

HEVC 부호화기를 위한 효율적인 적응적 루프 필터 설계

신승용 · 박승용 · 류광기
한밭대학교 정보통신공학과

An Efficient Adaptive Loop Filter Design for HEVC Encoder

Seung-yong Shin · Seung-yong Park · Kwang-ki Ryoo

Dept. of Information Communication Eng., Hanbat National University

E-mail : ssingss@naver.com

요 약

본 논문에서는 필터 계수 추출을 위한 HEVC 적응적 루프 필터(ALF, Adaptive Loop Filter)의 효율적인 설계를 제안한다. ALF는 필터 계수를 추출하기 위해 10×10 행렬의 쉐레스키 분해를 반복적으로 수행한다. ALF의 쉐레스키 분해는 루트 연산 및 나눗셈 연산 등 하드웨어로 설계하기 어려운 연산들로 구성되어 있고, LCU(64×64) 한 개당 최대 30비트의 큰 값들을 소수점 단위로 연산하기 때문에 많은 연산량과 수행 시간을 필요로 한다. 본 논문에서 제안한 하드웨어 구조는 멀티플렉서와 뺄셈기, 비교기 등을 이용하여 쉐레스키 분해에 사용되는 루트 연산을 구현하였다. 또한, 쉐레스키 분해의 특징적인 연산 과정들을 파이프라인 구조로 설계함으로써 효율적이면서 적은 연산량을 갖는 하드웨어 구조로 구현하였다. 구현한 하드웨어는 Xilinx ISE 14.3 Vertex-6 XC6VCX240T FPGA 디바이스를 사용하여 설계하였으며, 최대 동작 주파수 150MHz에서 4K UHD(4096×2160) 영상을 초당 40프레임으로 실시간 처리할 수 있다.

ABSTRACT

In this paper, an efficient design of HEVC Adaptive Loop Filter(ALF) for filter coefficients estimation is proposed. The ALF performs Cholesky decomposition of 10×10 matrix iteratively to estimate filter coefficients. The Cholesky decomposition of the ALF consists of root and division operation which is difficult to implement in a hardware design because it needs to many computation rate and processing time due to floating-point unit operation of large values of the Maximum 30bit in a LCU(64×64). The proposed hardware architecture is implemented by designing a root operation based on Cholesky decomposition by using multiplexer, subtractor and comparator. In addition, The proposed hardware architecture of efficient and low computation rate is implemented by designing a pipeline architecture using characteristic operation steps of Cholesky decomposition. An implemented hardware is designed using Xilinx ISE 14.3 Vertex-6 XC6VCX240T FPGA device and can support a frame rate of 40 4K Ultra HD(4096×2160) frames per second at maximum operation frequency 150MHz.

키워드

HEVC Encoder, HM-7.0, Adaptive Loop Filter(ALF), Cholesky decomposition, Hardware design

I. 서 론

최근 다양한 멀티미디어 기기들의 등장으로 인해 Ultra High Definition(UHD)와 같이 초고해상도 및 초고화질의 영상을 요구하는 경우가 점차 늘어나고 있다. 이러한 요구에 부응하여 ITU-T VCEG(Video Coding Experts Group)과 ISO/IEC MPEG(Moving Picture Experts Group)으로 구성된 JCT-VC(Joint Collaborative Team on Video

Coding)은 차세대 비디오 부호화 기술인 HEVC(High Efficiency Video Coding)를 개발하기 시작하였고, 2013년 1월 표준화를 완료하였다. HEVC는 기존의 비디오 코덱 표준인 H.264/AVC에 비해 약 2배의 부호화 성능을 보인다. 하지만, HEVC는 부호화 성능을 향상시키기 위해 추가적인 기술들을 적용함으로써 연산 복잡도가 크게 증가하였다. HEVC는 부호화 성능을 향상시키기 위한 여러 가지 기술들이 존재하지만, 본 논문

서는 그 중에서 양자화에 의해 발생한 오류들을 보상함으로써 주관적 화질과 압축 효율 모두를 향상시키는 적응적 루프 필터(ALF, Adaptive Loop Filter) 기술에 초점을 맞추고 있다[1,2].

ALF는 위너 필터를 기반으로 원본 영상과 복원된 영상 간의 평균자승오차를 최소화시키는 기술에 해당된다. ALF는 HEVC Model(HM)-7.0에서 채택하고 있는 9×7 십자가 모양에 3×3 직사각형 모양이 합쳐진 필터 형태를 가지며, 필터의 대칭적인 구조에 따라 총 10개의 필터 계수를 갖는다. ALF는 10개의 필터 계수를 추출하기 위해 10×10 행렬의 출레스키 분해를 반복적으로 수행한다. ALF는 10×10 행렬의 출레스키 분해를 수행할 때, HEVC에서 처리하는 최대 블록 크기인 64×64 화소 단위 한 개당 최대 30비트에 해당하는 큰 값의 상관관계 정보를 입력으로 받는다. ALF는 입력 받은 상관관계 정보를 통해 루트 연산 및 나눗셈 연산을 소수점 단위로 연산하기 때문에 많은 연산량과 수행 시간을 필요로 한다. 또한, 10×10에 해당하는 상당히 큰 행렬의 출레스키 분해는 각각의 행과 열이 갖는 특징적인 구조로 연산하기 때문에 많은 연산량과 수행 시간을 차지하게 된다[2,3].

본 논문에서는 ALF의 필터 계수를 추출하기 위해 반복적으로 수행하는 10×10 행렬의 출레스키 분해 과정을 루트 연산, 나눗셈 연산, 소수점 연산 그리고 출레스키 분해의 행과 열이 갖는 특징적인 연산 구조 등을 효율적으로 연산하기 위한 측면에서 고려하여 설계하였다. 그 결과 루트 연산은 멀티플렉서와 뺄셈기, 비교기 등을 이용한 구조로 설계하였고, 출레스키 분해의 특징적인 연산 구조는 행과 열 단위를 파이프라인 구조로 설계하여 연산량 및 수행 시간을 최소화하였다.

본 논문의 구성은 다음과 같다. 2장에서는 ALF의 전체적인 연산 과정과 필터 계수를 추출하기 위한 출레스키 분해의 알고리즘에 대하여 기술하고, 3장에서는 제안하는 ALF의 하드웨어 구조를 기술한다. 4장에서는 하드웨어 합성 결과를 기술하며, 마지막으로 5장에서는 결론으로 끝을 맺는다.

II. 적응적 루프 필터

ALF의 전체적인 연산 과정은 총 세 가지의 하위 과정으로 구성된다[4]. 첫 번째 과정인 바운드리 패딩은 복원된 영상의 바깥쪽 경계나 필터링하고자 하는 블록의 경계에서 필요한 픽셀 값들이 존재하지 않는 부분을 이웃한 픽셀 값들로 채워준다. 두 번째는 필터 계수들을 추출하는 과정이다. 필터 계수들을 추출하는 과정은 복원된 과거 영상 또는 현재 영상을 이용하여 위너 필터 기반의 통계적 특성을 통해 필터 계수들을 계산한다. 두 번째 과정이 본 연구의 핵심부분에 해당하며, 중심적으로 다루고자 한다. 마지막 과정은

복원된 영상의 픽셀 값들과 계산한 필터 계수들을 이용하여 그림 1과 같이 대칭적인 구조의 필터 형태로 필터링을 수행한다.

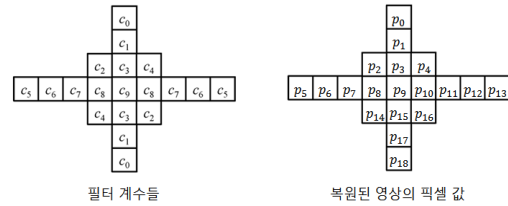


그림 1. HM-7.0의 ALF 필터 형태

아래 식 (1)은 ALF결과 영상 $f[r]$ 을 복원된 영상 $t[r]$ 과 필터계수 C_n 와의 관계를 식으로 나타낸 것이다. 여기서 P_n 은 그림 1의 오른쪽 필터 형태에서 n 번째에 해당하는 위치 정보를 의미한다[5].

$$f[r] = \sum_{n=0}^{N-1} c_n t[r+p_n] \quad (1)$$

식 (1)을 이용하여 $f[r]$ 과 원본 영상 $s[r]$ 간 최소의 오차제곱합을 계산하기 위해 '0' 과 같다고 가정하고, ALF 필터 형태의 대칭적인 구조를 적용하여 계산하면, 식 (2)와 같이 위너-흡 식을 유도할 수 있다. 여기서 R 은 복원된 영상 $t[r]$ 의 블록 단위 크기를 의미한다.

$$\begin{bmatrix} \sum_{r=0}^{R-1} t[r+p_0]t[r+p_0] & \sum_{r=0}^{R-1} t[r+p_0]t[r+p_1] & \dots & \sum_{r=0}^{R-1} t[r+p_0]t[r+p_9] \\ \sum_{r=0}^{R-1} t[r+p_1]t[r+p_0] & \sum_{r=0}^{R-1} t[r+p_1]t[r+p_1] & \dots & \sum_{r=0}^{R-1} t[r+p_1]t[r+p_9] \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{r=0}^{R-1} t[r+p_9]t[r+p_0] & \sum_{r=0}^{R-1} t[r+p_9]t[r+p_1] & \dots & \sum_{r=0}^{R-1} t[r+p_9]t[r+p_9] \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_9 \end{bmatrix} = \begin{bmatrix} \sum_{r=0}^{R-1} s[r]t[r+p_0] \\ \sum_{r=0}^{R-1} s[r]t[r+p_1] \\ \vdots \\ \sum_{r=0}^{R-1} s[r]t[r+p_9] \end{bmatrix} \quad (2)$$

왼쪽의 10×10 행렬은 자기 상관관계를 의미하고, 오른쪽의 10×1 벡터는 상호 상관관계를 의미한다. 여기서 t' 는 식 (4)와 같다.

$$t'[r+p_n] = t[r+p_n] + t[r+p_{18-n}] \quad (3)$$

식 (2)의 위너-흡 식에서 필터 계수 C_n 를 계산하기 위해 가우시안 제거 알고리즘에서 대각선을 기준으로 좌우 대칭적인 행렬을 가질 때 사용할 수 있는 출레스키 분해법을 이용한다. 출레스키 분해법을 이용한 전체적인 연산 과정은 그림 2와 같다. Factorization부분은 식 (4)를 이용하여 10×10 행렬 U 와 U 의 전치행렬을 계산한다. Forward부분은 계산한 U 의 전치행렬을 이용하여 10×1 벡터 d 를 계산하고, Back부분은 계산한 행렬 U 와 벡터 d 를 이용하여 최종 해인 x 를 계산한다[3].

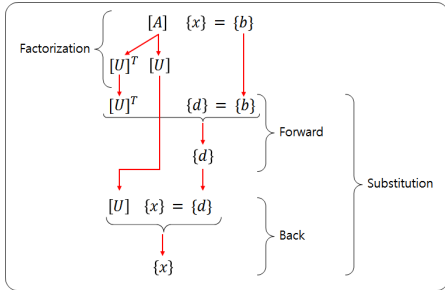


그림 2. 출레스키 분해법을 이용한 전체적인 연산 과정

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ki} u_{ki}} \quad u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki} \times u_{kj}}{u_{ii}} \quad (4)$$

식 (4)에서 a는 식(2)의 왼쪽 10×10 행렬에서 i 위치와 j위치에 해당하는 요소이고, j는 i+1부터 시작하여 행렬의 최대 크기까지를 의미한다.

III. 제안하는 적응적 루프 필터 하드웨어 구조

본 논문에서 제안하는 ALF 하드웨어 구조는 그림 3과 같다. 제안하는 ALF는 그림 2의 출레스키 분해법을 이용한 전체적인 연산 과정과 같이 Factorization_top 모듈, Forward_top 모듈, Back_top 모듈 그리고 최종적인 값의 정수표현 및 반올림을 위한 Shift_Round_top 모듈로 구성된다.

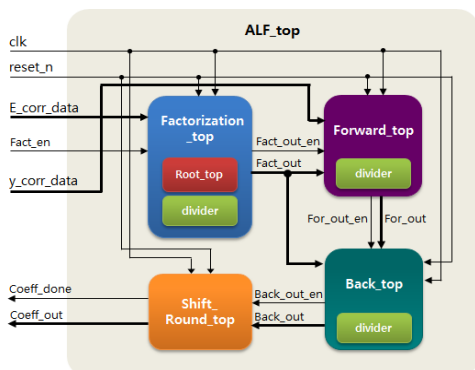


그림 3. 제안하는 ALF_top 하드웨어 구조

제안하는 ALF 하드웨어는 Factorization_top 모듈과 Forward_top 모듈간의 파이프라인 처리가 가능하고, Forward_top 모듈과 Back_top 모듈은 Factorization_top 모듈의 결과 값인 Fact_out 신호를 필요로 하기 때문에 결과 값을 저장하는 구조로 설계하였다. 또한, Factorization_top 모듈, Forward_top 모듈, Back_top 모듈의 내부에 각각

디바이더를 사용하여 병렬적으로 나눗셈 연산을 수행할 수 있도록 설계하였다. 그림 4는 제안하는 Factorization_top 모듈의 하드웨어 구조를 나타낸다. Factorization_top 모듈은 앞에서 미리 계산된 상관관계 정보 E_corr_data 신호를 입력 받고, 출력인 Fact_out 신호는 Forward_top 모듈과 Back_top 모듈의 입력 신호로 받는다.

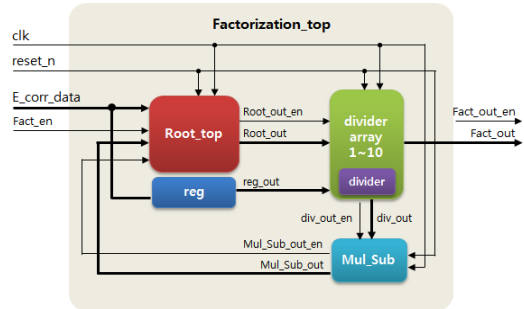


그림 4. 제안하는 Factorization_top 하드웨어 구조

Factorization_top 모듈에서 루트를 연산하는 Root_top 모듈은 그림 5와 같이 비교기, 멀티플렉서, 뺄셈기 등을 이용하여 설계한 하드웨어 구조를 나타낸다. Root_top 모듈은 상관관계 정보에 따라 최대 42비트까지 연산이 가능하고 파이프라인 구조로 설계하여 20사이클 만에 연산 결과를 얻을 수 있다[6]. divider_array1~10 모듈은 Root_out 신호를 입력받아 레지스터에 저장한 후, E_corr_data 신호와 나눗셈 연산 기능을 수행한다. 또한, 나눗셈 연산을 수행한 결과 값은 행과 열 각각 디바이더 배열을 통해 저장된다. Mul_Sub 모듈은 나눗셈 연산을 수행한 div_out 신호를 입력 받는다. Mul_Sub 모듈은 입력받은 div_out 신호의 몫과 나머지를 각각 곱셈 연산을 수행한 후, 레지스터에 저장된 E_corr_data 신호와 뺄셈 연산을 수행한다.

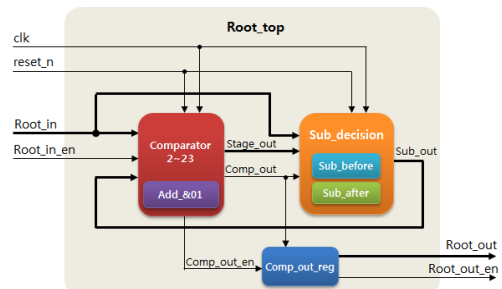


그림 5. 제안하는 Root_top 하드웨어 구조

그림 5에서 Comparator2~23 모듈은 Root_in 신호와 Sub_decision 모듈의 결과 값인 Sub_out 신호에 2비트인 “01” 을 오른쪽에 추가적으로 연결한 값을 비교하는 기능을 수행한다. 그리고 결과 값인 Comp_out 신호는 Comp_out_reg 모듈에 저장되고, 마지막인 LSB(Least Significant Bit)

까지 연산을 완료하면 저장했던 결과 값을 모두 연결하여 Root_out 신호로 출력한다.

본 논문에서는 HEVC 표준 참조 소프트웨어인 HM-7.0에서 데이터를 추출하였고[3], 제안한 하드웨어 구조를 통해 시뮬레이션 한 결과 참조 소프트웨어에서 추출한 데이터와 비교하여 정상적으로 동작함을 확인하였다. Factorization_top 모듈은 42비트 루트 연산에서 20 사이클이 소요되고 나눗셈 연산에서 45 사이클이 소요된다[7]. 10×10 행렬의 출레스키 분해를 연산하기 위해서는 총 10번의 루트 연산 및 나눗셈 연산을 수행하기 때문에 총 650 사이클이 소요된다. 또한, Forward_top 모듈은 Factorization_top 모듈과 파이프라인 동작으로 나눗셈 연산을 한번만 수행하여 총 45 사이클을 소요하고, Back_top 모듈은 총 10번의 나눗셈 연산을 수행하기 때문에 총 450 사이클이 소요된다. 마지막인 Shift_Round_top 모듈의 수행 사이클까지 모두 더하면 총 1155 사이클이 소요된다.

IV. 하드웨어 합성 결과

본 논문에서 제안하는 ALF 하드웨어 구조 Verilog HDL로 설계하였다. 하드웨어 합성은 IDEC에서 지원하는 Xilinx ISE 14.3 Vertex-6 XC6VCX240T FPGA 디바이스를 사용하여 구현하였다. Divider는 Xilinx core generator에서 지원하는 IP를 사용하여 설계하였으며[7], 표 1은 제안하는 ALF 하드웨어 구조의 합성 결과를 나타낸다.

표 1. 제안하는 ALF 하드웨어 합성 결과

구분	결과
동작 주파수	150MHz
사이클 수	1155
슬라이스 레지스터 수	51244(17%)
슬라이스 LUT 수	16579(11%)

V. 결 론

본 논문에서는 HEVC 부호화기 ALF의 연산량과 수행 시간을 최소화하기 위해 루트 연산을 멀티플렉서, 뿔셈기, 비교기 등을 이용하여 설계하였고, 나눗셈 연산을 수행하는 모듈들은 파이프라인 구조로 설계하였다. 또한, 나눗셈 연산을 수행하여 얻은 결과 값을 필요로 하는 모듈에서 사용할 수 있도록 디바이더 배열에 저장하는 구조로 설계하였다. 제안하는 ALF 하드웨어 구조를 Xilinx ISE 14.3 Vertex-6 XC6VCX240T FPGA에서 합성한 결과 최대 동작 주파수는 150MHz이고, 총 1155 사이클이 소요된다. 또한, 4K UHD 영상을 초당 40 프레임으로 실시간 처리할 수 있다.

제안하는 ALF 하드웨어 구조는 전체 ALF 프로세스 중에서도 필터 계수를 추출하는 부분에 대해서 집중적으로 다루었다. 향후 연구 방향은 나머지 프로세스에 해당하는 바운드리 패딩 과정과 최적의 필터 계수를 추출하기 위한 영역 기반 과정, 필터링 과정을 하드웨어 구조로 설계하여 전체적인 ALF Top 모듈을 설계하는 것이 되겠다.

감사의 글

본 연구는 교육부와 한국연구재단의 지역혁신인력양성사업(NRF-2012H1B8A2025862)과 미래창조과학부 및 정보통신산업진흥원의 해외인재스카우팅사업(NIPA-HB616-13-1001)의 지원으로 수행되었습니다.

참고문헌

- [1] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding(HEVC) Standard," IEEE Transaction on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [2] "High-efficiency video coding text specification draft 7," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG, JCTVC-I1003, May. 2012.
- [3] "HM7.0: High efficiency video coding HEVC test model 7.0," (https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-7.0/)
- [4] C. Cristani, P. Dall'Oglio and M. Porto, "High Throughput Hardware Design for the Adaptive Loop Filter of the Emerging HEVC Video Coding," Intergrated Circuits and Systems Design (SBCCI), Aug. 2012.
- [5] C.Y. Tasi, C.Y. Chen, T. Yamakage, I.S. Chong, Y.W. Huang, C.M. Fu, T. Itoh, T. Watanabe, T. Chujoh, M. Karczewicz and S.M. Lei, "Adaptive Loop Filtering for Video Coding," IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 6, pp. 934-945, Dec. 2013.
- [6] T. Sutikno, "An Optimized Square Root Algorithm for Implementation in FPGA Hardware," TELKOMNIKA Indonesian Journal of Electronic Engineering, vol. 8, no. 1, pp. 1-8, Apr. 2010.
- [7] Xilinx Logicore, "Divider Generator v4.0," (<http://www.xilinx.com>, Jun. 2011)