

효율적인 필터 계수 추출을 위한 HEVC 부호화기의 고성능 ALF 하드웨어 설계

신승용 · 류광기*

Hardware Design of High Performance ALF in HEVC Encoder for Efficient Filter Coefficient Estimation

Seungyong Shin · Kwangki Ryoo*

Department of Information and Communication Engineering, Hanbat National University, Daejeon 305-791, Korea

요 약

본 논문에서는 필터 계수를 효율적으로 추출하기 위한 고성능 ALF(Adaptive Loop Filter)의 하드웨어 구조를 제안한다. HEVC의 ALF 기술은 고해상도 및 고화질의 영상을 높은 효율로 압축하고 주관적 화질을 향상시키기 위해 영상의 통계적인 특성을 이용한 필터 계수를 추출하여 필터링을 수행한다. 제안하는 ALF 하드웨어 구조는 필터 계수를 추출하기 위한 촘레스키 분해의 연산 관계를 분석하여 2단 파이프 구조로 설계함으로써 수행 사이클을 감소시켰다. 또한, 촘레스키 분해의 연산 과정에서 필요한 루트 연산은 멀티플렉서와 뺄셈기, 비교기 등을 이용하여 설계함으로써 적은 면적과 연산량, 복잡도를 갖는 하드웨어 구조로 설계하였다. 제안한 하드웨어는 Xilinx ISE 14.3 Vertex-7 XC7VCX485T FPGA 디바이스를 사용하여 합성한 결과 4K(3840x2160)@40fps의 영상을 실시간 처리할 수 있고, 최대 동작주파수는 186MHz이다.

ABSTRACT

This paper proposes the hardware architecture of high performance ALF(Adaptive Loop Filter) for efficient filter coefficient estimation. In order to make the original image which has high resolution and high quality into highly compressed image effectively and also, subjective image quality into improved image, the ALF technique of HEVC performs a filtering by estimating filter coefficients using statistical characteristics of image. The proposed ALF hardware architecture is designed with a 2-step pipelined architecture for a reduction in performance cycle by analysing an operation relationship of Cholesky decomposition for the filter coefficient estimation. Also, in the operation process of the Cholesky decomposition, a square root operation is designed to reduce logic area, computation time and computation complexity by using the multiplexer, subtractor and comparator. The proposed hardware architecture is designed using Xilinx ISE 14.3 Vertex-7 XC7VCX485T FPGA device and can support 4K UHD@40fps in real time at a maximum operation frequency of 186MHz.

키워드 : HEVC encoder, ALF(적응적 루프 필터), 촘레스키 분해, FPGA

Key word : HEVC encoder, ALF(Adaptive Loop Filter), Cholesky decomposition, FPGA

접수일자 : 2014. 11. 15 심사완료일자 : 2015. 01. 12 게재확정일자 : 2015. 01. 26

* **Corresponding Author** Kwangki Ryoo(E-mail:kkryoo@gmail.com, Tel:+82-42-821-1710)

Department of Information and Communication Engineering, Hanbat National University, Daejeon 305-791, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.2.379>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

최근 UHD(Ultra High-Definition) 방송 상용화와 더불어 고화질 콘텐츠 시청에 대한 소비자들의 욕구가 증가하면서 UHD TV 대중화가 급속히 진행되고 있다. 또한, 모바일 시장에서도 기본화면 해상도가 HD급인 스마트폰이 출시되면서 고해상도 및 고화질의 영상에 대한 사용자들의 수요가 급증하고 있는 추세이다. 그러나 고해상도 및 고화질의 영상은 데이터의 양이 매우 방대하기 때문에 기존에 쓰이던 표준 영상 압축 기술인 H.264/AVC는 고해상도 및 고화질 영상서비스를 지원하기에는 한계가 있다. 이에 따라 ITU-T VCEG(Video Coding Experts Group)와 ISO/IEC MPEG(Moving Picture Experts Group)은 H.264/AVC보다 더 높은 압축률과 더 낮은 복잡도의 새로운 차세대 영상 압축 표준의 필요성에 의해, 2010년 초 HEVC(High Efficiency Video Coding)라는 표준화 활동을 시작했으며, 2013년 1월 초 스위스 제네바 회의에서 HEVC 최종 표준안(FDIS, Final Draft International Standard)을 완성하였다. HEVC는 기존의 H.264/AVC에 비해 약 35%의 부호화 효율을 보인다[1].

HEVC의 ALF 기술은 기존의 H.264/AVC보다 더 높은 압축 성능과 주관적 화질을 향상시키기 위해 HEVC에 새로 삽입된 루프내(In-loop) 필터링 방법 중 가장 마지막으로 처리하는 필터이다. ALF 기술은 양자화 과정에서 발생하는 정보의 손실을 보상하기 위해서 복원된 영상에 적응적으로 필터링을 수행하는 기술이다. 즉, ALF는 위너 필터를 기반으로 원본 영상과 복원된 영상 간의 평균자승오차를 최소화시키는 기술이다. ALF 기술은 적응적으로 필터링을 수행하기 위해 필터 계수를 추출하기 위한 10×10 행렬의 출레스키 분해를 반복적으로 수행하기 때문에 복잡한 연산 과정과 상당한 연산 수행시간을 필요로 한다. 또한, ALF는 HEVC에서 처리하는 최대 블록 크기인 64×64 화소 단위로 연산을 수행하기 때문에 많은 연산량과 수행시간을 요구한다[2].

본 논문에서 제안한 ALF 하드웨어 구조는 필터 계수를 효율적으로 추출하기 위한 10×10 행렬의 출레스키 분해의 특징적인 연산 관계를 분석하여 전체적인 수행 사이클을 감소시킨 2단 파이프라인 구조로 설계하였다. 또한, 출레스키 분해에 사용되는 루트 연산은 멀티플렉서와 뺄셈기, 비교기 등을 이용한 구조로 설계하여 연

산량 및 연산 수행시간을 최소화하였다.

II. Adaptive Loop Filter

ALF 기술은 양자화에 의해 발생한 오류들을 보상하여 주관적 화질과 압축 효율을 향상시키는 기술로써, 그림 1과 같이 HEVC 부호화기에 새로 삽입된 루프내 필터링 방법이다. ALF의 특징은 각 영상에 대해 최적의 필터 계수를 적응적으로 적용함으로써 주관적 화질과 압축 효율을 상당히 향상시킬 수 있다[3].

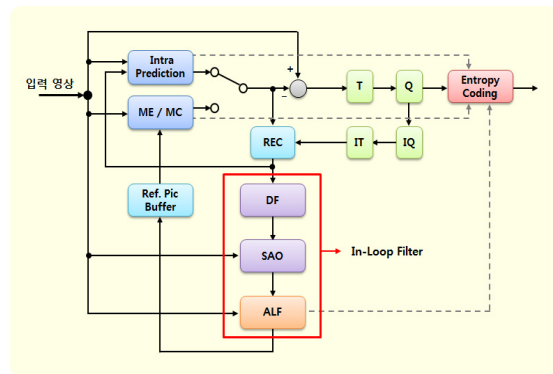


그림 1. HM 7.0 부호화기 블록 다이어그램
Fig. 1 Block diagram of HM 7.0 Encoder

ALF의 전체적인 연산 과정은 총 세 가지의 하위 과정으로 구성된다[4]. 첫 번째 과정인 바운드리 패딩은 복원된 영상의 바깥쪽 경계나 필터링 하고자 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 통계적인 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 2와 같은 필터 형태로 필터링을 수행한다. ALF는 그림 2와 같이 HEVC Model(HM)7.0에서 채택하고 있는 9×7 십자가 모양에 3×3 정사각형 모양이 합쳐진 필터 형태를 가지며, 필터의 대칭적인 구조에 따라 총 10개의 필터 계수를 갖는다[5].

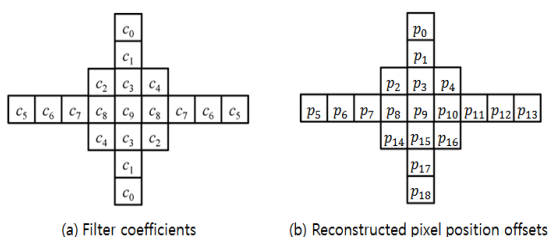


그림 2. HM 7.0의 ALF 필터 형태
Fig. 2 ALF filter shape of HM 7.0

ALF는 위너 필터 개념을 기반으로 원본 영상과 복원된 영상 간의 평균자승오차(MSE, Mean Square Error)를 최소화함으로써 복원된 영상이 원본 영상에 가까운 신호로 필터링을 수행한다. 식 1은 위너 필터 기반의 ALF 필터링 결과를 수식으로 나타낸 것이다. 식 1에서 $f[r]$ 은 ALF의 결과 영상을 의미하고, $t[r]$ 은 복원된 영상(Sample Adaptive Offset)을 의미한다. r 은 2D 영상의 $[x,y]$ 에 해당하는 좌표 정보를 의미하고, c_n 은 ALF의 필터 계수를 의미한다. p_n 은 필터링을 수행하기 위한 위치 오프셋을 의미한다.

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

식 1을 이용하여 ALF 결과 영상 $f[r]$ 과 원본 영상 $s[r]$ 간의 최소의 오차제곱합(SSE, Sum of Squared Error)을 계산하기 위해 '0'과 같다고 가정하고 ALF 필터 형태의 대칭적인 구조를 적용하여 계산하면, 식 2와 같이 위너-흡 방정식을 유도할 수 있다. 식 2에서 R 은 복원된 영상 $t[r]$ 의 블록 단위 크기를 의미한다. 왼쪽 항의 10×10 행렬은 자기 상관관계를 의미하고, 오른쪽 항의 10×1 벡터는 상호 상관관계를 의미한다. t' 는 식 3을 이용하여 계산할 수 있다[6].

$$\begin{bmatrix} \sum_{r=0}^{R-1} t'[r+p_0]t'[r+p_0] & \sum_{r=0}^{R-1} t'[r+p_1]t'[r+p_0] & \dots & \sum_{r=0}^{R-1} t'[r+p_9]t'[r+p_0] \\ \sum_{r=0}^{R-1} t'[r+p_0]t'[r+p_1] & \sum_{r=0}^{R-1} t'[r+p_1]t'[r+p_1] & \dots & \sum_{r=0}^{R-1} t'[r+p_9]t'[r+p_1] \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{r=0}^{R-1} t'[r+p_0]t'[r+p_9] & \sum_{r=0}^{R-1} t'[r+p_1]t'[r+p_9] & \dots & \sum_{r=0}^{R-1} t'[r+p_9]t'[r+p_9] \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_9 \end{bmatrix} = \begin{bmatrix} \sum_{r=0}^{R-1} s[r]t'[r+p_0] \\ \sum_{r=0}^{R-1} s[r]t'[r+p_1] \\ \vdots \\ \sum_{r=0}^{R-1} s[r]t'[r+p_9] \end{bmatrix} \quad (2)$$

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

식 2의 위너-흡 방정식에서 필터 계수 $C_0 \sim C_9$ 를 계산하기 위해 가우시안 제거 알고리즘을 이용한다. 기본적인 연립방정식을 이용한 가우시안 제거 알고리즘은 10개의 미지수인 필터 계수들을 계산하는데 연산량과 복잡도가 매우 많기 때문에, 자기 상관행렬인 10×10 행렬 부분의 대칭적인 특성을 이용한 출레스키 분해법을 이용하여 효율적으로 필터 계수들을 계산할 수 있다. 출레스키 분해법을 이용한 전체 연산 과정은 그림 3과 같다. 그림 3에서 첫 번째 과정인 Factorization 연산 과정은 10×10 행렬 A 를 식 4와 식 5를 이용하여 10×10 행렬 U 와 U 의 전치행렬 U^T 를 계산한다. 두 번째 과정인 Forward 연산 과정은 Factorization 연산 과정에서 계산한 행렬 U^T 를 이용하여 10×1 벡터 d 를 계산한다. 마지막으로 Back 연산 과정은 Factorization 연산 과정에서 계산했던 행렬 U 와 Forward 연산 과정에서 계산한 벡터 d 를 이용하여 최종 해인 10×1 벡터 x 를 계산한다[7].

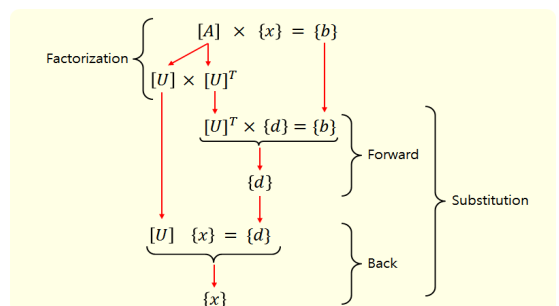


그림 3. 출레스키 분해법을 이용한 가우시안 제거 알고리즘
Fig. 3 Gaussian elimination algorithm using Cholesky decomposition

식 4, 5는 10×10 행렬 U 와 U 의 전치행렬 U^T 를 계산하기 위한 출레스키 분해법의 방정식이다. 식 4, 5에서 a 는 10×10 행렬 A 에서 i 위치와 j 위치에 해당하는 행렬 요소를 의미하고, j 는 $i+1$ 위치부터 시작하여 행렬의 최대 크기까지인 10을 의미한다.

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

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki} \times u_{kj}}{u_{ii}} \quad (5)$$

III. 제안하는 ALF 하드웨어 구조

제안하는 ALF 하드웨어 구조는 수행 사이클을 감소시키기 위해 출레스키 분해의 특징적인 연산 관계를 분석하여 병렬적으로 연산을 수행하기 위한 2단 파이프라인 구조로 설계하였고, 출레스키 분해에 사용되는 루트 연산은 연산량 및 연산 시간을 감소시키기 위해 멀티플렉서와 뺄셈기, 비교기를 이용하여 설계하였다. 그림 4는 제안하는 ALF 하드웨어 구조를 나타낸다.

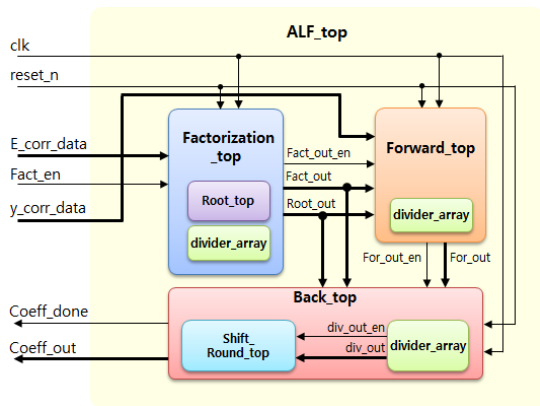


그림 4. 제안하는 ALF 하드웨어 구조
Fig. 4 Proposed ALF hardware architecture

제안하는 구조는 Factorization_top 모듈, Forward_top 모듈, Back_top 모듈로써 총 3개의 상위 모듈로 구성되며, 각 모듈 내부에 Root_top 모듈, divider_array 모듈 등의 하위 모듈로 구성된다. Factorization_top 모듈은 내부에 루트 연산을 수행하는 Root_top 모듈과 나눗셈 연산을 수행하고 레지스터 배열에 저장하는 divider_array 모듈로 구성되어 있고, 10×10 행렬 U 를 계산하는 기능을 수행한다. Forward_top 모듈은 Factorization_top 모듈에서 계산한 10×10 행렬 U 의 값을 입력받아 10×1 벡터 d 를 계산하는 기능을 수행한다. Back_top 모듈은 쉬프트와 라운딩을 연산하는 Shift_Round_top 모듈과 divider_array 모듈로 구성되어 있고, Factorization_top 모듈에서 계산한 10×10 행렬 U 의 값과 Forward_top 모듈에서 계산한 10×1 벡터 d 의 값을 입력받아 최종 해인 10×1 벡터 x , 즉 필터 계수 $C_0 \sim C_9$ 를 계산한다. 제안하는 구조는 10×10 행렬 U 를 계산하는 Factorization_top 모듈과 10×1 벡터 d 를 계산하는

Forward_top 모듈 간의 출레스키 분해법을 이용한 가우시안 제거 알고리즘의 특징적인 연산 관계에 따라 10×10 행렬 U 의 n 번째 행의 계산이 완료되면, 10×1 벡터 d 의 $n-1$ 번째 행을 계산할 수 있다. 따라서 Factorization_top 모듈과 Forward_top 모듈은 그림 5와 같이 2단 파이프라인 구조로 설계하여 병렬적으로 동작하도록 구현함으로써, 수행 사이클을 감소시켰다.

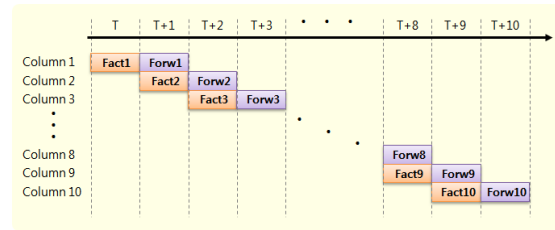


그림 5. 2단 파이프라인 구조
Fig. 5 2-step pipelined architecture

제안하는 하드웨어 구조에서 첫 번째로 연산을 수행하는 Factorization_top 모듈은 그림 6과 같이 Root_top, divider_array, Mul_sub 모듈로 구성된다.

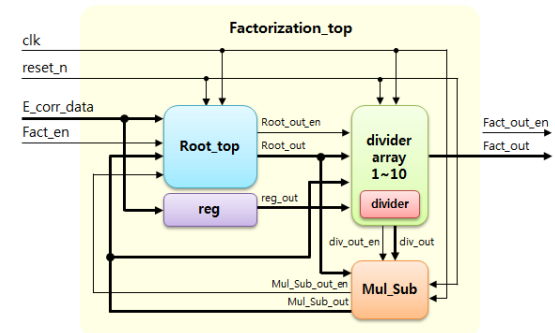


그림 6. Factorization_top 모듈 하드웨어 구조
Fig. 6 Hardware architecture of the Factorization_top module

Factorization_top 모듈은 식 4, 5를 이용하여 10×10 행렬 U 를 계산하고, 각 행에 대한 루트 연산 결과 값은 Root_out 신호로, 10×10 행렬 U 의 행렬 요소들을 Fact_out 신호로 출력한다. Factorization_top 모듈은 상관관계 정보인 E_corr_data 신호를 입력받아 10×10 행렬 U 를 계산하고, 루트 연산을 수행하는 Root_top 모듈은 상관관계 정보에 따라 최대 32비트까지 연산 가능하며 루

트 연산의 결과 값을 출력하기까지 15 사이클을 소모한다. 루트 연산의 결과 값은 divider_array 모듈에서 나눗셈 연산을 수행할 때 나눗셈의 분모 역할을 수행하고, 나눗셈 연산을 수행한 결과 값은 각각의 레지스터 배열에 저장된다. Mul_sub 모듈은 곱셈기와 뺄셈기 등의 연산기로 구성되어 있고, 결과 값은 Root_top 모듈의 입력으로 사용되며 divider_array 모듈에서 나눗셈 연산을 수행할 때 나눗셈의 분자 역할을 수행한다.

그림 6에서 Root_top 모듈은 루트 연산을 수행하는 모듈로써, 입력 값의 비트 수가 적을 경우 LUT(Look up table)을 이용하여 비교적 간단한 구조로 설계할 수 있다. 그러나 ALF는 필터 계수를 추출하기 위한 입력 정보가 최대 32비트에 해당하는 상당히 큰 값이기 때문에, LUT를 이용하여 설계할 경우 하드웨어 구조가 복잡해지고 면적과 연산량이 증가하는 단점이 있다. 따라서 루트 연산을 효율적으로 계산하기 위해 하드웨어 설계에 적합한 알고리즘을 분석하여 복잡도와 면적, 연산 수행시간을 감소시킨 루트 연산 하드웨어 구조를 제안한다. 그림 7은 하드웨어 설계를 위한 루트 연산 알고리즘을 나타낸 것이다[8]. 그림 7에서는 '93'이라는 값을 샘플로 루트 연산 알고리즘을 이용한 연산 과정을 나타낸 것으로, 루트 연산 알고리즘을 이용한 결과 값과 실제 결과 값의 차이는 소수점 부분에서 미세하게 존재하지만 실제 사용되는 값은 정수 부분만을 사용하기 때문에 결과 값에는 영향을 미치지 않는다.

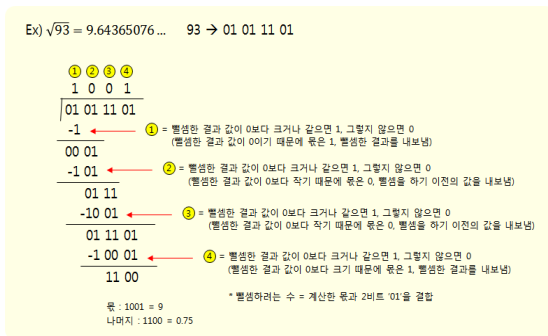


그림 7. 하드웨어 설계를 위한 루트 연산 알고리즘
 Fig. 7 Square root operation algorithm for hardware design

그림 8은 그림 7의 루트 연산 알고리즘을 멀티플렉서와 뺄셈기, 비교기만을 이용한 루트 연산 하드웨어 구조를 나타낸 것이다.

그림 8에서 Root_top 모듈은 Comparator2~23, Sub_decision 모듈로 구성된다. Comparator2~23 모듈은 Root_in 입력 신호와 Sub_decision 모듈의 결과 값인 Sub_out 신호에 2비트 '01'을 오른쪽에 결합한 값과 비교하는 기능을 수행한다. 결과 값인 Comp_out 신호는 레지스터에 저장되고, 입력 신호 Root_in의 마지막인 LSB(Least Significant Bit)까지 연산을 완료하면 레지스터에 저장했던 결과 값을 모두 결합하여 Root_out 신호로 출력한다.

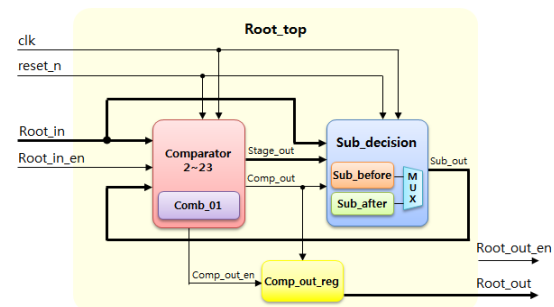


그림 8. 루트 연산 하드웨어 구조
 Fig. 8 Hardware architecture of square root operation

제안하는 하드웨어 구조에서 첫 번째로 동작하는 Factorization_top 모듈 간의 파이프라인 구조로써 병렬적으로 동작하는 Forward_top 모듈은 그림 9와 같이 divider_array, Mul_sub 모듈로 구성된다.

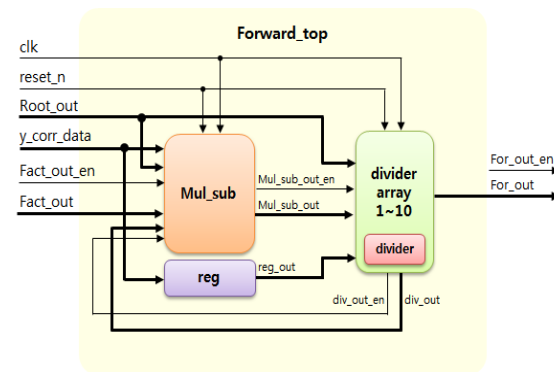


그림 9. Forward_top 모듈 하드웨어 구조
 Fig. 9 Hardware architecture of the Forward_top module

Forward_top 모듈은 상관관계 정보인 y_corr_data 신호와 Factorization_top 모듈에서 계산한 10×10 행렬

UT 결과 값을 입력 받아 10×1 벡터 d의 결과를 For_out 신호로 출력한다. 내부의 Mul_sub 모듈은 곱셈기와 뺄셈기의 연산기로 구성되고, divider_array 모듈에서 나눗셈 연산을 수행할 때 나눗셈의 분자 역할을 수행한다.

제안하는 하드웨어 구조의 Back_top 모듈은 그림 10과 같이 divider_array, Mul_sub, Shift_Round_top 모듈로 구성된다. Back_top 모듈은 Factorization_top 모듈에서 계산한 10×10 행렬 U 결과 값과 Forward_top 모듈에서 계산한 10×1 벡터 d 결과 값을 입력 받아 10×1 벡터 x의 결과를 Back_out 신호로 출력한다. 내부의 Mul_sub 모듈은 곱셈기와 뺄셈기 등의 연산기로 구성되어 있고, divider_array 모듈에서 나눗셈 연산을 수행할 때 나눗셈의 분자 역할을 수행한다. 또한, Back_top 모듈 내부의 Shift_Round_top 모듈은 divider_array 모듈에서 계산한 나머지 값을 쉬프트와 반올림 연산을 수행하여 최종적인 필터 계수 값을 생성한다.

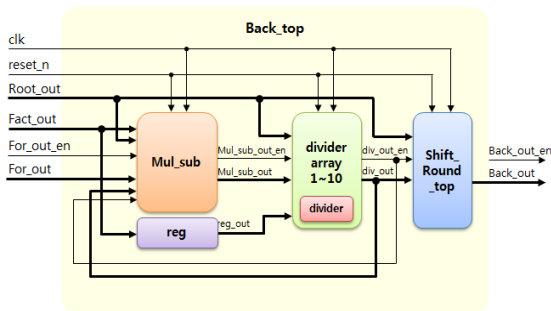


그림 10. Back_top 모듈 하드웨어 구조
Fig. 10 Hardware architecture of the Back_top module

IV. 검증 및 결과

제안하는 ALF 하드웨어 구조는 Verilog HDL로 설계하였고, 시뮬레이션은 Modelsim SE-64 10.1c를 이용하여 검증하였다. 제안하는 ALF 하드웨어를 검증하기 위해 HEVC 표준 참조 소프트웨어 HM 7.0[4]에서 ALF를 통과하기 전후의 영상과 필터 계수 값을 추출하였고, ALF 하드웨어 구조의 시뮬레이션 결과를 추출하여 비교하였다. 검증 결과 제안하는 구조의 출력 데이터와 HM 7.0에서 추출한 데이터가 일치함을 확인하였다.

그림 11은 ALF_top 모듈을 시뮬레이션 한 결과이다. 제안하는 하드웨어 구조는 최대 32비트에 해당하는 상

관관계 정보를 입력받아 1657의 사이클을 수행한 후 10개의 필터 계수 정보를 Coeff_out0~9로 전달한다.

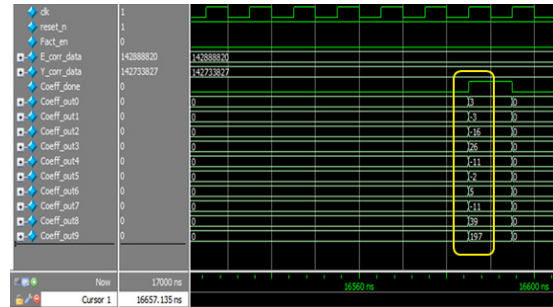


그림 11. ALF_top 모듈 시뮬레이션 결과
Fig. 11 Simulation Result of the ALF_top module

제안하는 하드웨어 구조는 IDEC에서 지원하는 Xilinx ISE 14.3 Vertex-7 XC7VCX485T FPGA 디바이스를 사용하여 합성하였으며, 나눗셈 모듈은 Xilinx Core Generator에서 지원하는 IP를 사용하였다[9]. 표 1은 제안하는 ALF 하드웨어의 합성 결과를 나타낸다.

표 1. 제안하는 ALF 하드웨어 합성 결과
Table. 1 Synthesis Results of Proposed ALF Hardware

구분	결과
동작 주파수	186 MHz
사이클 수	1657
슬라이스 레지스터 수	519529(85%)
슬라이스 LUT 수	229782(75%)

V. 결론

본 논문에서는 필터 계수를 효율적으로 추출하기 위한 고성능 HEVC 부호화기의 ALF 하드웨어 설계를 위해 출레스키 분해 알고리즘의 특징적인 연산 관계를 분석하여 파이프라인 구조로 설계함으로써 수행 사이클 수를 34.4% 감소시켰다. 또한, 루트 연산 알고리즘을 분석하여 멀티플렉서와 뺄셈기, 비교기만을 이용한 하드웨어 구조로 설계함으로써 기존 루트 연산 IP[10]에 비해 LUT는 65.7%, LUT-FF는 70.1% 감소시켰으며 수행 사이클 수는 53.1% 감소시켰다. 제안하는 하드웨어 구조는 Xilinx ISE 14.3 Vertex-7 XC7VCX485T FPGA에서 합성한 결과 최대 동작 주파수는 186MHz이고, 총

1657 사이클이 소요된다. 또한, 4K UHD(3840×2160) 영상을 초당 40 프레임으로 실시간 처리할 수 있다.

향후 연구 과제로는 하드웨어 면적이 상당히 크다는 단점을 보완하기 위해 면적을 최적화시킨 하드웨어 구조를 설계할 것이다. 또한, 나눗셈 연산 IP[9]를 사용하지 않고 나눗셈 연산에 최적화된 나눗셈 연산 모듈을 설계하여 ASIC 설계 및 검증을 진행할 것이다.

감사의 글

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

REFERENCES

- [1] G. J. Sullivan, et-al., “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,” *Integrated Circuits and Systems Design (SBCCI)*, Aug. 2012.
- [5] “HEVC Test Model 7 Encoder Description,” Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG, JCTVC-I1002, May 2012.
- [6] C. Y. Tasi, et-al., “Adaptive Loop Filtering for Video Coding,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 934-945, Dec. 2013.
- [7] L. B. Richard, and J. D. Faires, *Numerical Analysis*, 9th ed, 2010, pp. 357-430.
- [8] 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.
- [9] Xilinx Logicore, “Divider Generator v4.0,”(<http://www.xilinx.com>, Jun. 2011)
- [10] Xilinx Logicore, “IP CORDIC v5.0,”(<http://www.xilinx.com>, Jun. 2011)



신승용(Seungyong Shin)

2013년 한밭대학교 공과대학 정보통신공학과 공학사
 2013년~현재 한밭대학교 정보통신전문대학원 정보통신공학과 석사과정
 ※관심분야 : SoC 플랫폼 설계 및 검증, 멀티미디어 코덱 설계



류광기(Kwangki Ryoo)

2000년 한양대학교 대학원 전자공학과 공학박사
 1991년~1994년 육군사관학교 교수부 전자공학과 전임강사
 2000년~2002년 ETRI 시스템C설계팀 선임연구원
 2010년~2011년 Univ of Texas at Dallas 방문교수
 2003년~현재 한밭대학교 공과대학 정보통신공학과 교수
 ※관심분야 : 공학교육, SoC 플랫폼 설계 및 검증, 멀티미디어 코덱 설계