
MPEG-4 부호화기를 위한 스트림 생성기 설계

송인근* · 서기범*

The design of Stream producer for MPEG-4 encoder

In-Keun Song* · Ki-Bum Suh*

요약

이 논문에서는 MPEG-4를 부호화하기 위한 스트림 생성기의 효율적인 구조 설계에 대하여 제안한다. 이 모듈은 DCT 및 양자화 모듈로부터 출력된 매크로 블록 단위의 양자화된 계수를 입력으로 하여 부호화기의 부호화 모드에 따라 VLC 부호화를 수행하며, MPEG-4 의 오류 은닉 방법을 지원하며, 데이터 파티션을 지원한다. 이구조는 VHDL 을 사용하여 설계하였으며, 포스트 시뮬레이션을 통해 설계된 부호화기의 성능을 평가하였다.

ABSTRACT

In this paper, we propose the architecture of stream producer for MPEG-4 Video encoding. This module receives the quantized coefficient from DCT and Quantization module in macroblock unit and performs the VLC coding according to the encoding mode, and supports the error concealment mode of MPEG-4 and data partitioning mode. Using the VHDL, we designed the module using this architecture and performed the evaluations of this module by performing the post-simulation

키워드

MPEG-4 Video, VLC, data partitioning, error concealment

1. 서론

최근, 동화상·정지화상·음성·텍스트 등의 각종 데이터를 통합적으로 취급하는 멀티미디어 시스템의 연구개발이 각 방면에서 활발하게 행해지고 있다. 이와 같은 각종 데이터 중에서도 특히 화상 데이터는, 사용자에게 주는 시각효과라고 하는 점에서 가장 중요한 정보원(source)의 하나이지만, 그 방대한 정보량을 효과적으로 축적 및 전송하기 위한 압축기술의 도입도 불가결하다.

이것은, 이 시기에 LSI 기술이나 컴퓨터 환경 등이 비약적으로 진보한 것도 한 요인이었지만,

가장 큰 공헌을 한 것은 국제표준화 움직임이다. 현재에는 예상되는 각종 application에 따른 국제 규격이 심의되어, 모든 방면에 실제로 사용하도록 제공되고 있다. 종래 전송 에러는 네트워크 등의 시스템 측에서 정정부호 등을 추가하는 것으로 대처하여 왔다. 이것은 종래의 표준화가 ISDN 이나 광 fiber, 혹은 위성통신 등의 비교적 전송에러가 적은 환경에서의 사용을 전제로하여 왔기 때문이다. 그러나 MPEG-4 에서는 PHS 등의 지상파나 인터넷도 포함한 글로벌한 네트워크에서의 사용도 고려하고 있기 때문에 에러 발생율은 종래 고려대상이었던 네트워크에 비해 105 - 107 배나 된

*우송대학교 컴퓨터전자정보공학부

접수일자 : 2003. 10. 14

다. 또 packet loss 와 같은 랜덤한 데이터 손실도 빈번하게 일어날 가능성이 있다. 그 때문에 시스템측에서 정정하지못했던 에러가 부호화부분으로 흘러들어 오게 되지만, 이 에러에 대처해야할, 부호화 자체가 오류의 영향을 최소한도로 막아내야 할 성질(오류내성)을 가지고 있도록 요구되고 있다.[1][2]

따라서 이 논문에서는 이러한 오류내성을 갖는 MPEG-4 의 에러 오류 은의 방법을 제공하는 MPEG-4 Encoder의 부호화기의 비트스트림을 생성하는 4가지 은의 방법을 처리하는 구조에 대하여 제안한다.

제안된 구조는 비트스트림을 원하는 구조를 선택하여 부호화 할 수 있도록 ARM 프로세서를 이용하여 부호화 방법을 지정할 수 있도록 AMB-A(Advancec Microcontroller Bus Architecture) 2.0 Spec 의 APB bus를 통하여 설계되었다. 설계된 Spec은 부호화 부분을 담당하는 VLC 부분과 실제 비트스트림을 분할해서 출력해주는 Stream producer부분으로 나누어 설계되며, II 절에서는 수용해야 할 MPEG-4 부호화 방식에 대하여 설명하고 III 절에는 제안된 구조에 대하여 설명한다. IV절에는 성능평가 및 검증방법에 대하여 설명하고, V 절에서 결론을 맺는다

II. MPEG-4 의 부호화 방식

MPEG-4 가 종래의 표준방식과 다른 점은, 임의형상의 물체나 영역(Object)을 개별적으로 부호화하는 것이 가능한 구조를 가지고 있는 것이다. 그림 1 에 MPEG-4 비트스트림의 계층구조를 나타낸다.

화상의 전체적인 시퀀스에 상당하는 계층은 VS(Video Session)라 불리고, 그 가운데는 1 개 이상의 VO(Video Object)가 존재한다. 예를 들어 배경 중에 인물이 존재하는 경우, 인물만의 연속 움직임을 하나의 VO 로 기술가능하고, 또 배경만의 시퀀스도 별개로 기술할 수 있다.

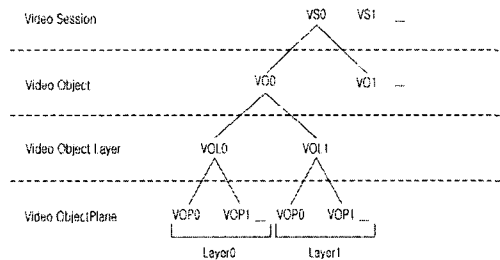


그림 1. MPEG4의 비트스트림의 계층 구조
Fig. 1 The hierarchy of MPEG-4 bitstream

나아가 각 VO 는 그 아래에 VOL (Video Object Layer)이라고 하는 계층을 가지고 있다. VOL 은 각각의 VO 에 대하여, 복수의 공간해상도 혹은 시간해상도를 제공 하기 위한 계층이고, 해상도변환을 동반하는 부호화 수법은 이 계층을 이용하여 표현할 수 있다. 최하위 계층인 VOP (Video Object Plane)는 각 VO 의 각각의 해상도에 있어서 "어느 순간"의 데이터(snap shot)를 의미한다. 종래의 표준방식에서는, 화상프레임을 1 부호화 단위로 하고 있었기 때문에, 각 VS 내에는 하나의 VO 만이 존재하고, 게다가 미리 설정된 해상도만으로 부호화하기 때문에, VOL 도 하나였다. 결국 종래의 데이터 구조는 MPEG-4 데이터 구조의 특수한 경우이고, VOP 가 그대로 프레임(혹은 필드)의 개념에 해당하는 것으로 된다.

종래 Codec(MPEG-1/MPEG-2) 들의 경우 전송 에러는 네트워크 등의 시스템 측에서 정정부호등을 추가하는 것으로 대처 하여 왔다. 이것은 종래의 표준화가 ISDN 이나 광 fiber, 혹은 위성통신 등의 비교적 전송에러가 적은 환경에서의 사용을 전제로 하여 왔기 때문이다. 그러나 MPEG-4 에서는 지상 파나 인터넷, 무선환경을 포함한 글로벌 한 네트워크에서의 사용도 고려하고 있기 때문에 에러 발생율은 종래 고려대상이었던 네트워크에 비해 105 ~ 107 배나 된다. 또 Packet Loss 와 같은 랜덤한 데이터 손실도 빈번하게 일어날 가능성이 있다. 그 때문에 시스템 측에서 정정하지 못했던 에러가 부호화 부분으로 흘러 들어 오게 되지만, 이 에러에 대처 해야 할, 부호화 자체가 오류의 영향을 최소한도로 막아내야 할 성질(오류내성)을 가지고 있도록 요구되고 있다. 이에

따라 MPEG-4에서는 다음과 같은 오류내성 부호화 방식을 처리하여야 한다.

1. Video Packet Resynchronization

압축된 비디오 데이터를 무선 채널을 통하여 전송할 경우, 채널 오류로 인해 비트열이 훼손되는 경우가 발생한다. 훼손된 비트열을 복호화할 경우 부호화기와의 동기가 달라져서, 현재 복호화하고 있는 데이터가 전체 프레임에서 정확히 어느 부분을 나타내는지 알 수 없게되어 이후 프레임의 화질이 급격히 떨어지게 된다. 급격한 화질 저하를 막기 위해 고안된 방법이 재동기화 코드를 삽입하는 것이다. MPEG-4는 기본적으로 H.263 모드를 포함하고 있지만, 재동기화의 위치는 다르다. 재동기화의 위치는 H.263의 경우에는 논리적인 단위(GOB : group of block)마다 삽입하는 방식을 채택하고 있고, MPEG-4의 경우에는 논리적인 단위 또는 일정한 길이마다 삽입(video packet)하고 있다. 데이터 보호 효율은 일정한 길이마다 삽입하는 방식이 우수하다. MPEG-4 부호화 방식은 H.263 mode(Short Video Header Mode)를 기본적으로 포함하기 때문에 하드웨어 설계시 두방식 모두 처리하여야 한다.

2. Data Partitioning (DP)

비디오 패킷 내에 검출되지 않은 오류가 있는 경우 다음 재동기화 위치에서 복호화가 정확히 끝나지 않는다. 이런 경우 현재의 비디오 패킷 전체를 버리고 다음 비디오 패킷의 재동기화 코드를 찾는 것이 보통이다. MPEG-4에서는 이러한 경우에도 비디오 패킷 내의 유효한 정보를 최대한 복원하기 위해 그림 2과 같이 Motion Boundary Marker(MBM)를 삽입하는 옵션을 제공한다.

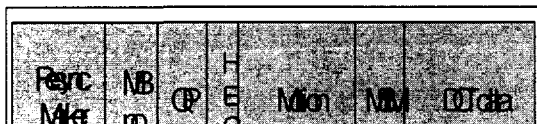


그림 2. 움직임 데이터와 DCT 데이터의 데이터 분할
Fig. 2 Data partition for Motion and DCT data

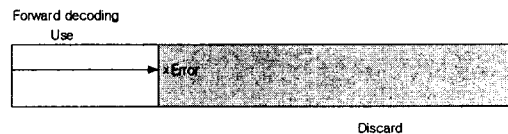
비디오 패킷 오류가 발생하고 MBM이 정확히 검출되었다면, 오류는 DCT 데이터부분에서 발생한 것이므로 움직임 데이터는 버리지 않고 복호화 과정에 사용한다.

3. Reversible VLCs (RVLCs)

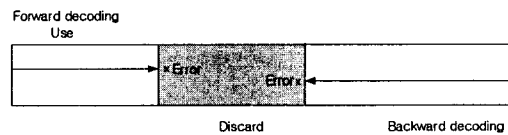
일반적인 VLC를 사용할 경우 한 번 오류가 발생하면 재동기화가 될 때까지 중간의 모든 데이터를 버려야 한다(그림 3(a) 참조). Reversible Variable-Length Codes(RVLC)를 사용할 경우 다음 재동기화 위치에서부터 역방향으로 코드를 읽어낼 수 있으며 순방향 복호화에서 발생한 오류와 역방향 복호화에서 발생한 오류 사이의 데이터만을 버린다(그림 3(b)참조). 따라서 Backward Decoding을 사용하여 에러 난 위치까지 다시 복호화를 진행하므로써 에러를 은닉시킬 수 있다. 이에 따라 MPEG-4에서는 RVLC가 아닌 종래의 부호화 table을 B.14의 표로 RVLC 모드의 경우에는 B.23의 표로 다른 부호화 table을 제공한다.[2] 따라서 부호화 시에는 부호화기의 제어 모드에 따라 선택적으로 부호화를 진행하게 한다.

4. Header extension code (HEC)

헤더 데이터에는 나머지 비트스트림을 해석하는데 매우 중요한 정보가 포함되어 있다.



(a) 종래의 VLC에 의한 복호



(b) 쌍방향으로부터 복호가능한 RVLC에 의한 복호

그림 3. 종래의 VLC와 RVLC의 비교
Fig. 3 The comparison of previous VLC and RVLC

이러한 데이터에는 복호화 할 영상의 공간적인 크기, 타임스탬프, 부호화 모드(INTRA 또는IN-TER) 등이 있다. 이들 데이터가 채널 오류에 의해서 훼손되었을 경우 현재 프레임에 속해있는 모든 정보가 쓸모 없게 된다.

이를 방지하기 위해서 MPEG-4 표준에서는 Header Extension Code(HEC)를 도입하였다. 그림 2에서 볼수 있듯이 각 비디오 패킷에는 1 비트 길이의 HEC 비트가 있으며, 헤더 정보가 해당 비디오 패킷에 포함되어 있는지를 나타낸다. 비디오 프레임의 헤더 정보가 훼손된 경우라도, 이후에 복호화한 비디오 패킷에 포함된 정보를 이용하여 헤더를 복원할 수 있다.

III. 제안된 구조

제안된 구조는 MPEG-4 의 비트스트림을 생성하는 모듈은 비트스트림을 생성하는데 있어, Data Partition을 수행하는지 아닌지에 따라 데이터의 처리 방법이 다르기 때문에 각각의 계수의 비트스트림을 생성을 따로 하여, 추후 merging을 수행하는 방법을 취한다. 따라서 비트스트림을 생성하는 모듈을 TVLC(Texture Variable Length Coding) 블록으로 설계하고 비트스트림을 merging 을 담당하는 모듈을 SP(stream producer)블록으로 따로 나누어 설계한다. 부호화 모드에서의 TVLC및 SP 모듈의 기능은 그림 4에 보이는 것과 같이 ARM 프로세서의 입력을 AMBA APB (advanced peripheral bus)를 통하여 데이터를 받고 그 정보에 따라, 부호화를 수행하게 된다. DCT와 양자화를 한 계수값을 SCAN order 에 따라 Texture Coefficient Buffer(Tcoeff_Buf)에 저장시키며, 그값을 6개의 블록 위치에 저장한다. 이 저장이 끝난 후 TVLC 블록은 ARM CPU로부터 부호화를 수행하라는 명령을 받고, Tcoeff_Buf로부터 데이터를 읽어서 그것에 맞는 비트스트림을 발생시켜 비트스트림 버퍼에 전달한다. 이 비트스트림을 Data Partition이 된 경우에는 정보를 모아서 따로 모아 부호화 하여야 하므로, Partition Buffer에 따로 정보를 모아 저장한다. TVLC 모듈은

Texture VLC 를 수행할 때 RVLC또는 normal VLC 의 여부와 intra/non intra 여부 coded_block pattern 값을 ARM 프로세서로부터 받아 비트스트림을 Bitstream Buffer에 저장한다. 저장되는 포맷은 추후 스트림 프로듀서가 분할정보를 잘 처리하도록 블록별로 정리되어 있다. Stream producer모듈은 비트스트림 버퍼로부터 데이터를 받으며, ARM 프로세서로부터 처리할 SP mode(VOP HEADER 제작, MB 제작, MB Stuffing 동작)에 따라 제작을 처리하도록 되어 있다.

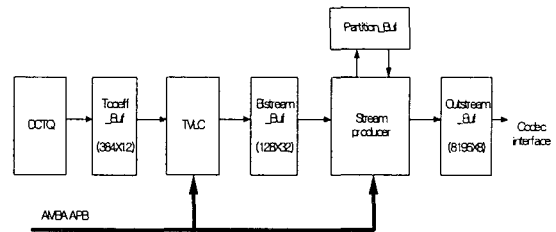


그림 4. SP 및 TVLC 블록의 MPEG-4 Encoder에서의 기능

Fig. 4 The function of TVLC and SP in MPEG-4 encoder

1.TVLC(Texture Variable Length Coding)모듈

TVLC 블록에서는 12bit x 384 word RAM 에서 DC Prediction 및 DCT/Q를 수행한 결과로 얻어지는 하나의 매크로블록(8x8 휘도블록 4개 + 8x8 색도블록 2개)을 입력으로 하여 이를 MPEG-4 syntax를 따르는 VLC code로 정해진 포맷에 맞춰 메모리에 출력하며 총 소요 clock은 최대 776clock/MB이다.

TVLC는Run Length Coding 기능, ESC Type Decision 기능, Codeword Generation 기능, 패킹, 포맷팅기능을 수행하며 TVLC의 top 블록도는 그림5 와 같다.

APB_if 블록은 부호화 할 Tcoeff Memory에 있는 영상데이터를 처리하기 위한 레지스터의 내용(mb_type, intra여부, short_video header, rvlc 여부)을 전달하여 주는 블록이며 RLC 블록은 APB_if 의 tvlc_start 신호에 의하여 내부의

Address counter가 초기화되고 입력 메모리의 데이터를 액세스 하여 Run Length coding 을 수행한다. 입력 메모리는 12bit x 384 크기로 한 번의 start 신호마다 1개의 매크로블록(6개의 8x8 블록:4개의 luminance 블록과 2개의 chrominance 블록)의 계수를 Run Length Coding한다. ESC Type Decision 블록은 RLC 블록의 출력을 이용하여 normal VLC Code로 부호화할 것인지 아니면 ESC 코드로 부호화 할 것인지를 결정한다. MPEG-4의 경우, Normal VLC Code Table의 입력에 해당하는 코드가 존재하지 않을 때, level - LMAX에 대한 코드, run - RMAX에 대한 코드의 존재 여부를 차례로 확인함으로써 최악의 경우 3번의 Code Table Access가 필요하다. 하지만 이

ble을 내장하고 있으며 ESC 코드에 대한 처리도 포함한다. packer block 블록은 입력된 비트열을 32비트 단위의 워드로 변환한 후 버퍼로 출력하는 블록이다. 단, DC 계수의 경우는 예외적으로 32비트가 되지 않더라도 출력해야 한다. 왜냐 하면, DC 계수는 AC 계수와 분리하여 출력 버퍼에 쓰고 있다. Output controller block은 packer 블록에서 만들어진 32비트 워드를 출력 버퍼의 해당 포맷에 맞도록 만들고 해당 주소에 써주는 역할을 수행한다. 비트스트림버퍼의 출력 포맷은 표 1과 같다.

Packer 블록에서 만들어진 32 비트데이터는 표 1과 같은 형태로 출력 버퍼에 쓰여진다. 버퍼의 주소 0부터 5까지의 6 개의 워드에는 6개의 블록에 대한 VLC 코드가 저장된 출력 버퍼 주소가 저장된다. AC 코드의 주소는 하위 16비트에 DC 코드의 주소는 상위 16비트에 쓰여진다. DC 코드의 주소에 해당하는 위치에는 상위 16비트에는 DC CODE SIZE, 하위 16비트에는 실제 DC CODE가 저장된다. 여기서 저장되는 코드는 하위 비트에 맞춰 정렬되어 있다. AC 코드의 주소에 해당하는 위치에는 AC 코드의 총 길이, 그 다음 위치부터 실제 코드가 저장된다.

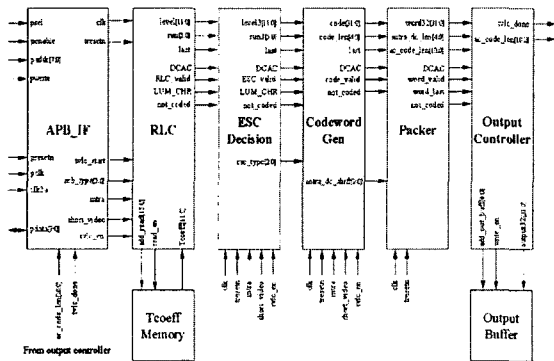


그림 5. TVLC 블록도
Fig. 5 The block diagram of TVLC

러한 순차적인 Code Table Access는 VLC의 동작 속도를 떨어뜨리므로 본 구조에서는 우선 ESC Type을 한번에 결정하고 이를 출력함으로써 Code Generation 블록은 이 결과를 이용하여 한번만 Code Table에 access함으로써 동작 시간을 단축시킨다. 즉, ESC Type Decision 모듈은 각각의 ESC Type에 대한 코드가 존재하는지를 병렬적인 구조로 확인하여 3번의 루프를 도는 위와 같은 문제는 발생하지 않는다. ESC Type Decision 모듈은 크게 NON-RVLC ESC Selection 블록, RVLC ESCdecision 블록, ESC correction 블록, RLD (run level decision) 모듈로 구성되어 있다. Codeword Generation Block의 블록은 VLC Ta-

2. SP 모듈

SP 기능모듈은 APB 와의 인터페이스 모듈, FSM 및 비트 스트림 제작 모듈, 데이터 Partition 모듈 및 Packing 모듈로 이루어져 있으며 기능 블록도는 그림 6과 같다.

SP 모듈은 Header 제작 및 MB(Macro Block)에 대한 비트 스트림을 생성하는 기능을 한다. 우선 VOP Header (Short Video Header) 비트스트림을 제작한 후에 MB에 대한 비트스트림을 출력 버퍼에 쓰는 동작은 다음과 같다. ARM 프로세서가 움직임 벡터를 포함하는 부호화에 필요한 부호화 정보를 APB I/F 레지스터에 써준 후 시작 신호를 발생시키면 SP는 Register의 내용과 TVLC의 출력 버퍼 (TVLC_BUF)의 DC 스트림과 AC 스트림을 이용하여 비트스트림을 만든다.

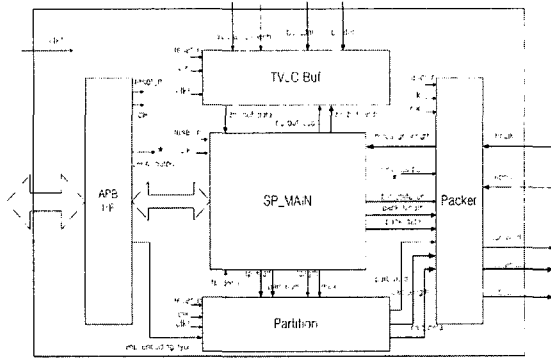


그림 6. SP 모듈의 기능 블록도
Fig. 6 The functional block diagram of SP module

Data Partitioning이 아닌 경우에는 MB에 대한 비트스트림을 만들어 출력버퍼(OS Buffer)에 저장하며 Data Partitioning인 경우에는 우선 각 Part 별 패킷 버퍼에 저장되어 있는 비트스트림의 길이와 현재 MB의 비트스트림 길이를 더한 값이 미리 설정한 최대 패킷 길이를 넘어서면 각 Part

별 패킷 버퍼의 비트스트림을 순서(Part1 Stream + marker + Part 2 Stream + Part 3 Stream)에 맞게 재구성하여 출력 버퍼에 저장한다. 비트스트림 길이가 최대 패킷 길이보다 작을 경우 MB의 비트스트림을 각 Part별 패킷 버퍼에 채워나간다. (단 Part1 Stream은 내부 버퍼에 저장할 필요 없이 출력 버퍼(OS Buffer)에 바로 저장하도록 구현한다.)

SP의 입력버퍼는 TVLC의 출력버퍼이며 이 버퍼에는 하나의 MB에 해당하는 6개의 블록에 대한 DC와 AC 스트림이 각각 다른 위치에 저장되어 있다. 시작 위치에는 각 스트림의 길이가 저장되어 있어야 하며 이후 번지부터 실제 스트림 데이터가 저장된다. 세번째 동작인 Stuffing MB 생성하여 Rate Control 에 의해 설정된 stuffing 비트를 추가하기 위해 stuffing MB 를 제작한다.

PARTITION 기능 블록에서는 두 개의 Partition Buffer를 가지고 있으며 두 개의 packer가 있어 8비트로 데이터를 packing 하여 버퍼에 저장

표. 1 출력버퍼의 포맷
Table. 1 The format of output buffer

Addr	31 30 29 17 16	15 14 13 2 1 0
0x00	DC ADDRESS (block 0)-DA0	AC ADDRESS (block 0)-AA0
0x01	DC ADDRESS (block 1)-DA1	AC ADDRESS (block 1)-AA1
0x02	DC ADDRESS (block 2)-DA2	AC ADDRESS (block 2)-AA2
0x03	DC ADDRESS (block 3)-DA3	AC ADDRESS (block 3)-AA3
0x04	DC ADDRESS (block 4)-DA4	AC ADDRESS (block 4)-AA4
0x05	DC ADDRESS (block 5)-DA5	AC ADDRESS (block 5)-AA5
DA0	DC CODE SIZE (block 0)	DC CODE (block 0)
AA0	AC CODE SIZE (block 0)	
AA0+1	AC CODE (block 0)	
...	...	
DA1	DC CODE SIZE (block 1)	DC CODE (block 1)
AA1	AC CODE SIZE (block 1)	
AA1+1	AC CODE (block 1)	
...	...	
DA5	DC CODE SIZE (block 5)	DC CODE (block 5)
AA5	AC CODE SIZE (block 5)	
AA5+1	AC CODE (block 5)	
...	...	

한다. Stuffing 이 끝나고 나면 SP_MAIN 모듈로 부터 fp_start 신호가 입력되고, partition1 buffer 와 partiotion2 버퍼의 내용들이 차례로 SP_PACKER 모듈로 출력된다.

SP_PACKER 기능블록은 SP_MAIN 모듈에서 생성한 스트림이나, data patition 모드의 경우 PARTITION 모듈에서 출력하는 데이터를 8비트 단위로 packing하여 OS_BUF에 저장한다. 데이터의 출력은 parallel과 serial mode에 따라 달라지며 nBORE, BOCLK 신호들에 의해 출력된다.

SP_MAIN 기능블록은 스트림제작 부이며 비트 스트림을 제작하여 Partition이나 Packer부로 출력시킨다. 이때의 제어 모드는 그림 7과 같이 제어된다. 여기에서 각 state 는 VS(Video Session 제작 부), VOL(Video Object Layer 제작 sequence), VOP_SH(VOP short header), VOP (Video object plane 제작)하는 Video Header부 제작 mode 와 MB(macro block)제어 모드, MB_Stuffing(zero로 된 비트를 padding 하는 모드) 모드로 크게 나눈다.

MB(Macro Block) 생성 모드에는 Data Partition Mode가 아닐 때 사용되는 CS(Create Stream) 모드가 있으며, PH 는 Packet Header 생성, PH_SH 는 H.263 mode 일 경우의 Packet Header 생성, IV_PART는 I-VOP인 경우에 data partition 이 사용되는 경우의 처리, PV_PART 는 P-VOP 인 경우에 data partition 이 사용되는 경우의 처리를 수행하고 있다. 또한 FP(Flush Partition) mode에서는 Partitioning 된 data (part2_buf, par3_buf의 내용)를 OS_BUF로 전달하는 모드이다.

IV.합성 결과 및 시뮬레이션

제안된 설계의 검증을 위하여, VHDL 을 이용하여 설계를 수행하였고, Reference C 를 사용하여 Test Vector의 생성한 후 이 벡터를 가지고 시뮬레이션을 수행하였다. 또한 Synopsys Design Compiler Tool을 사용하여 Hynix 0.35 μ m TLM 공정의 cb35os142d library로 합성한 후, 그 netlist

를 가지고 pre-simulation 을 수행하였다.

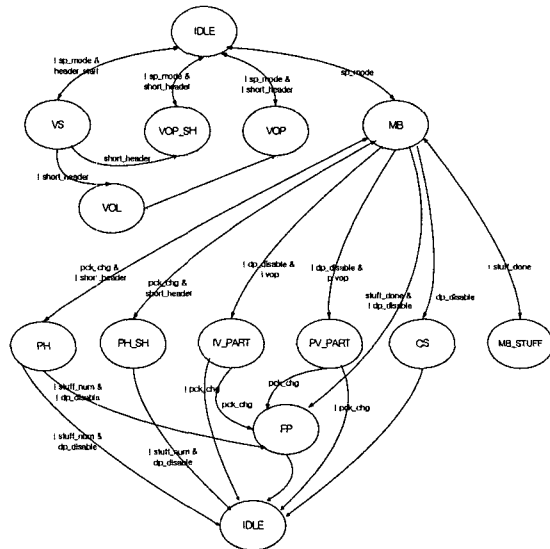


그림.7 스트림 생성기 모듈의 제어기
Fig. 7 State machine for stream producer

그림 8은 최종 생성된 비트스트림이 chip 의 출력단을 통해서 나오는 결과를 보인다. 최종 출력은 nbore2 라는 신호가 nega-edge 로 들어갈 때, 출력되며, bout 단자로 출력된다.

화면에 출력된 결과는 0x000001B0 (visual_object_sequence_start_code)와 0x000001B5 (visual_object_start_code)의 순으로 데이터가 출력됨을 알 수 있다. 그리고 synopsys design compiler로부터 출력 합성된 모듈의 크기를 표 2 에 정리하였다. 또한 그림9와 같이 TVLC의 자체의 수행 시간은 776 clock(54MHz)에 한 매크로블록을 처리할수 있어 최대 초당 69657 매크로블록을 처리 가능하다(CIF를 30frame/sec 으로 처리시 11520 매크로블록의 동작이 초당 필요). 따라서 이 부호화기의 성능은 CIF 급의 영상뿐만 아니라 4CIF 급의 영상도 처리 가능하다.

수행된 설계는 MOVA[4,5] 에 Porting 되어 MPEG-4 Codec의 서브 모듈로서 사용되었고, chip이 실시간으로 동작하였다.

표 2. 합성된 모듈의 크기
Table. 2 The size of synthesized module

모듈	게이트수	필요 RAM
TVLC	15566	TCOEF_BUF:384x12
SP	10804	BS_BUF : 128 x 32 Part2_buf: 512 x 8 Part3_BUF: 512 x 8 OS_BUF : 2048 x 8x 4
총합	26370 gate	총 82432 bit

V. 결론

이 논문에서는 MPEG-4를 부호화하기 위한 스트림 생성기의 효율적인 구조 설계에 대하여 제안하였다. 제안된 구조는 MPEG-4의 오류 은닉방법을 처리하기 위한 데이터 파티션 방법과 RVLC, Resync Marker, HEC를 지원하며, 최대 54 MHz의 클럭에서 동작이 가능하다. 이 구조는 VHDL을 사용하여 설계하였으며, 포스트 시

뮬레이션을 통해 설계된 부호화기의 성능을 평가하였다.

참고 문헌

- [1] N. Davies, J. Finney, A. Scott, "Supporting Adaptive Video Applications in Mobile Environments," IEEE Communications Magazine, vol. 36, no. 6, pp. 138-143, June 1998.
- [2] ISO/IEC 14496-2 Final Draft of International Standard
- [3] S. M. Park and et al., "A Single-Chip Video/Audio Codec for Low Bit Rate Application", ETRI J., pp. 20-29 vol. 22, no. 1, Mar. 2000.
- [4] Seongmo Park, and et al., "A MPEG-4 Video Codec Chip with Low Power Scheme for Mobile Application, ITC-CSC-C2002, July, 2002
- [5] Seong-Min Kim, and et al., "Hardware-Software Impelmentation of MPEG-4 Video Codec", ETRI Journal, vol25,no.6, pp. 489~502, Dec. 2003

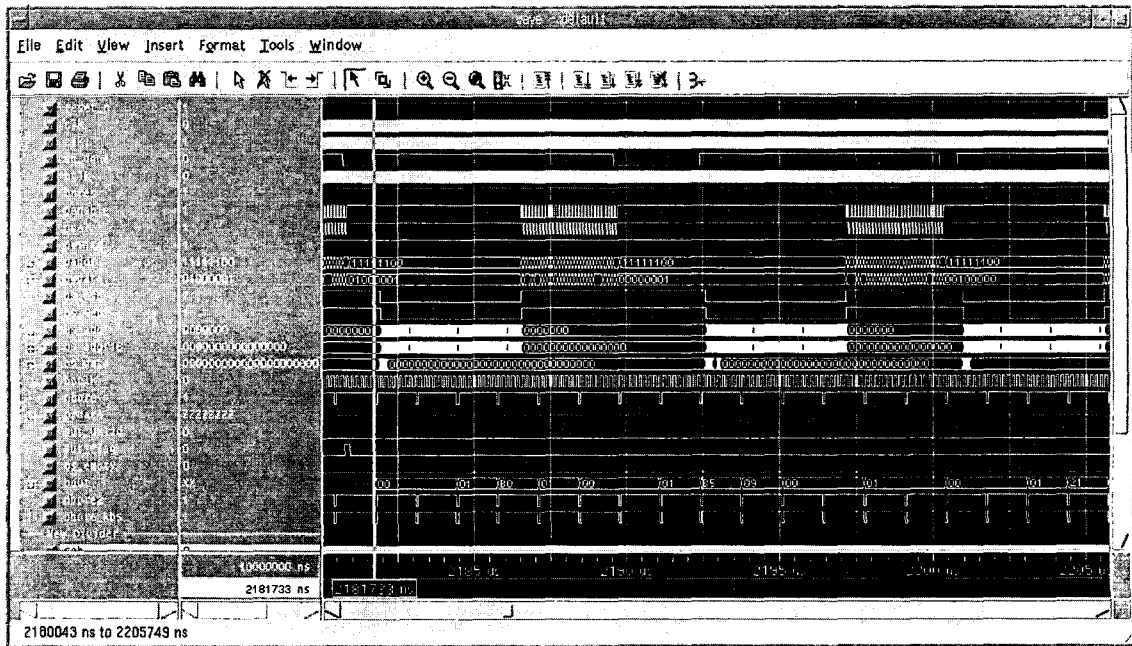


그림 8. 시뮬레이션 결과
Fig. 8 Simulation Result

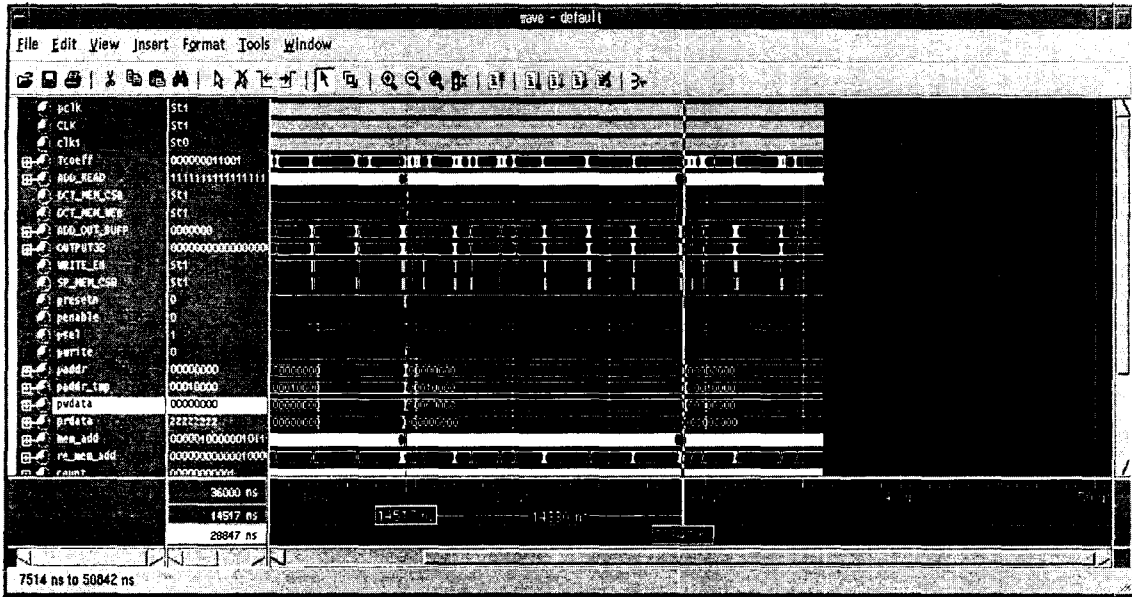


그림 9. 매크로블록당 처리시간
 Fig. 9. The processing time for one macroblock

저자 소개



송인근(In-Keun Song)

1978년 고려대학교 전자공학과 학사
 1983년 고려대학교 전자공학과 석사
 1984년 3월~1995년 2월: 한국전자통신연구원
 신연구원

1999년 명지대학교 전자공학과 박사
 1995년~현재우송대학교 컴퓨터전자정보공학부 부교수
 ※관심분야 : 이동 통신, 컴퓨터통신



서기범 (Ki-bum Suh)

1989년 한양대학교 전자공학과 졸업(공학사)
 1991년 한양대학교 전자공학과 졸업(공학석사)

2000년 한양대학교 전자공학과 졸업(공학박사)
 2000년 ~ 2002년 한국전자통신연구원 선임연구원
 2002년 ~ 현재 우송대학교 컴퓨터 전자정보공학부
 전임강사
 ※ 관심분야 : 영상통신 코덱, ASIC