 고속 H.264/AVC 복호화기를 여러 개를 사용하여 구성한 다중 코어 플랫폼(Multiple Core Platform)을 Coware(社)의 Platform Architecture를 사용하여 설계하였다. 결과적으로, 병렬 복호화기는 동시에 여러 개의 매크로블록을 동시에 복호화함에 따라 복호화 사이클이 고속 H.264/AVC 복호화기(표 4) 대비 약 77.1% 감소하였다.

가. H.264/AVC 병렬 복호화

다중 코어 플랫폼 설계에 앞서, 결정해야 할 사항은 병렬화(Parallelism) 레벨이다. H.264/AVC는 GOP (Group Of Picture), 프레임, 매크로블록, 블록 레벨까지 여러 단계로 병렬화 가능하다.[3] 본 논문에서는 다양한 병렬화 레벨 중 메모리 사용량, 병렬화에 따른 성능 향상률을 고려해 볼 때 임베디드 시스템 환경에 적합한 형태인 매크로블록 레벨 병렬화 방법을 사용하였다.

더불어, 매크로블록 레벨 병렬 복호화기를 설계할 때는 주변 매크로블록과의 종속성(dependancy)을 고려해야 한다. H.264/AVC는 이미 복호화된 주변 블록의 정보를 적응적으로 참조하여 사용함으로써 기존 코덱에 비해 더 높은 성능을 얻는 구조이다.[1] 따라서, 현재 매크로블록을 복호화하기 위해서는 주변 블록의 정보가 필요하다. 이에 코어 간에 필요한 정보를 전달 받을 수 있는 공유 메모리 (Shared Memory)를 사용토록 설계하였고 한편 종속성이 만족되는 경우에만 매크로블록 복호화를 진행하도록 코어 간 동기화 확인 코드를 추가하였다. 아래 그림 2는 복호화시 필요한 주변 매크로블록의 참조 정보를 나타낸다.

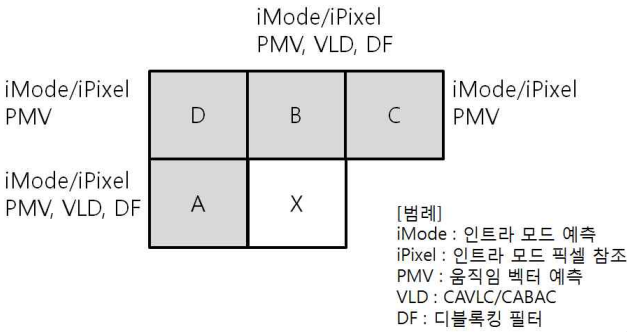

그림 2. 매크로블록 복호화시 참조되는 주변 블록의 정보

나. 다중 코어 플랫폼 설계

병렬 복호화기 설계를 위하여 앞서 설계한 고속 H.264/AVC 복호화기를 여러 개 사용하여 플랫폼을 구성하였다. 이때 사용한 설계 툴은 Coware(社)의 Platform Architecture를 이며 다중 코어 플랫폼의 설계 구조는 아래의 그림 3과 같다.

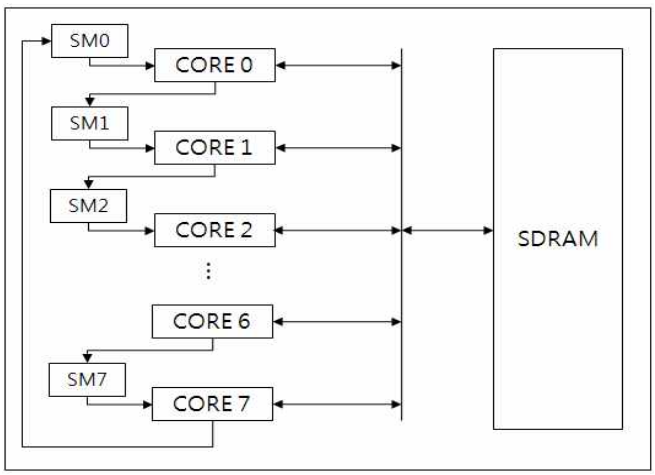

그림 3. 병렬 복호화를 위한 다중 코어 플랫폼 구조

그림 3에서 볼 수 있듯이, 여러 개의 매크로블록 복호화를 동시에 처리할 수 있도록 고속 H.264/AVC 복호화기 코어를 여러 개 사용하였으며, 매크로블록 복호에 필요한 주변 매크로블록의 정보를 참조할 수 있도록 코어 간에 공유메모리 (Shared Memory)를 사용하였다.

다. 다중 코어 플랫폼에서의 병렬 복호화기 성능 측정

성능 측정은 코어 두 개, 네 개, 그리고 여덟 개 코어로 구성된 플랫폼 상에서 진행하였으며, 앞서와 마찬가지로 Coware(社)의 Processor Designer에서 제공하는 사이클 기반 프로파일링 기능을 사용하여 측정하였다. 테스트에 사용된 영상의 크기는 QCIF (176x144)와 Full HD (1920x1080)이며 I(Intra), P(Predicive), B(Bi-Predicive) 프레임 조합으로 구성된 비트스트림 복호화에 소요된 총 사이클의 합이다. 아래 표 5, 표 6, 표 7은 각 각 코어 두 개, 네 개, 여덟 개 플랫폼 환경에서의 시뮬레이션 결과를 나타낸다. 성능 측정 결과에서 확인할 수 있듯이, 병렬 복호화에 따른 성능 향상은 매크로블록 단위 복호화 기능 모듈에서 나타난다.

표 5. 코어 두 개 플랫폼에서 병렬 복호화기의 프로파일링 결과

모듈명	복잡도 (cycle)	비율(%)
IDCT	1,142,482	7.72
IPRED	1,243,065	8.41
MEM_MC	1,705,780	11.54
MC	4,540,440	30.71
RECON	886,545	5.99
DB	5,262,384	35.60
Total	14,780,696	100

표 5에서 알 수 있듯이, 코어 두 개를 사용한 다중 코어 플랫폼에서의 복호화기 성능은 앞서 설계한 고속 H.264/AVC 복호화기(표 4) 대비 약 44.00%의 사이클이 감소하였다.

표 6. 코어 네 개 플랫폼에서 병렬 복호화기의 프로파일링 결과

모듈명	복잡도 (cycle)	비율(%)
IDCT	728,563	8.00
IPRED	760,003	8.34
MEM_MC	1,171,469	12.8
MC	2,802,615	30.78
RECON	531,927	5.84
DB	3,110,452	34.16
Total	9,105,029	100

표 6에서 알 수 있듯이, 코어 네 개로 구성한 다중 코어 플랫폼에서의 병렬 복호화기 성능은 고속 H.264/AVC 복호화기 대비(표 4) 약 65.5%의 사이클이 감소하였다.

표 7. 코어 여덟 개 플랫폼에서 병렬 복호화기의 프로파일링 결과

모듈명	복잡도 (cycle)	비율(%)
IDCT	520,724	8.60
IPRED	522,871	7.56
MEM_MC	894,485	14.78
MC	1,839,789	29.15
RECON	354,618	4.56
DB	2,146,478	35.31
Total	6,051,435	100

표 7에서 알 수 있듯이, 코어 8개를 사용한 다중 코어 플랫폼에서의 복호화기 성능은 고속 H.264/AVC 복호화기 대비 약 77.1%의 사이클이 감소하였다.

4. 결론

본 논문에서는 최근 IPTV에서 서비스하는 Full HD와 같은 고해상도 동영상을 실시간으로 복호화 하기 위한 고속 H.264/AVC 병렬 복호화기를 설계하였다. 우선, 하드웨어 구조에 적합한 H.264/AVC 복호화기를 설계한 후 ASIP 기술을 사용하여 복잡도가 높은 모듈은 H/W로 구현하고 빈도수가 높은 명령어는 LISA로 멀티미디어 전용 명령어를 설계하여 명령어 집합에 추가함으로써 복호화기를 고속화하였다. 성능 측정 결과, 고속 H.264/AVC 복호화기는 기존 복호화기 대비 약 33.8% 동작 사이클이 감소하였다. 추가 성능 향상을 위하여, 앞서 설계한 고속 복호화기를 여러 개 사용하여 구성한 다중 코어 플랫폼을 설계하고 여러 개의 매크로블록을 동시에 복호화 하였다. 여덟 개 코어로 구성된 병렬 복호화기의 성능을 측정한 결과, 고속 H.264/AVC 복호화기 대비 약 77%의 동작 사이클이 감소하였다.

참고문헌

[1] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, pp. 560-576, July 2003.

[2] S. D. Kim et.al, "ASIP approach for implementation of H.264/AVC," Journal of Signal Processing Systems, vol. 50, no. 1, pp. 53 - 67, Jan. 2008.

[3] C. H. Meenderinck, A. Azevedo, M. Alvarez, B. H. H. Juurlink, and A. Ramirez, "Parallel Scalability of H.264," in Proceedings of the first Workshop on Programmability Issues for Multi-Core Computers, 2008.

[4] J. Ostermann, T. Wedi, et al., BVideo coding with H.264/AVC: tools, performance, and complexity,IEEE Circuits and Systems Magazine, vol. 4, 2004, pp. 7 - 28.

감사의 글

본 연구는 지식경제부의 MPcore 플랫폼 기반 다중포맷멀티미디어 SoC사업의 일환으로 수행하였음.