

VC-1용 VLC 기반 엔트로피 디코더의 고성능 VLSI 설계에 관한 연구

A Study on a High-Performance VLSI Design of VLC-Based Entropy Decoder for VC-1

송형돈, 손승일
한신대학교

Song Hyeong-Don, Sonh Seung-Il
Hanshin Univ.

요약

Abstract

디지털 비디오 압축 기술은 대역폭과 저장 공간이 제한되는 멀티미디어 데이터의 효율적인 전송과 저장을 가능하게 하는 중요한 역할을 해왔다. 현재 VC-1은 차세대 DVD의 표준으로 채택되어 사용되고 있고 H.264/AVC의 대안으로 제안되고 있다. 이에 본 논문에서는 VC-1의 엔트로피 디코더에 대하여 소프트웨어 검증을 통한 최적화 알고리즘을 사용하여 하드웨어 설계 언어인 VHDL 언어를 이용하여 기술하고 ModeSim6.0a를 이용한 데이터 검증을 수행하였다.

I. 서론

최근 영상을 중심으로 여러 형태의 정보를 결합하여 저장하거나 전송하는 멀티미디어가 많은 관심을 받고 있으며 제한된 채널과 데이터양, 처리속도에 대한 문제점들이 제기되어 영상 압축에 대한 관심이 높아졌다. 영상 압축은 디지털 TV, 인터넷 스트리밍 비디오 그리고 DVD-비디오 같은 영상 매체에서 핵심 요소가 되고 있다. 텔레비전, DMB 등의 방송 분야 및 DVD, VCR 분야에서는 MPEG-2, MPEG-4, H.264 및 VC-1 등의 압축 코덱이 채용되고 사용되고 있다. VC-1 경우 윈도우에 부호화기가 내장되어 널리 사용되고 있고, 많은 인터넷 스트리밍 서비스(Internet Streaming Service)가 VC-1 코덱으로 제공되고 있기 때문에 콘텐츠가 매우 풍부한 장점이 있다[1-3].

전형적인 비디오 코딩 표준안들은 고정된 통계적 모델에 기반하여 비디오 데이터 스트림의 양을 줄이기 위해 자주 발생하는 데이터에 짧은 코드워드를 할당하고 자주 발생하지 않는 데이터에 대하여 긴 코드워드를 할당하여 데이터 압축을 수행하는 VLC(Variable Length Coding)를 사용한다[4]. 특히 VC-1의 경우 콘텐츠 유형에 따라 최적화된 다중의 테이블을 사용하여 VLD(Variable Length Decoding)를 수행한다. VC-1의 실시간 처리요구에 있어 VLD의 전용 하드웨어 구현을 필수적으로 요구하고 있다[5-6].

이에 본 논문은 다중의 테이블에 대하여 효율적인 VLD 수행을 위한 알고리즘을 개발하여 이를 소프트웨어로 검증을 통해 최적화하여 이를 VHDL로 기술한 후 ModelSim6.0a를 이용하여 파형을 검증하여 VC-1 모듈에서 핵심소자로 사용할 수 있도록 설계하였다.

II. 관련 연구

VLC(Variable Length Coding)의 핵심 개념은 평균적인 코드워드 길이(Average Codeword Length)를 최소화하는 것이다. 자주 발생하는 데이터는 짧은 코드워드를 할당하고 자주 발생하지 않는 데이터에 대해서는 긴 코드워드를 할당하여 데이터 압축을 수행한다. MPEG-1/2/4처럼 전형적인 MPEG 비디오 코딩 표준안들은 고정된 통계적 모델에 기반 하여 비디오 데이터 스트림의 양을 줄이기 위해 VLC기법을 사용해 왔다.

H.264/AVC에서 채택된 엔트로피 코딩 방식인 CAVLC(Context-based Adaptive Variable Length coding)에 관한 논문에 따르면 모든 테이블을 저장하는 경우 메모리에 대한 접근이 늘어나게 되고, 발생 확률에 따라 높은 발생확률의 코드워드들을 모아서 테이블화하고, 나머지 낮은 발생 확률의 코드워드는 다른 테이블로 나누어 저장하면 발생확률이 높은 것부터 활성화하여 수행하므로 테이블을 메모리에 저장하는 칩 면적은 변함이 없지만, 메모리 접근 횟수를 줄일 수 있다[7]. 메모리를 사용하지 않고 알고리즘을 적용하는 방법은 모든 경우에 대하여 테이블 없이 복호하기 때문에 메모리의 사용이 없어 칩 면적은 적게 들지만 각각의 상태에서 적용되지 않는 부분이 발생하는 단점이 있다[8].

데이터 압축률을 한층 증가시키기 위해 VC-1에서는 콘텐츠 유형에 따라 최적화된 다중의 테이블중 선택된 테이블을 사용하여 VLD를 수행한다. VLD의 단계 중에서 테이블의 개수가 많은 DC 단계와 AC 단계에서 사용되는 테이블은 전체 테이블의 85%이상을 차지한다. DC VLD에서 사용되는 테이블 수는 Low-Motion Luma(120) + Low-Motion Color Difference

(120) + High-Motion Luma(120) + High-Motion Color Difference(120) = 480 이고, AC에서 사용되는 테이블 수는 Intra와 Inter를 합쳐서 High-Motion(650+601=1251) + Low-Motin(469+538=1007) + High-Rate(577+615=1192) + Mid-Rare(378+390=768) = 4218이다[9]. VLD 구현 시 다중의 테이블 사용으로 인하여 구현 시 많은 메모리 공간이 필요하며 이로 인한 메모리 참조가 요구되며 칩의 크기가 증가하는 문제점이 발생한다.

III. VLD 개요

VLD 과정은 8x8 블록 단위로 먼저 DC 디코딩을 한 후 AC 디코딩을 수행한다. AC 디코딩은 index 단계와 run 단계, level 단계로 수행한다.

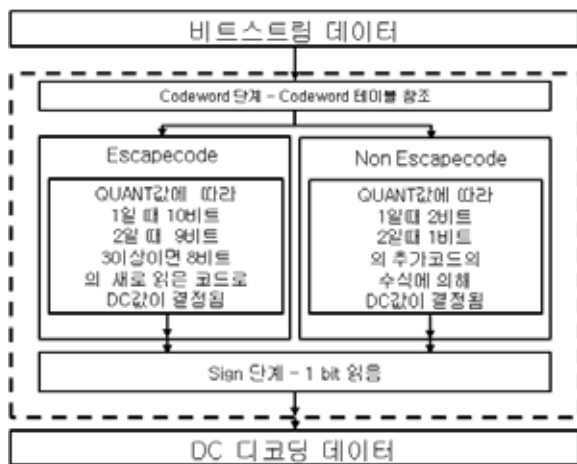
3.1 DC 비트스트림 디코딩

VC-1헤더에 정의된 TransDCTab 값에 의해 선택된 테이블에 따라 입력된 비트스트림에서 코드길기와 코드값과 일치하는 값을 찾는다. 그 값이 ESCAPE값이라면 QUANT값에 따라 10비트, 9비트, 8비트의 고정비트를 읽은 값이 DC값이 되고 ESCAPE값이 아니라면 QUANT값에 따라 1비트, 2비트를 읽어 아래 식(1)과 식(2)에 의해 DC오차값이 생성된다.

$$Quant = 1 \text{ 일 때} \\ DC오차값 = DC오차값 \times 4 + \text{추가 2비트} - 3 \quad (1)$$

$$Quant = 2 \text{ 일 때} \\ DC오차값 = DC오차값 \times 2 + \text{추가 1비트} - 1 \quad (2)$$

DC 값을 디코딩 한 후 1비트를 읽어 DC 값의 부호를 결정한다. [그림 1]은 DC 값의 디코딩 과정을 보여주고 있다.



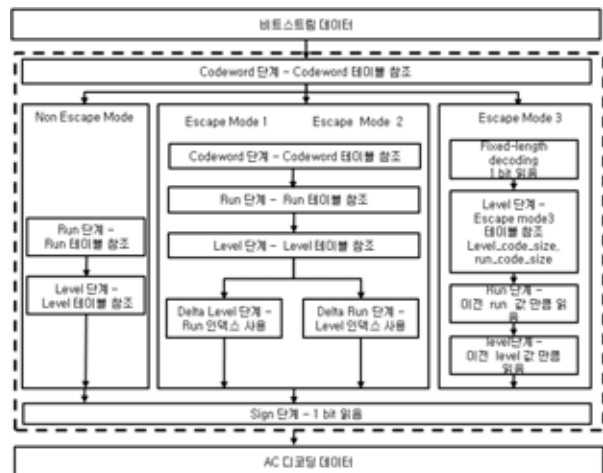
▶▶ 그림 1. DC VLD 디코딩 흐름도

3.2 AC 비트스트림 디코딩

AC 계수의 디코딩 과정은 인덱스 디코딩, run 디코딩 그리고 level 디코딩의 3단계로 되어있고 3 단계가 끝나면 1비트를 읽어 부호를 결정한다. run 단계와 level 단계는 인덱스 단계에서 디코딩된 인덱스 값에 따라 Non Escape모드와 Escape 모드로 나뉘어 디코딩된다. Escape모드는 헤더에 정의된 PQUANT값에 모드 1, 모드 2, 모드 3으로 나누어진다. 모드1과 모드 2는 다시 인덱스와 run과 level 단계를 통하여 디코딩하여 델타 run0, 델타 run1 및 델타 level0, 델타 level1 테이블을 참조하여 구한 델타run값과 델타level값을 더하여 run과 level을 디코딩한다. 모드 3은 고정된 비트를 추가로 읽어 run 값과 level값을 디코딩한다. 각각 모드에 대하여 다음과 같이 요약할 수 있다.

- (1) Escape 모드 1
run 값을 사용하여 델타 level 테이블을 참조하여 델타level 값 구하고 level 값과 더하여 level 값을 구한다.
- (2) Escape 모드 2
level 값을 사용하여 델타 run 테이블을 참조하여 델타 run 값 구하고 run 값과 더하여 run 값을 구한다.
- (3) Escape 모드 3
고정 길이만큼 읽은 level 값을 가지고 다시 비트스트림을 읽어 run 값과 level 값을 구한다.

[그림 2]는 AC VLD에 참조되는 테이블 및 NonEscape모드와 Escape모드에 따라 디코딩되는 처리 단계를 보여주고 있다.

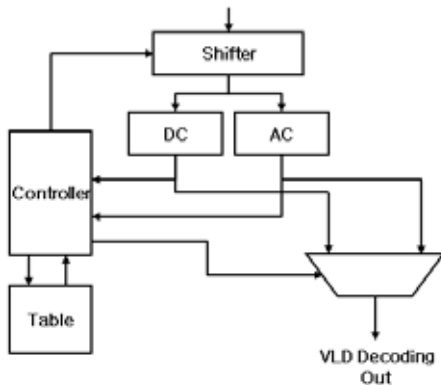


▶▶ 그림 2. AC VLD 디코딩 흐름도

IV. 제안한 엔트로피 디코더 설계

4.1 엔트로피 디코더 블록도

엔트로피 디코더는 DC 디코딩과 AC 디코딩 2부분으로 이루어져있다. DC와 AC는 테이블을 사용하여 디코딩해야 한다. 테이블을 메모리로 저장할 경우 칩의 면적 증가와 메모리 접근에 따른 성능저하가 발생한다. 이에 본 논문에서는 DC 디코딩에서 사용하는 테이블의 규칙성을 찾아 테이블의 약 44%만 사용하여 디코딩을 수행하는 알고리즘을 제안하며, 이를 적용하여 엔트로피 디코더를 설계하였다.

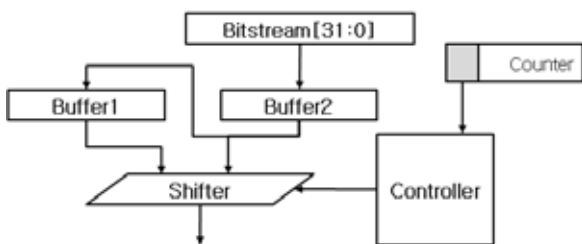


▶▶ 그림 3. 엔트로피 디코더 블록도

[그림 3]은 엔트로피 전체 블록도를 보여주고 있다. 쉬프트를 통해 32비트의 비트스트림 데이터에서 DC 디코딩을 하고, AC 디코딩을 수행한다. 컨트롤러에 의해 디코딩된 데이터를 출력한다.

4.2 쉬프트

쉬프트의 동작 신호가 인가되면 32비트의 비트스트림 데이터를 버퍼1과 버퍼2에 저장한다. 카운터는 모듈로 32로 32가 넘으면 컨트롤러에 동작을 알려준다. 이때 버퍼2의 데이터는 버퍼1로 전송되면 버퍼2는 새로운 비트스트림 데이터를 입력 받아 저장한다. 이 과정은 하나의 블록이 디코딩을 완료할 때까지 동작한다. [그림 4]는 본 논문에서 설계된 쉬프트 블록도이다.



▶▶ 그림 4. 쉬프트 블록도

4.3 테이블의 일부를 사용한 DC 디코딩

DC 디코딩은 120개로 구성된 휘도와 색차에 따라 4개의 테이블 중 선택된 테이블을 사용한다. 480개의 테이블 중에서 디코딩을 하기위해 모든 테이블을 저장해야 한다. 통계적 모델에 기반한 VLD는 자주 사용되지 않는 테이블이라도 디코더는 모든 테이블을 가지고 있어야 한다. 본 논문에서는 DC 디코딩에 사용되는 테이블을 분석을 바탕으로 수학적 연산으로 해결할 수 있는 규칙성을 도출하였다.

[표 1]은 DC 디코딩 과정에서 사용되는 4개의 테이블 중 수학적 연산으로 해결할 수 있는 테이블의 수를 보여주고 있다.

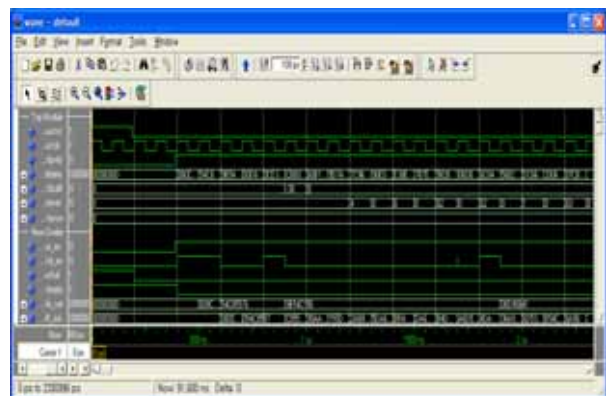
[표 1] 수학적 연산으로 처리 가능한 테이블

DC 테이블 개수	테이블 불필요 개수
Low-Motion Luma	120
Low-Motion Color-Diff	120
High-Motion Luma	120
High-Motion Color-Diff	120

디코딩될 비트스트림의 부분 비트열과 덧셈기를 사용하여 전체 480개의 테이블 중 56%에 해당하는 테이블을 사용하지 않고 DC 디코딩이 가능하였다.

V. 설계 결과

엔트로피 디코더의 입력은 32비트이며 8X8블록 단위로 처리된다. 첫 데이터를 전송받고 6클럭 이후 DC값이 출력되고 2클럭 마다 AC값이 출력된다. 본 논문에서는 출력된 DC와 AC값에 대하여 8X8블록 단위 출력은 구현하지 않았고, DC와 AC의 VLD를 구현하여 Xilinx 7.1i로 합성하고 설계 결과에 대하여 Modelsim6.0a를 사용하여 시뮬레이션 하였다. 아래 [그림 5]는 시뮬레이션 결과를 보여준다.



▶▶ 그림 5. 출력 파형

아래 [표 2]는 본 논문에서 설계한 엔트로피 디코더의 합성 결과를 보여주고 있다. 타겟 디바이스는 Xilinx의 XC2V1000을 사용하였으며, 동작 속도는 약 52MHz이다.

[표 2] 제안한 엔트로피 디코더의 합성 결과

Target Device	No. of Slices	No. of Gates	Timing
XC2V1000-6fg256	2,996	101,626	Minimum period: 19.091ns Maximum Frequency: 52.380MHz

VI. 결 론

VC-1의 엔트로피 디코더는 다중의 테이블을 사용하여 DC 디코딩 단계와 AC 디코딩 단계에서 테이블을 사용한다. 디코더를 설계할 경우 모든 테이블을 메모리에 모두 저장하여야 하며 최악의 경우 입력 데이터와 맞는 데이터를 찾기 위해 모든 메모리를 검색해야 하는 단점이 발생하고, 칩 면적이 크게 되며 처리 속도는 감소하고 전력 소모는 증가하게 된다. 이에 본 논문은 테이블에 대하여 메모리 사용에 효율적으로 모든 DC 테이블을 사용하지 않고 상위 비트를 확인하여 덧셈기를 사용하여 VLD를 처리하는 알고리즘을 적용하여 테이블의 약 44%만을 메모리로 저장하여 처리하였다. 칩 면적을 줄이고 저전력의 고속 엔트로피 디코더를 FPGA로 구현 및 검증함으로써 VC-1 모듈에서 핵심 소자로 사용이 가능할 것으로 사료된다.

오늘날 발표된 모든 영상 코덱은 압축을 위해 가변길이 코딩을 사용하고 있다. 향후 본 논문에서 설계한 엔트로피 디코더를 VC-1이 디코더에 탑재하여 검증을 수행할 예정이며 다중의 테이블간의 연관성을 분석하여 테이블의 사용을 좀더 줄일 수 있는 엔트로피 디코더 대한 연구를 진행할 예정이다.

■ 참고 문헌 ■

- [1] Microsoft, "VC-1 Technical Overview," Official Windows Media Web site, <http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx>
- [2] "Trade-offs with H.264, VC-1 and other advanced code," <http://www.ti.com>
- [3] "TMS320DM6446 Digital Media System-on-Chip," <http://www.ti.com>
- [4] Tung-Chien Chen, Chung-Jr Lian and Liang-Gee Chen, "Hardware architecture design of an H.264/AVC video codec," in Proc. Asia and South Pacific Conf. Design Automation, Jan. 2006
- [5] Ming-Ting Sun, "VLSI Architecture and implementation of a High-Speed entropy decoder," IEEE Int. Symp. Circuits Syst., pp.200-203, Jun. 1991.
- [6] Masaru Hase, Kazushi Akie, Masaki Nobori, Keisuke Matsumoto, "Development of Low-power and Real-time VC-1 and H264 and MPEG-4 Video Processing Hardware", in Proc. Asia and South Pacific Conf. Design Automation, pp 637-642. Jan. 2007
- [7] Wu Di, Gao Wen, Hu Mingzeng, Ji Zhenzhou, "A VLSI Architecture Design of CAVLC Decoder," in Proc. of 5th IEEE Int. Conference on ASIC, pp.962-965. Oct 2003,
- [8] Yong Ho Moon, Gyu Yeong Kim, Jae Ho Kim, Member, IEEE, "An Efficient Decoding of CAVLC in H.264/AVC Video Coding Standard", IEEE. 2005
- [9] SMPTE 421M (2005): "VC-1 Compressed Video Bitstream Format and Decoding Process," Draft Standard, August. 2005.