

논문 2009-46SP-3-5

JPEG2000을 위한 효율적인 EBCOT의 VLSI 설계 및 구현

(A VLSI Efficient Design and Implementation of EBCOT for JPEG2000)

양 상 훈*, 유 혁 민*, 박 동 선**, 윤 숙***

(Sanghoon Yang, Hyuck Min Yoo, Dong Sun Park, and Sook Yoon)

요 약

차세대 정지영상 압축방식인 JPEG2000은 DWT와 EBCOT로 구성 되어 있다. EBCOT는 컨텍스트 추출부(BPC)와 산술부호화기(AC)로 구성되는데 본 논문에서는 효율적인 EBCOT 설계에 새로운 알고리즘을 적용하여 설계하였다. BPC(Bit Plane Coding)는 context 기반의 부호화기를 사용하였고, 현재의 SigStage register의 값과 상위 비트 플랜의 column 값을 가공한 데이터와 현재의 column 값을 이용하여 코딩패스를 미리 예측하는 기법을 사용하였다. BAC(Binary Arithmetic Coder)에는 4단계 pipeline을 적용하였다. 설계된 EBCOT은 Verilog HDL 모델링후 Xilinx FPGA technology를 이용하여 합성한 후 동작을 검증하였다.

Abstract

The new still image compression standard JPEG2000 is consisted of DWT and EBCOT. In this paper, proposed and designed new algorithm in efficient EBCOT. BPC based on the context. Proposed BPC Algorithm is forecasted coding pass using SigStage, column, mpass value. BAC design apply 4-pipeline stage. EBCOT designed using Verilog HDL. Verification and Synthesis using Xilinx FPGA technology.)

Keywords : JPEG2000, EBCOT, BPC, BAC

I. 서 론

현재 가장 널리 사용되고 있는 정지영상 압축방법의 표준인 JPEG은 여러 가지 장점을 가지고 있지만, 저화질 비트에서의 화질열화와 고압축에서의 블록화 현상 등의 치명적인 단점을 가지고 있다. 이런 단점을 보완할 수 있는 새로운 정지영상 압축방법 표준이 요구되었고, 차세대 정지영상 압축부호화 방식으로서 JPEG2000의 표준화가 완료되었다^[1~3].

JPEG2000은 기존의 정지 영상 압축의 단점을 보완하

였고 동시에 보다 높은 압축 성능을 보인다. 뿐만 아니라 JPEG에서 제공하지 못했던 관심영역(ROI) 부호화 등의 여러 가지 새로운 특징들을 제공한다.

JPEG과 JPEG2000의 구조상의 차이점은 그림 1 과 같다. JPEG은 압축 프로세서로 DCT(Discrete cosine transform)와 허프만 부호화(Huffman coding)를 사용하지만, JPEG2000 에서는 DWT(Discrete Wavelet Transform)와 산술부호화(EBCOT)를 사용하여 영상의 압축을 수행한다.

JPEG2000은 전처리과정에서 전체 영상이미지를 타일(tile)화한 후, 각각의 타일을 이산 웨이블릿 변환(Discrete Wavelet transform)한 다음에 양자화(Quantization)를 진행 한다. 웨이블릿 계수로 구성된 양자화된 서브밴드들을 주어진 크기의 코드블록(code block)으로 나눈 후 각각의 코드블록에 대해서 독립적으로 이진 산술부호화를 수행한다.

Li 등은 JPEG2000의 서브모듈인 BPC(Bit Plane Coding)을 설계하기 위한 부분병렬 알고리즘과

* 학생회원, ** 정회원, 전북대학교 전자정보공학부
(Div. of Electronic & Information Engineering
Chonbuk National Univ.)

*** 정회원, 목포대학교 정보공학부
(School of Information Engineering Mokpo National
Univ.)

※ 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임.

접수일자: 2008년9월13일, 수정완료일: 2009년4월13일

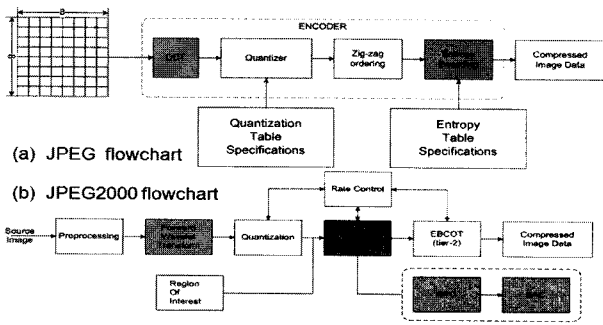


그림 1. JPEG와 JPEG2000 흐름도
Fig. 1. JPEG and JPEG2000 Flow chart.

BAC(Binary Arithmetic Codr)의 4단계 pipeline을 제안하였다^[4-5]. 이 알고리즘은 BPC 알고리즘에 NC Generator를 통해서 컨텍스트 추출에 필요한 주변의 정보를 계산하여 3개의 pass에 대해서 동시에 컨텍스트를 추출하는 방법과 실질적인 압축을 진행하는 BAC의 동작 속도를 높이기 위해서 4단계 파이프라인을 적용하는 것을 소개하였다.

JPEG2000에서 현재 bit를 엔트로피 코딩을 하기 위해서는 주위의 sigma, sigma delay, visiting bit, 등의 주변상태정보가 필요하다. 본 논문에서는 Li 등이 제안한 알고리즘을 기반으로 하여 각각의 상태 정보의 예측을 통한 coding pass와 coding operation을 결정하여 현재 비트의 컨텍스트를 추출하는 알고리즘을 제안한다. EBCOT의 전체적인 성능을 높이기 위해서 4단계 파이프라인을 적용한 BAC를 설계하였다.

본 논문의 구성은 II장에서는 Context 기반을 BPC(Bit Plane Coding)의 구조와 parallel 개념의 BPC, 4단계 파이프라인의 BAC에 대한 설명과 제안된 알고리즘에 대해서 설명하였다. III장에서는 제안된 알고리즘을 바탕으로 설계된 BPC와 BAC의 실험 결과를 IV장에서는 결론을 맺는다.

II. Cotext 기반의 Bit Plane Coding

BPC는 DWT의 계수 값들이 양자화된 영상이미지를 각각의 subband별 code-block 내의 데이터를 bit-plane으로 펼쳐 놓은 상태에서 진행한다. 현재 bit의 sample, Significance bit(σ), Visited one bit(ν), Magnitude refinemnet coded bit(μ), Sign bit(β)들과 같은 주변 상태 정보를 이용하여, 그림 2의 순서도에 따라 3가지 coding pass(SPP, MRP, CUP)로 그룹화되어 진다.

코딩 패스별로 그룹화는 BPC 다음 단계 수행될 산

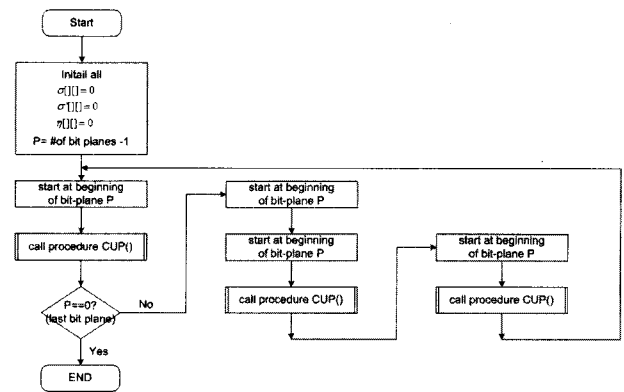


그림 2. BPC 알고리즘 흐름도
Fig. 2. Bit Plane Coding Flow chart.

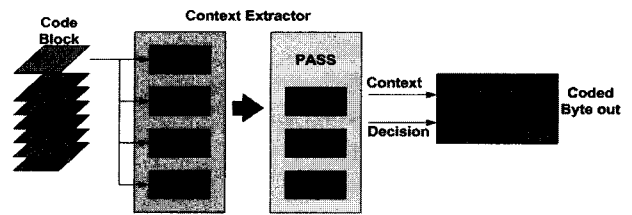


그림 3. 컨텍스트 추출기 블록도
Fig. 3. Context EXtractor Block Diagram.

술부호화 과정의 데이터량을 최대한 줄일 수 있다. 각각의 bit는 4가지 coding operations(Zero Coding, Sign Coding, Magnitude Refinement Coding, Run Length Coding)을 통하여 Context와 Decision을 출력하게 된다.

Li 등이 제시한 Bit Plane Coding은 3가지 병렬 처리 알고리즘이 적용 되었다. 첫 번째는 각각의 Bit-Plane 개별적으로 수행되었던 산술 부호화를 병렬 처리를 통해서 동시에 진행하는 것이다. 두 번째는 각 서브밴드 별로 3가지 코딩 패스를 동시에 스캔하여 그룹화 하였다. 각 strip별로 4개의 sample을 병렬처리 기법을 이용하여 동시에 처리하여 컨텍스트와 디시전 값을 추출하도록 하고 있다. BAC는 MQ-Coder에 기반을 둔 이진 산술부호화기로서 복잡한 곱셈과 나눗셈 연산을 간단한 덧셈과 시프트 연산으로 근사화 하여 구현하였다. 전체적인 동작 속도를 높이기 위해서 4 단계 파이프 라인을 적용하였다.

2.1. 상태 정보 예측 기법을 적용한 BPC

BPC는 3가지 병렬 처리 알고리즘인 3 Parallelism (Bit-Plane, coding pass, strip)이 적용되었다. Load Logic의 기능을 강화하여 Coding Operation을 수행하기 위한 상태 정보 비트의 Register를 감소시킬 수 있었고,

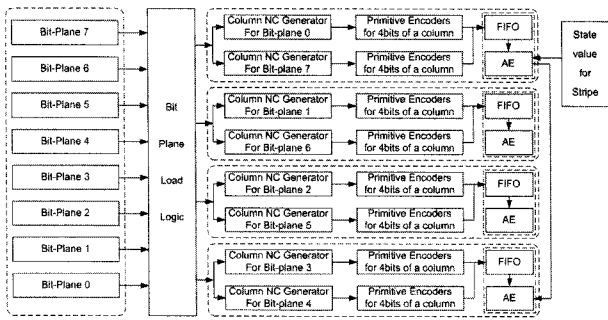


그림 4. EBCOT 전체 블록도
Fig. 4. EBCOT Top Module Design.

이로 인해 기존의 구조보다 면적이 감소하였다. 또한 Register의 Access time을 줄임으로써 동작속도가 향상되었다. 그림 5는 EBCOT의 Top module 블록 다이어그램이다.

BPC는 각 Bit-Plane별로 Coding 이루어지며, 하나의 strip이 즉 4개의 Column Sample이 하나의 clock Cycle 처리되어 FIFO를 통해서 산술 부호화기로 연결되어 진다. BPC는 위의 구조에서 알 수 있듯이 Load Logic, NC Generator, Primitive Encoder로 구성 되어 진다.

가. Bit Plane Coding

(1) 예측 기법을 적용하기 위한 Load Logic

양자화된 DWT의 계수 값을 Bit Plane 형태로 풀어 주고, 이를 strip 단위로 출력하는데, 이때 각 bit plane의 sample들을 이용하여 Significance bit(σ), Visited one bit(v), Magnitude refinement coded bit(μ)등의 값을 예측하여 보내며, 이후 NC Generator 블록에서 σ 값을 미리 연산하고 update 함으로써 Coding operation을 신속하게 정하도록 해주는 정보를 제공하는 블록이다. 그림 5는 본 논문에서 제시한 Load Logic의 블록도이다.

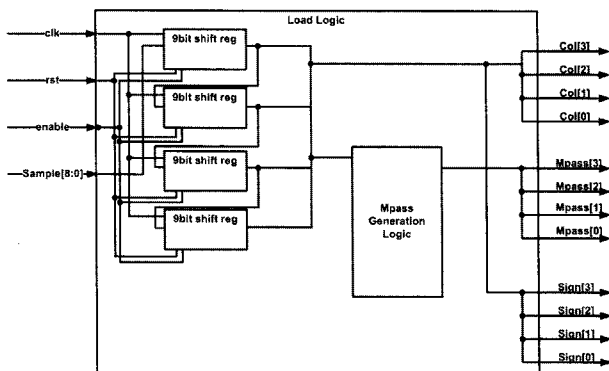


그림 5. Load Logic H/W 블록도
Fig. 5. Load Logic Design Block diagram,

(2) NC(Neighbor Contribution) Generator

JPEG2000에서는 자기의 픽셀 정보를 주변의 값들과의 관계를 이용하여 표현하고 있다. Load Logic을 통해서 넘어오는 정보들을 조합하여 Significant State값을 미리 예측함과 동시에 update 하여 Primitive Encoder에서 각 sample들의 coding pass와 coding operation을 선택 제공한다. Coding Pass 와 Coding Operation을 결정하기 위해서는 현재 자신의 상태 정보, 방문정보, sign bit, 자신의 sample value 뿐만 아니라 자신의 주변(8개 sample) sample의 정보 또한 알고 있어야 가능하다.

본 논문에서 사용하는 구조에서는 상태 정보와 column 값을 기반으로 하여 코딩 패스를 selection 하도록 설계되었으며, 코딩 패스를 통해서 이루어지는 현재 sample의 상태정보를 곧바로 update 하도록 설계하였다.

그림 6은 설계된 NC Gen의 블록다이어그램이다.

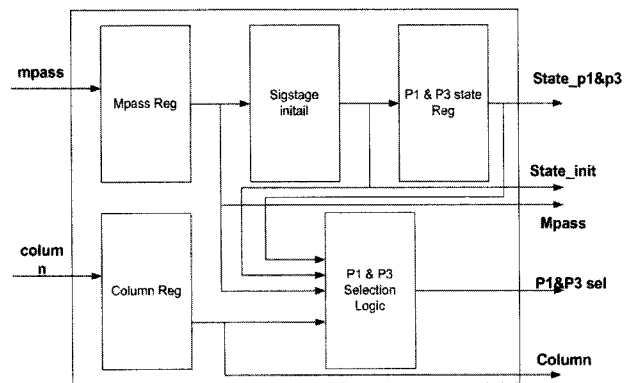


그림 6. NC Generator 블록도
Fig. 6. Neighbor Contribution Generator Block Diagram.

(3) Primitive Encoder

DWT의 계수 값을 통해서 필요한 정보를 수집한 후에 실제적인 context와 decision을 추출하는 블록이다. NC Gen을 통해서 수집된 상태정보 들과, sample, coding pass selecting, coding operation selecting 등의 정보를 받아 들어 각각의 sample에 대한 콘텍스와 디시전을 추출한다.

나. Binary Arithmetic Coder

다음 그림은 각 샘플간의 상관관계를 이용해 추출된 컨텍스트를 이용하여 적응적으로 확률을 추정하는 방법을 나타내는 블록이다.

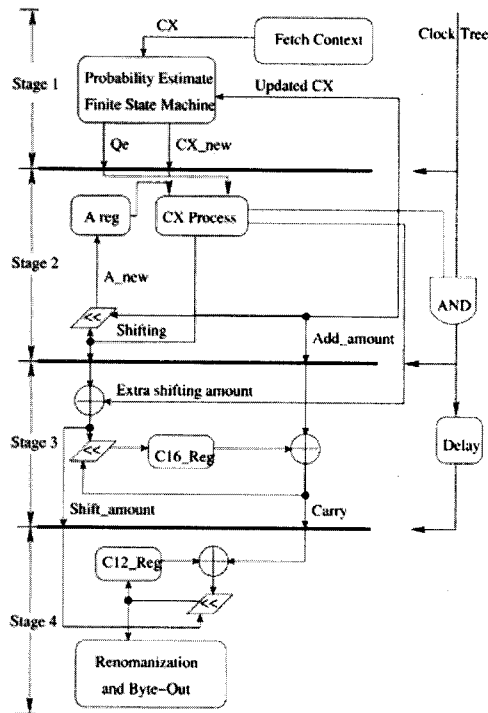


그림 7. 4단계 파이프라인 MQ-Coder 블록도
Fig. 7. 4-stage pipelined MQ-Coder Block Diagram

부호화는 현재의 부호화 구간을 2개의 부구간 즉, LPS (Less Probable Symbol)에 대한 구간과 MPS (More Probable Symbol)에 대한 구간으로 분할하는 절차를 반복적으로 수행하여 누적확률 분포를 산술적으로 계산하여 코드워드를 생성한다.

AC는 47개의 각기 다른 인덱스(Index, I)에 대한 확률 예측값(Qe(I(CX)))을 제공하는 확률예측기, 구간 분할 연산과 실제적인 압축코드 생성을 담당하는 Context Processor, 그리고 컨트롤러로써 구성된다. 본 논문에서 사용되어지는 AC는 확률 예측 구간, CX Processing, C update, Byte-out으로 4단계 Pipe-line을 하여 동작 속도를 높이도록 하였다.

(1) 확률예측기

확률예측의 정확도에 따라 AC의 압축률과 동작속도가 결정된다. 확률예측기 블록은 2개의 LUT (Lookup Table)로 구성된다. 각각의 컨텍스트와 연관된 인덱스와 MPS 값을 출력하는 Index LUT와 인덱스에 대한 확률예측값인 Qe, NMPS, NLPS, SWI 값을 출력하는 Qe LUT로 이루어져 있다. 표 1은 입력된 컨텍스트에 따른 초기 인덱스 와 MPS 값이다.

그림 8은 확률예측 순서도로서 입력된 심볼값에 대

표 1. 컨텍스트에 대한 초기 인덱스와 MPS값
Table 1. Context and MPS initial value.

Context	index	MPS
0	3	0
1	4	0
2 ~ 17	0	0
18	46	0

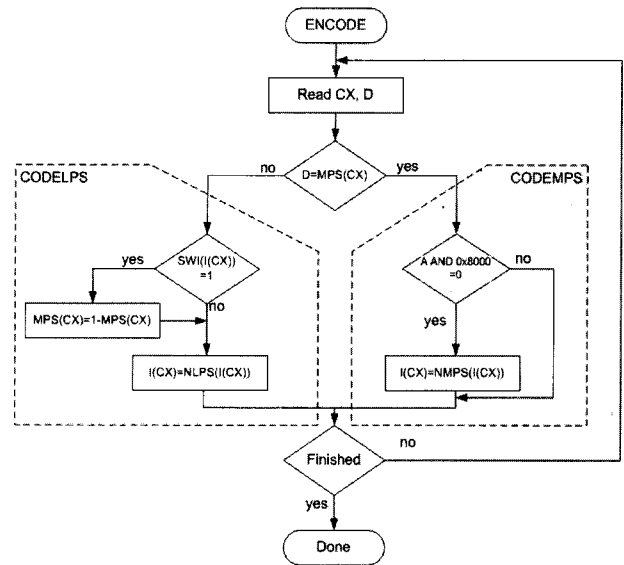


그림 8. 확률예측 순서도
Fig. 8. Probability Estimation Flow chart.

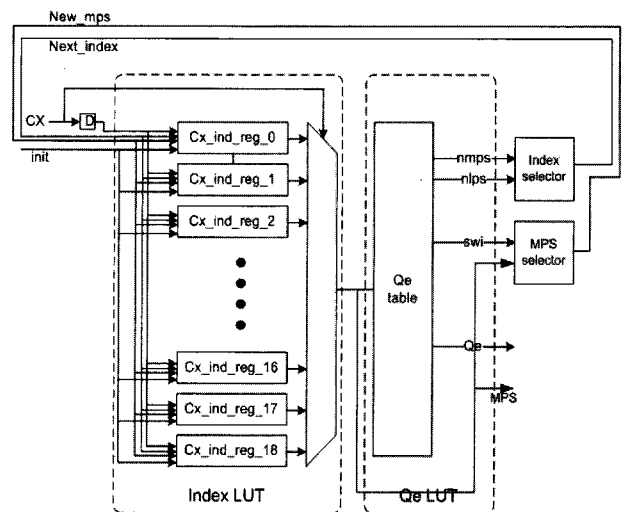


그림 9. 확률 예측기 블록도
Fig. 9. Probability Estimation H/W Block Diagram.

한 Coding condition을 비교하여 컨텍스트에 대한 인덱스값을 갱신한다.

그림 9는 설계된 확률 예측기의 블록도 이다. 19개의 컨텍스트에 따른 인덱스값에 Qe, nmps, nlps, swi 값을 출력하고 현재 컨텍스트의 Index 와 MPS 값들을 update 하도록 구성 하였다.

(2) Context Processing

반복적인 확률구간 분할과정은 이진 산술부호화의 기본알고리즘 중의 하나이다. 일반적인 알고리즘에서는 구간분할을 위해서 곱셈과 나눗셈 연산을 수행 하지만, 처리시간의 단축과 하드웨어 면적의 감소를 위해서 단순 덧셈기와 barrel shifter만을 이용하여 구간 분할 연산과 update가 이뤄지도록 근사화 하는 것을 제안하였고, 이를 바탕으로 구현하였다.

그림 10은 Context Processor의 블록도이다. 이는 Qe LUT의 값들(Qe, nmpps, nlps, swi, mps)을 이용하여 Index, MPS 값들을 연산 한 후 이전 Stage로 값을 통해서 update가 이뤄지도록 한다. A_reg값은 barrel shift를 이용하여 update한다. 이는 A(현재 확률 구간의 크기)와 C(codeword)의 재정규화 작업을 1 Clock Cycle에 진행하여 처리 속도를 단축 하였다.

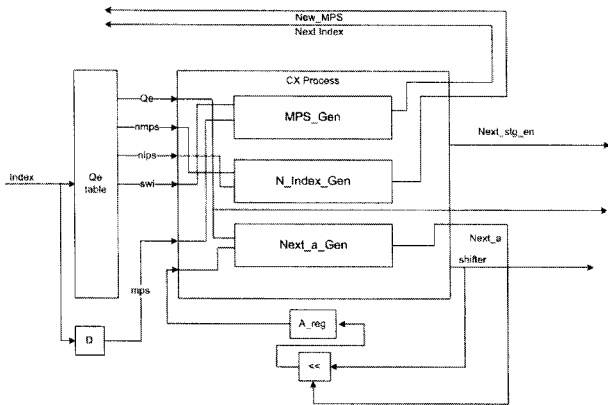


그림 10. CX_Processor 블록도
Fig. 10. Context Processor H/W Block Diagram.

(3) C_register Calculation & Byte-out

Pipe-line stage는 4단계를 이용하였고, Index update, Cx_process, C_reg 연산, Byte out 구간으로 나누어서 설계하였다.

stage 1에서는 각각의 컨텍스트에 따른 인덱스를 통해서 Qe 값과 MPS 값들을 stage 2로 넘겨주며, stage 2에서 연산된 인덱스값을 통해서 현재의 컨텍스트의 인덱스를 update한다.

stage 2에서는 stage 1의 컨텍스트의 새로운 인덱스 값과, mps 값을 결정하여 stage 1으로 보내준다. 또한 A, Qe, mps 값을 이용하여 재정규화시 필요한 barrel shift의 shifting bits와 Coding Operation을 결정한다.

stage 3와 stage 4에서는 C의 재정규화 와 Byte out 이이루어진다. C는 총 28bit구성된다. 이것을 동작 속도

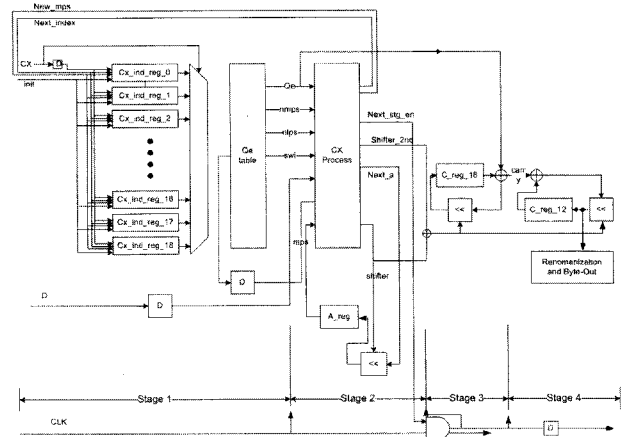


그림 11. 산술 부호화기 전체 블록도
Fig. 11. Arithmetic Coding H/W Top Block Design.

의 향상과 구조의 단순화를 위해서 16bit와 12bit로 분할하였다. 하위 16bit는 단순 덧셈(C+Qe)과 shifting (Lshift) 연산만을 수행한다.

상위 12bit는 덧셈(C+carry)과 shifting(Lshift) 연산을 수행하지만 C값에 따라 MQ-Coder의 실제적인 압축이 일어나는 Byte-out 혹은 Maker Code 인 FLUSH 등을 수행한다. 다음 그림 11은 설계된 BAC의 블록 다이어그램이다.

III. 실험

설계된 예측 기법을 적용한 BPC는 하드웨어 기술 언어인 Verilog HDL로 모델링하였다. 설계된 하드웨어 IP를 검증하기 위하여, Xilinx FPGA Device (XC5VLX110)를 이용하였다. Xilinx의 라이브러리를 이용하여 Xilinx FPGA의 디자인 툴인 ISE(Ver 9.2i)의 XST를 사용하여 Compile 과 합성을 진행하였다. 기본적인 동작 검증을 위해서 Maxplus, ModelSim XE를 이용해서 Simulation을 진행하였다. 합성 결과 BPC의 기본 unit의 크기는 (메모리를 포함한 전체 게이트 수) 29000여 system gate가 사용되었고, BAC의 면적은 17280 system gate가 사용되었다.

설계된 H/W는 ISE의 Synthesis 결과 각 Unit의 최대 동작 주파수는 BPC : 163MHz, BAC : 142MHz가 되었다. 각 Unit 별 최대 동작 주파수를 사용할 수 있으나, 시스템의 안정도와 DWT와의 동작 주파수를 고려하여 PLDA사의 Xpress LXT 110을 통하여100MHz로 동작을 검증하였다. 그림 12는 BPC와 BAC의 Simulation Result이다.

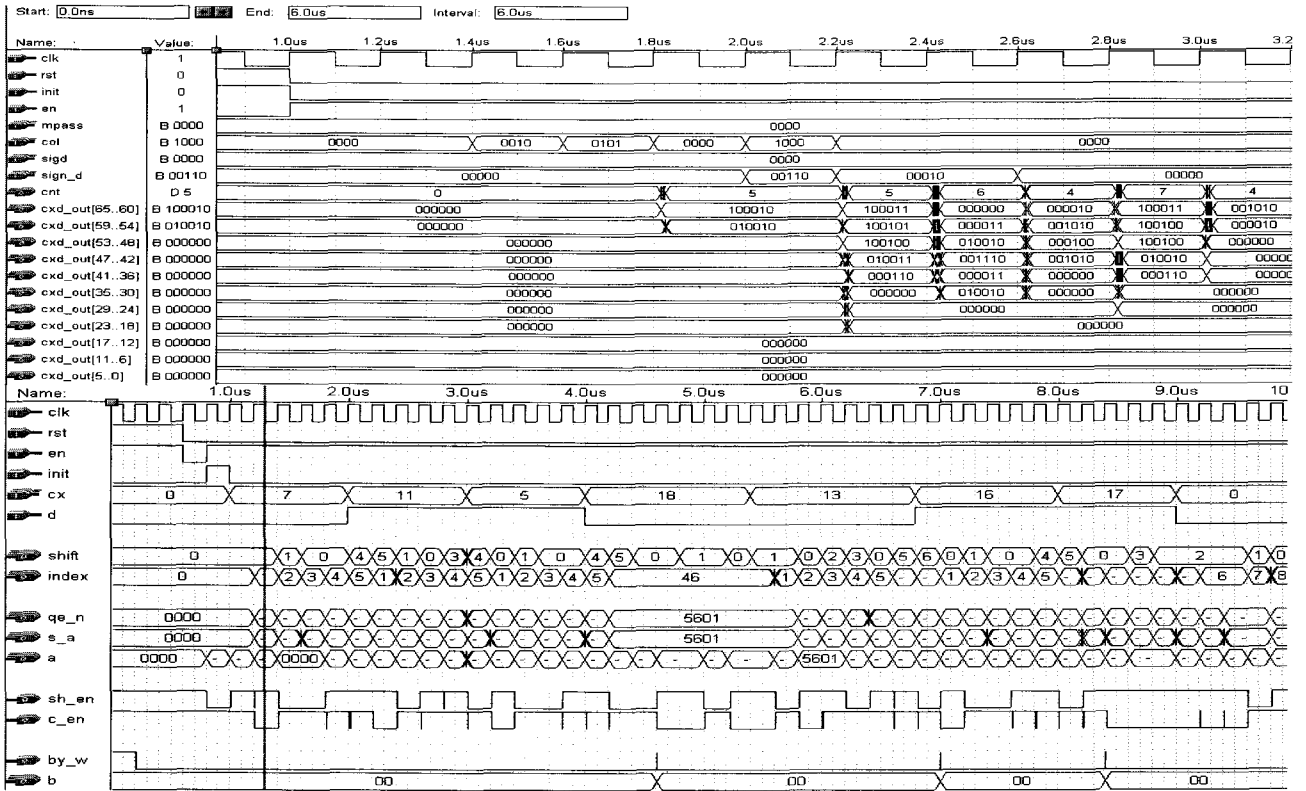


그림 12. 산술 부호화기 전체 시뮬레이션 결과
 Fig. 12. AC Top Block Simulation Result.

IV. 결론

본 논문에 앞서 Li등이 제안한 3가지의 병렬 처리 알고리즘은 JPEG2000의 병목 현상지인 BPC의 상태 정보를 미리 예측함으로써 처리 속도를 높이고자 하였다. 결과적으로 상태 정보를 미리 예측함으로써 각 bit plane에서 Context를 추출하기 위해서 소비 되는 메모리 액세스 시간과 Latency를 감소시킴으로써 전체적인 성능 향상을 이뤄낼 수 있었다.

또한, 하드웨어의 동작 속도를 고려한 4단계 파이프라인을 적용한 MQ-Coder를 제안하고 설계하였다. 산술부호화기는 곱셈기, 나눗셈기, 뺄셈, 덧셈으로 구성되어 있으나 이를 근사화 하여 단순 덧셈기와 시프트기를 이용하여 구조를 단순화 하는 방식을 채택하였다. 이는 구조의 단순화 뿐만아니라, 연산 시간의 단축, 하드웨어 설계 및 구현의 단순화, 전체 면적의 감소 등의 효과를 얻을 수 있었다.

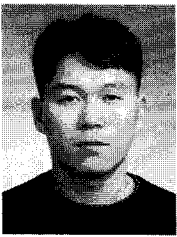
Xilinx FPGA technology를 통해서 확인 한 결과 두 모듈간의 결합 및 시뮬레이션 상의 전체적인 동작 주파수를 140MHz 대역까지 가능하나 시스템의 안정도와

DWT와의 동작 속도를 고려하여 100MHz에서 성능을 확인하였다.

참고 문헌

- [1] ISO/IEC 15444-1, "Information technology JPEG2000 image coding system-Part 1: Core coding system", 2001.
- [2] M. Boliek, C. Christopoulos, and E. Majani, JPEG2000 Part I: Final Publication Draft, ISO/IEC JTC1/SC29/WG1 N2678, Jul. 2002.
- [3] K.-K. Ong, W.-H. Chang, Y.-C. Tseng, Y.-S. Lee, and C.-Y. Lee, "A high throughput low cost context-based adaptive arithmetic codec for multiple standards," in Proc. IEEE Int. Conf. Image Process., vol. 1, pp. 872 - 875. Sep. 2002.
- [4] Y. Li, M. Elgamel, and M. Bayoumi, "A partial parallel algorithm and architecture for arithmetic encoder in JPEG 2000," in Proc. IEEE Int. Symp. Circuits Syst., pp. 5198 - 5201, May 2005.
- [5] Y. Li, M. Elgamel, and M. Bayoumi, "A Three-Level Parallel High-Speed Low-Power Architecture for EBCOT of JPEG2000," in Proc. IEEE Int. Symp. Circuits Syst., pp. 1153 - 1163, Sep 2006.

저 자 소 개



양 상 훈(학생회원)
 2005년 8월 전북대학교 전자정보
 공학부(전자전공) 학사
 2007년 8월 전북대학교 대학원
 정보통신 공학 석사
 2008년 3월~현재 전북대학교
 전자정보공학부 전자전공
 박사과정

<주관심분야 : 통신, 신호처리, image processing,
 임베디드 시스템, SoC>



유 혁 민(학생회원)
 2005년 3월 전북대학교 전자정보
 공학부(전자전공) 학사
 2007년 3월 전북대학교 대학원
 정보통신 공학 석사
 2007년 3월 ~현재 전북대학교
 전자정보공학부 전자전공
 박사과정

<주관심분야 : 통신, 신호처리, image processing,
 임베디드 시스템, SoC>



박 동 선(정회원)
 1979년 2월 고려대학교 전기전자
 공학과 졸업
 1984년 Missouri-Columbia 공학
 석사
 1991년 Missouri-Columbia 공학
 박사

1991년 3월~현재 전북대학교 전자정보공학부
 교수

<주관심분야 : 신경망, 패턴인식, 영상처리, 디지
 털 시스템설계>



윤 숙(정회원)-교신저자
 1993년 2월 전북대학교 정보통신
 공학과 학사 졸업.
 1996년 2월 전북대학교 정보통신
 공학과 공학석사
 2003년 2월 전북대학교
 전자공학과 공학박사

2006 9월~현재 목포대학교 정보공학부 조교수

<주관심분야 : 멀티미디어컴퓨팅, 신호처리>