VHDL 을 이용한 H.263+ Codec 의 설계

김은성, 김상철
한국외국어대학교 컴퓨터공학과

A Design of H.263+ Codec using VHDL

Eunsung Kim, Sangchul Kim
Dept. of Computer Science & Engineering, Hankuk University of Foreign Studies

요 약

H.263+[1]는 1996 년에 비디오압축 표준안으로 확정된 H.263 의 확장으로 현재 널리 퍼져 있는 저 전송률의 가정용 전화선을 위한 비디오 압축에 관한 표준안으로 1998 년에 확정되었다. H.263+의 기본 알고리즘은 H.263 과 같으나 사용상의 편의를 위한 여러 선택모드와 이종의 네트워크환경에서 발생하는 대역폭의 변화나 에러를 위한 scalability(계층부호화)와 같은 새로운 기능이 추가되었다. 새로운 표준안에 따른 실시간 비디오 전송을 처리하기 위해서 필연적으로 하드웨어 코덱의 개발 필요성이 대두되고 있으며 실시간 비디오 코덱은 영상회의 전화기나 멀티미디어 전송시스템의 핵심기술로써 적용될 수 있다. 본 논문에서는 ITU-T H.263+의 기본모드와 새로이 추가된 선택모드 일부를 지원하는 코덱을 하드웨어 기술 언어인 VHDL(VHSIC Hardware Description Language)을 사용하여 기술하고 각 모듈과 최상위 모듈을 시뮬레이션하여 동작을 검증했다.

1. 서론

최근 멀티미디어 데이터통신이 급격히 증가하는 추세에 따라서 멀티미디어 영상 전달이 더욱 많아지고 있다. 이러한 영상 정보는 크기가 매우 커서 압축을 하지 않고는 실시간에 전달하기는 거의 불가능하다.

H.263+는 1996 년에 비디오압축 표준안으로 확정된 H.263 의 확장으로 현재 널리 퍼져 있는 저 전송률의 가정용 전화선을 위한 비디오 압축에 관한 표준안으로 1998 년에 확정되었다. H.263+의 기본 알고리즘은 H.263 과 같으나 사용상의 편의를 위한 여러 선택모드와 이종의 네트워크환경에서 발생하는 대역폭의 변화나 에러를 위한 scalability(계층부호화)와 같은 새로운 기능이 추가되었다. 새로운 표준안에 따른 실시간 비디오 전송을 처리하기 위해서 필연적으로 하드웨어 코덱의 개발 필요성이 대두되고 있으며 실시간 비디오 코덱은 영상회의 전화기나 멀티미디어 전송시스템의 핵심기술로써 적용될 수 있다. 우리의 조사에 따르면 H.263+ 소프트웨어 코덱은 UBC(Univ. British Columbia)등에서 개발되었으나 현재까지 H.263+를 지원하는 하드웨어 코덱은 나와있지 않다.

본 논문에서는 ITU-T H.263+의 기본모드와 새로이 추가된 선택모드 일부를 지원하는 코덱을 하드웨어 기술 언어인 VHDL을 사용하여 DCT(discrete Cosine Transformer), IDCT(Inverse DCT), Q(Quantizer), IQ, ME(Motion Estimator) 및 기타모듈과 최상위 모듈을 기술하고 시뮬레이션하여 동작을 검증한다.

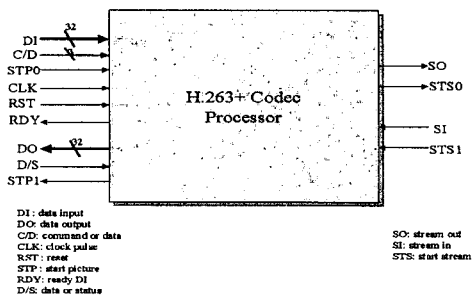
논문의 구성은 다음과 같다. 2 장에서는 설계를 위한 상위모듈의 정의, 상위모듈의 입출력 인터페이스, 각 명령어의 정의에 대해서 기술한다. 3 장에서는 코덱의 전반적인 동작을 기술하고, 4 장에서는 설계환경과 시뮬레이션에 대해서 기술하고, 마지막으로 5 장에서는 결론 및 향후 연구를 제시한다.

2. H.263+ 코덱의 설계

2.1 설계된 상위모듈

본 논문에서 설계된 코덱에서는 비디오 신호를 디지털신호로 변환하는 A/D 나 디지털신호를 비디오 신호로 변환 하는 D/A 를 처리하는 Preprocessor 나 Postprocessor 를 편의상 제외했다.

설계된 상위모듈은 <그림 1>과 같다. 설계된 상위모듈을 살펴보면 부호기/복호기의 입출력으로 32bits 의 데이터입출력 DI/DO, DI 가 명령인지 데이터인지를 선택하는 C/D, 입력데이터의 시작을 알리는 STP0, 다음 프레임을 입력받을 준비가 되었다는 상태를 알리는 RDY, DO 의 내용이 데이터인지 내부상태인지를 선택하는 D/S, 복호기의 출력신호를 알리는 STP1, 코덱의 bit-stream 출력인 SO 와 bit-stream 의 시작을 알리는 STS0, 복호기의 bit-stream 입력인 SI 와 시작을 알리는 STS1 으로 설계하였다.



< 그림 1> H.263+ 코덱의 최상위 모듈 구성도.

2.2 C/D pin 의 정의

H.263+ 코덱의 C/D 핀은 DI 의 내용이 프레임 데이터인지 코덱의 동작 파라미터들을 전달하는 명령어인지를 구분하는 기능을 수행한다. 명령의 내용은 다음 <표 1>과 같이 설계하였다.

<표 1> C/D pin 정의

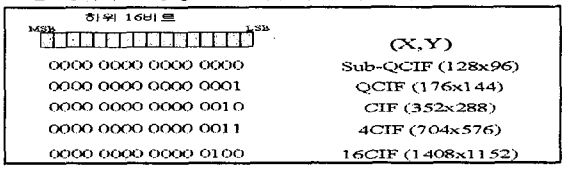
C/D	LH
000	Data
001	Picture_Size
010	User_Define_Size
011	Optional_Mode
100	Additional_Information
101	Not defined yet
111	Not defined yet

2.3 명령어의 정의

C/D 의 값이 000 이면 DI 의 내용이 데이터를 나타내고 001 에서 111 까지는 DI 의 내용이 명령어임을 나타낸다. 본 장에서는 각 명령어 별로 DI 의 내용을 간략히 기술한다.

2.3.1 Picture_ Size 명령어

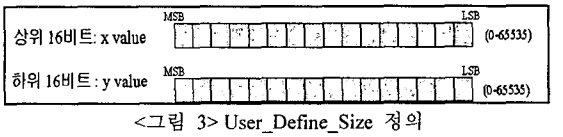
Picture_Size 은 H.263 이 지원하는 Sub-QCIF, QCIF, CIF, 4CIF, 16CIF 를 선택하는 명령으로 다음과 같이 설계하였다.



<그림 2> Picture_Size 정의

2.3.2 User_Define_Size 명령어

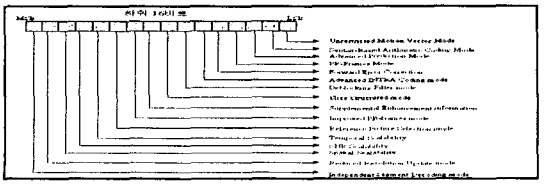
User_Define_Size 은 custom format 으로 위 Picture_Size 이외의 크기를 지정하는 명령어로 가로세로의 크기로 입력하도록 설계되었다. 화면 크기가 변화될 때 사용되는 것으로 다음과 같이 정의하였다.



<그림 3> User_Define_Size 정의

2.4.3 Optional_Mode 명령어

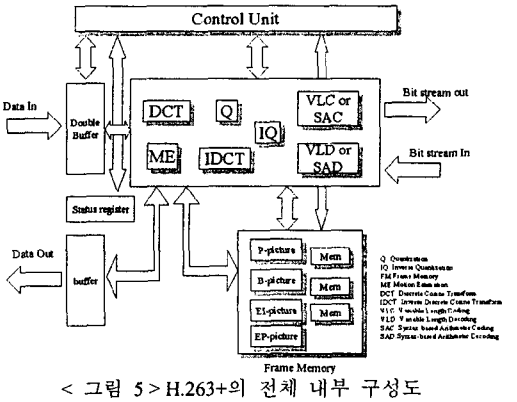
Optional_Mode 은 ITU-T H.263+에 정의된 optional 모드들의 실행여부를 선택하는 기능을 담당한다. . <그림 4>는 각 모드에 할당된 비트를 나타내고 있다. 각 비트의 입력 값이 '1'이면 해당 모드의 선택을 입력 값이 '0' 이면 해당 모드를 사용하지 않는 것을 의미한다. 각 모드는 중복적으로 선택될 수 있으며 선택 모드에 따라서 부가 적인 정보의 입력은 Additional_Information 명령어를 이용하여 입력 받도록 설계하였다



<그림 4> Optional_Mode 의 정의

2.4 H.263+코덱 전체 구성도.

비디오 입력은 프레임별로 입력되고 각 프레임은 DI 를 통해서 32 비트단위로 입력된다. 한 프레임의 입력과 그전 프레임의 코딩을 동시에 처리하기 위해 아래의 "Double Buffer"모듈은 2 개의 버퍼를 갖도록 설계되었다. 입력된 데이터는 공간적인 중복성 제거를 위해 DCT 와 시간적인 중복성의 제거를 위해서 프레임 예측을 통하여 양자화 및 VLC 를 거쳐서 bitstream 으로 구성된다. 예측 부호화를 위해서 양자화를 거친 데이터는 IQ 및 IDCT 를 통해서 Frame Memory 에 저장된다.



< 그림 5 > H.263+의 전체 내부 구성도

<그림 2>는 코덱의 천체 구성도로 DCT, IDCT, Q, ME 모듈에서 사용한 주요식[3]은 아래와 같다.

DCT: $$C_{m,n} = \alpha(m)\beta(n)\sum_{i=1}^{8}\sum_{j=1}^{8} B_{i,j} COS\left(\frac{\pi(2i+1)m}{16}\right) COS\left(\frac{\pi(2j+1)n}{16}\right),$$

$$\alpha(0) = \beta(0 = \sqrt{\frac{1}{8}}, \alpha(m) = \beta(n) = \sqrt{\frac{1}{4}} \qquad 0 \le m,n \le 7$$

IDCT: $$B_{i,j} = \sum_{m=1}^{8}\sum_{n=1}^{8} C_{m,n}\alpha(m) COS\left(\frac{\pi(2m+1)i}{16}\right)\beta(n)\cdot COS\left(\frac{\pi(2n+1)j}{16}\right),$$

Quantization: $$C^{q}_{m,n} = \frac{C_{m,n}}{Q_{m,n}}, 0 \le m,n \le 7 \qquad 0 \le i,j \le 7$$

여기서 Cm,n 은 DCT 계수, Qm,n 은 양자화 값이다.

ME : motion vector 를 구하기 위해서는 SAD(sum-of-absolute-differences)를 사용하여 탐색영역을 조사하여 가장 작은 값으로 motion vector 값을 구한다.

$$SAD = \sum_{k=1}^{16}\sum_{l=1}^{16} | B_{i,j}(k,l) - B_{i-u,j-v}(k,l) |$$

여기서 Bi,j(k,l)은 현재 그림의 공간영역(i,j)의 매크로블록내 (k,l)점을 나타내고, Bi-u,j-v(k,l)은 vector(u,v)에 의해서 치환된 참조그림의 공간(i,j)의 후보 매크로블록의 (k,l)점을 나타낸다.

3. H.263+ 코덱의 전반적인 동작

설계된 H.263+ 코덱의 흐름도는 아래와 같다

<Encoder>

1 IF RDY=TRUE and STP0=TRUE THEN 2 로 분기한다.
ELSE DI 로 어떠한 입력도 받지 않고 1 을 반복한다.

2 IF C/D = COMMAND(000 이외의 값) THEN 부호화에 필요한 각종 파라미터나 명령어를 DI 를 통해서 입력받는 후 1 로 분

기한다.
3 IF C/D = DATA(000) THEN DI 로 프레임 데이터를 입력받는다. // Double Buffer 모듈에 저장.
4 입력받은 프레임을 부호화 할 수 있으면(부호화 하면서 Double Buffer 에 저장할 수 있다면) RYD <= TRUE
그렇지 않다면 RYD <=FALSE.
5 전송된 프레임은 매크로블록 단위로 처리되며, 매크로블록과 이전 프레임간의 차이가 threshold 이상이면(화면의 변화가 심한 경우) 매크로블록을 INTRA(프레임내) 부호화하고 그렇지 않다면 INTER(프레임간) 부호화한다. 아래에서 (1)과 (2)는 INTRA 와 INTER 부호화 방법을 나타내고, 5.1 과 5.2 는 Picture 타입에 따른 부호화 방법을 5.3, 5.4, 5.5 는 scalability(계층구조화)에 따른 부호화 방법을 나타낸다.

(1) INTRA 경우: 매크로블록을 단지 DCT, Q 를 거쳐서 VLC 되어 bit-stream 으로 구성한다.
(2) INTER 경우: 한 매크로블록은 예측메모리를 이용하여 프레임간의 차이를 구하여 이 값을 변환기, 양자화기 VLC 를 거쳐서 bit-stream 으로 구성한다.

5.1 I-Type Picture 일 경우 : (1)만을 사용하여 부호화한다.
5.2 P-Type Picture 일 경우 : (1)나 (2)를 사용하여 부호화한다.
5.3 IF temporal –scalability-bit = true THEN 2 개의 참조프레임을 사용하여 B-Type Picture 를 생성.
5.4 IF SNR-scalability-bit = '1' THEN 보다 적은 양자화 스텝사이즈와 기본계층을 이용하여 EI, EP 를 생성한다.
5.5 IF spatial-scalability-bit='1' THEN 기본계층의 정보를 보간하여 해상도를 높인 후 EI,EP 로 부호화한다.
5.6 부호화된 bit-stream 은 최종적으로 H.263+의 계층구조의 bit-stream 으로 구성되어 코텍의 출력 버퍼로 전달된다.

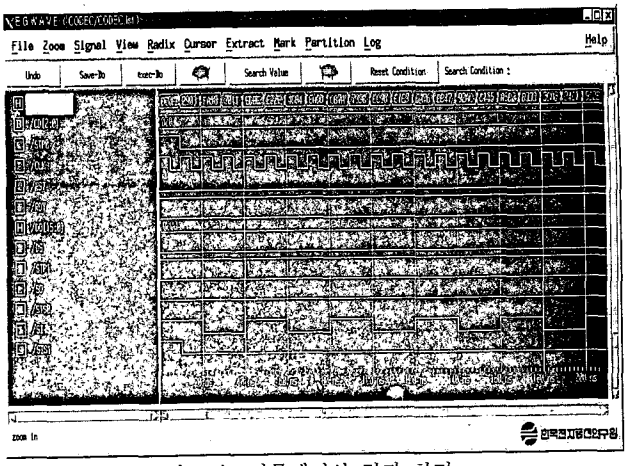
6 STS0 <= TRUE //start bit stream.
SO <= bit-stream

<Decoder>
1 SI <= TRUE // (stream-in)
bit-stream => STS1
2 Decoder 는 입력된 bit-stream 을 분석하고 엔코드의 역기능을 수행한다.
3 STP1 <= TRUE. //; (start picture)
DO <= DATA //(최종적으로 복원된 데이터를 32bits 로 전송한다.

4. VHDL 시뮬레이션

4.1 설계 환경
본 연구에서 H.263+코덱 개발에 사용한 설계 환경은 다음과 같다.
- 운영 체제 : SunOS 5.5, Solaris 2.5
- VHDL 개발 툴: LODECAP (Logic Design CAPture)2.6
- VHDL 시뮬레이션 : Mentor Graphics 사의 QuickVHDL

4.2 VHDL 시뮬레이션
본 연구에서 개발된 H.263+코덱의 기본모듈은 현재 설계된 상태이며 optional mode 은 설계 중에 있다. 모든 모듈은 VHDL 언어로 기술하고 Mentor Graphics 사의 QuickVHDL 를 사용하여 시뮬레이션하여 기능의 동작여부를 확인하였다.
<그림 6>는 시뮬레이션 결과 화면으로 주요 신호를 살펴보면 DI[31:0]및 DO[31:0]은 32bits data 입출력 값의 변화를 나타내고, CD[2:0]은 3bits 로 DI 가 명령어인지 Data 인지를 선택하는 신호를 나타낸다.



<그림 6>는 시뮬레이션 결과 화면

5. 결론 및 향후 연구과제

본 논문에서는 ITU-T 에서 규정한 H.263+에 따른 저 전송률의 비디오 코덱의 주요 모듈을 VHDL 로 기술하고 시뮬레이션하여 각 모듈의 기능을 확인하였다. 현재 H.263 를 지원하는 영상회의 시스템이나 영상전화기의 보급이 급증하는 추세에 있으므로 H.263 의 새로운 버전인 H.263+를 지원하는 시스템에 대한 요구가 증대할 것으로 기대된다. 따라서 본 연구는 향후 H.263+를 지원하는 영상회의 시스템의 핵심기술을 확립하는데 많은 도움이 될 것이다.
본 논문의 향후 연구과제로는 설계되지 않은 새로운 선택모듈의 설계와 각 모듈의 기능 검증이 필요하며, 실제 chip 으로 생산해 내는 것이다. 이를 위해서는 게이트 레벨의 시뮬레이션 검증과 실시간 처리의 검증이 필요할 것이다.

참고 문헌

[1] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate comunication." Draft ITU-T Recommendation H.263 Version2, Jan. 1998.
[2] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate comunication." Draft ITU-T Recommendation H.263, Mar. 1996.
[3] G. Côté, B. Erol, M. Gallant, and F. Kossentini. "H.263+: Video Coding at Low Bit Rates.", IEEE Transactions on Circuit and systems for Video Technology, Vol. 8, No. 7, November 1998.
[4] Low-power H.263 video CoDec dedicated to mobile computing;Morgan H.Miki, Gen Fujita, Takao Onoye, and Isao Shirakawa; Proceedings of the 1997 international symposium on Low power electronics and design , 1997, Page 80
[5] Signal Processing and Multimedia Lab., Univ. British Columbia, http://spmg,ece,ubc,ca.
[6] IEEE Standard VHDL Language Reference Manual, 1994
[7] 김영철, 정연모, 조중휘, 홍윤식 공저, 디지털 시스템 설계를 위한 VHDL, 홍릉과학출판사, 1998