JPEG2000 Encoder를 위한 EBCOT Tier-1의 하드웨어 구현

Hardware Implementation of EBCOT TIER-1 for JPEG2000 Encoder

이성목*, 장원우*, 조성대**, 강봉순*
Sung-mok Lee*, Won-woo Jang*, Sung-Dae Cho**, Bong-soon Kang*

요 약

본 논문은 JPEG2000 Encoder를 위한 EBCOT Tier-1의 하드웨어 구현에 관한 것이다. 2000년대 초반, JPEG의 단점을 극복하기 위해 차세대 정지영상 압축 표준으로 등장한 것이 JPEG2000이다. JPEG2000 표준은 DWT(Discrete Wavelet Transform)과 EBCOT Entropy coding 기술을 기반으로 하고 있다. 이 중 EBCOT(Embedded block coding with optimized truncation)은 JPEG2000 표준에서 실제 압축을 수행하는 가장 중요한 기술 중 하나이다. 하지만 EBCOT는 Bit-level 처리를 하기 때문에 JPEG2000 압축 과정 중 절반 정도의 연산 시간을 차지하는 단점을 가지고 있다. 그래서 이에 본 논문은 EBCOT 연산의 효율성을 높이기 위해 수정된 Context 추출 방법과 산술 부호화기 MQ- Coder를 하드웨어 구현하였다. 제안된 시스템은 Verilog-HDL로 구현되었으며 TSMC 0.25um ASIC 라이브러리로 합성한 결과, 게이트 카운트는 30,511개로 구현되었으며, 50MHz의 동작 조건을 만족한다.

Abstract

This paper presents the implementation of a EBCOT TIER-1 for JPEG2000 Encoder. JPEG2000 is new standard for the compression of still image for overcome the artifact of JPEG. JPEG2000 standard is based on DWT(Discrete Wavelet Transform) and EBCOT Entropy coding technology. EBCOT(Embedded block coding with optimized truncation) is the most important technology that is compressed the image data in the JPEG2000. However, EBCOT has the artifact because the operations are bit-level processing and occupy the harf of the computation time of JPEG2000 Compression. Therefore, in this paper, we present modified context extraction method for enhance EBCOT computational efficiency and implemented MQ- Coder as arithmetic coder. The proposed system is implemented by Verilog-HDL, under the condition of TSMC 0.25um ASIC library, gate counts are 30,511EA and satisfied the 50MHz operating condition.

Keywords : JPEG2000, EBCOT, TIER-1, Context, MQ- Coder

I. 서 론

멀티미디어의 발달로 저장매체와 전송선로 대역폭의 한계 등의 문제를 극복하기 위해 영상 압축은 필수적인 것이 되었다. 그래서 표준으로 정해진 것이 동영상 압축 표준인 MPEG과 정지영상 압축 표준인 JPEG이다. JPEG(Joint Photogra phic Expert Group)은 정지영상 압축 표준으로 채택 된 이후 다양한 멀티미디어 응용분야에 사용되고 있다. 그러나 JPEG은 낮은 비트율 환경에서 뚜렷한 성능 열화를 보이고 있으며 DCT(Discrete Cosine Trans form)을 기반으로 압축과정을 수행하기 때문에 블록화 현상과 같은 화질 열화가 발생하게 된다 [1]. 이러한 문제점들을 해결하기 위해 2000년대 초반 DWT (Discrete Wavelet Transform)와 EBCOT(Embedded Block Coding with Optimized Truncation)을 기반으로 한 차세대 정지영상 압축 표준JPEG2000이 새롭게 개발되었다. JPEG2000은 JPEG에서 단점으로 지적되었던 화질 열화를 개선하고 ROI(Region Of Interest)와 같은 다양한 부가기능이 추가된 차세대 정지영상 압축 표준이다. JPEG2000은 JPEG에 비해 훨씬 높은 압축 성능을 보여주지만 연산 과정이 훨씬 복잡하고 연산량이 많은 단점을 가지고 있다. JPEG2000 압축 과정 중 특히 주요 연산 과정인 EBCOT Tier-1 엔트로피 코딩이 연산량의 절반 가량을 차지하고 있다. 이것은 EBCOT Tier-1 코딩이 Bit-level 처리를 하기 때문이다. 이와 같은 문제점 때문에 EBCOT Tier-1 과정의 연산량을 줄여 고속화하는 연구가 많이 진행되고 있다. K. Andra와 H. Chen이 제시한 하드웨어 구조를 대표적인 예로 들 수 있다 [2-3]. K. Andra의 논문에서는 부대역의 병렬 코딩을 통해 고속화를 하였고, H. Chen이 제안한 방법은 행 단위 연산방법을 제안하

* 동아대학교 ** (주)삼성전자
투고 일자 : 2009. 3. 18 수정완료일자 : 2009. 4. 21
게재확정일자 : 2010. 4. 29
* 이 논문은 동아대학교 학술연구비 지원에 의하여 연구되었음.

여 연산 속도를 높였다. 이에 본 논문은 EBCOT Tier-1 코딩 중 가장 높은 연산량을 가지는 Context 추출의 연산 효율성을 높이기 위한 수정된 방법을 제안하고 이를 하드웨어로 구현하고자 한다. 또한 추출된 Context와 Decision값을 사용하여 산술 부호화기인 MQ- Coder를 하드웨어로 구현하여 JPEG2000 Encoder에 적용하고자 한다.

II. JPEG2000 기본 알고리즘

그림 1은 JPEG2000 인코딩 과정을 도식적으로 나타낸 것이다[4].

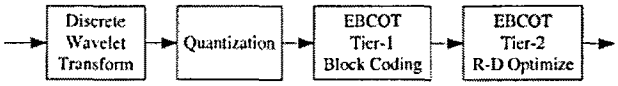

그림 1. JPEG2000 인코딩 과정
Fig. 1. Encoding process of the JPEG2000

JPEG2000의 인코딩 과정을 살펴보면 입력 영상에 DWT를 적용하고 각 부대역 별로 저장한다. 저장된 웨이블릿 계수는 각 부대역 별로 양자화를 거치게 되고 양자화된 데이터는 Tier-1 엔트로피 코딩 과정을 거친다. 최종적으로 Tier-2 과정에서 JPEG2000 압축데이터를 생성하게 된다. 이의 역과정을 수행하게 되면 압축 후 복원된 영상을 얻을 수 있다. JPEG2000 인코딩을 수행할 경우 각 처리 과정의 연산량을 표 1에 나타내었다[2]. 표 1은 컬러 영상을 JPEG2000 인코딩할 경우, 총 연산시간을 각 처리 과정 별로 백분율로 나타낸 것이다. EBCOT Tier-1의 연산량을 살펴보면 무손실 압축인 경우 약 69%, 손실 압축의 경우 약 44%이다. 이는 주요 연산부인 DWT의 약 2배에 달하는 수치이다. 연산 시간 백분율은 영상의 특성과 시뮬레이션 조건에 따라 달라질 수 있으며, 항상 고정적인 비율을 나타내지 않는다.

표 1. JPEG2000 인코딩의 주요 연산 시간 백분율
Table. 1 Run Time Percentage of JPEG2000 Encoding

Operation	Run Time Percentage	
	Loseless	Lossy
Color Transform	0.91	12.12
DWT	11.90	23.97
Quantization	-	5.04
EBCOT Tier-1	69.29	43.85
-Pass 1	13.90	12.39
-Pass 2	10.94	5.63
-Pass 3	25.12	13.77
-A.E	19.33	12.06
EBCOT Tier-2	17.90	13.01

EBCOT Tier-1은 양자화된 DWT 계수의 분할 단위인 코드 블록 별로 연산을 수행한다. 코드 블록의 크기는 최소 8×8에서 최대 64×64 크기로 표준에서 지정되어 있다. 이것은 코드 블록 별로 인코딩과 디코딩 과정을 수행함으로써 블록 간 간섭이 발생하지 않고 데이터의 왜곡이 최소화되기 때문이다. 또한 JPEG2000에서 추가된 ROI(Region of Interest)와 같은 추가기능 등을 쉽게 활용할 수 있게 된다. 코드블록의 독립적인 부호화는 위와 같은 장점을 가지게 된다. 하지만 JPEG2000은 코드블록을 bit-level 단위로 다시 분리하여 인코딩하게 된다. 8×8 크기와 8비트의 정보를 가진 코드블록이라면 1개의 부호 비트 평면과 7개의 크기 비트 평면을 가지게 된다. 이와 같은 특성 때문에 연산시간을 증가시키는 요인으로 작용하게 된다.

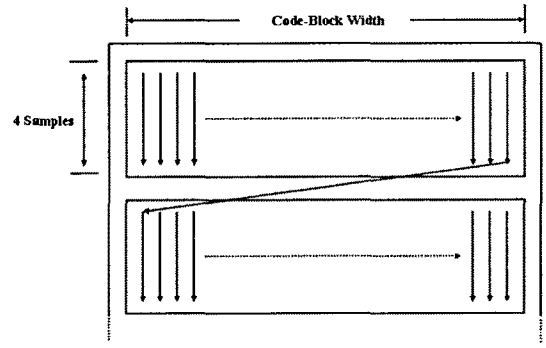

그림 2. Context 추출을 위한 스캔 패턴
Fig. 2. Scan Pattern for Context Extraction

그림 2는 Context 추출을 위한 스캔 패턴을 나타내고 있다[5]. JPEG2000은 일반적인 영상의 가로 스캔 패턴과는 다르게 독특한 스캔 패턴을 사용하는데 첫 번째 열의 세로 4 샘플을 먼저 스캔하고 그 다음 열로 이동하여 코드 블록의 크기만큼 반복 스캔한 후 5번째 행으로 이동하여 다시 세로 4 샘플을 스캔한다. 그림 3은 코딩하고자 하는 샘플위치가 x라고 가정하였을 경우 주변의 참조 샘플 8개의 위치를 나타낸 것이다[6].

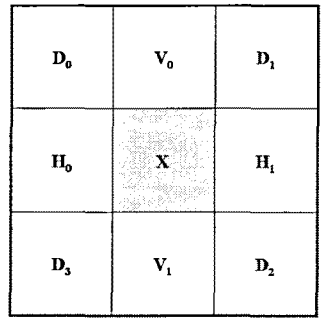

그림 3. 코딩샘플과 이웃 샘플의 예
Fig. 3. Example of coding sample and neighborhood sample

EBCOT Tier-1에서는 현재 비트 평면 샘플과 코드 블록의 크기와 동일한 1비트 상태를 표시하기 위한 Significant state와 Delayed significant state를 참조하여 코딩하게 된다. Tier-1 코더는 MSB 위치의 상위 비트 평면부터 하위 비트 평면으로 내려오면서 코딩하게 된다. 이와 같은 비트 평면 코딩은 일반적으로 영상의 중요 정보가 LSB보다는 MSB 쪽에 많기 때문이다. 중요도를 표시하는 Significant state와 Delayed Significant state는 초기 상태에서 모두 0이 되고 코

딩을 수행하면서 샘플의 중요도를 체크하는 중요한 정보가 된다. JPEG2000은 JPEG에 비해 축약된 19개의 Context와 Decision을 추출하게 되는데 Context를 추출하기 위한 방법으로 4가지 코딩 기법인 ZC(Zero Coding), SC(Sign Coding), RLC(Run Length Coding), MRC(Magnitude refinement Coding)를 사용하여 코딩하게 된다. 그리고 이 코딩 기법을 조합하여 총 3 가지의 코딩패스(Significant Propagation pass, Magnitude Refinement pass, Clean-up Pass)를 사용하여 Context와 Decision을 추출하고 그룹화하게 된다. Significant Propagation Pass는 ZC과 SC을 수행하게 된다. 그리고 Magnitude Refinement Pass는 MRC를 수행하고 Clean-up Pass는 RLC, SC와 ZC를 수행한다. 이를 통해 추출된 Context와 Decision은 산술부호화기를 통해 부호화되게 된다. MQ- Coder는 JPEG2000에서 채택하고 있는 적응 산술부호화기로서, 각 샘플간의 상관관계를 이용해 Context 코더에서 추출된 Context에 기반을 둔 통계적인 확률 예측 테이블을 이용하여 적응적으로 확률을 추정하는 방법을 사용한다. 그림 4는 MQ- Coder의 기본 구조를 나타내고 있다[7].

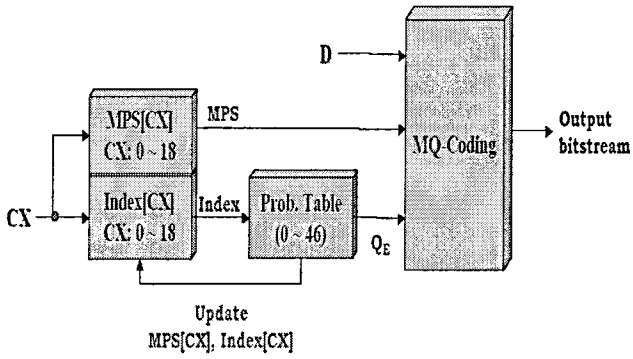

그림 4. MQ- Coder의 구조

Fig. 4. Structure of MQ- Coder

산술 부호화기는 누적확률구간 [0,1]내에서 현재의 부호화 구간을 2개의 부구간(sub-interval) 즉, LPS (Less Probable Symbol)에 대한 구간과 MPS(More Probable Symbol)에 대한 구간으로 분할하는 절차를 반복적으로 수행하며, 입력 Context와 Decision에 대해 누적 확률 분포를 산술적으로 계산하여 코드워드를 생성한다. 이 생성된 코드워드는 Tier-2에서 Header를 삽입하고 데이터를 정렬하여 최종적으로 JPEG 2000 비트 스트림을 생성하게 된다.

III. 제안된 EBCOT 코더의 알고리즘

본 논문은 2절에서 언급한 바와 같이 가장 많은 연산시간을 차지하고 있는 EBCOT Tier-1을 하드웨어 구현하고자 한다. 또한 시스템을 고속화하고 효율성을 높이기 위한 수정된 Context 추출방법을 제안한다. JPEG2000 알고리즘에서는 부호 비트 평면을 제외한 크기 비트 평면의 최상위 비트 평면에는 Clean -up pass만이 적용되고 나머지 비트 평면은 일정한 순서로 3가지 코딩패스를 통과하게 된다. 하지만 각각의 샘플들은 3가지 패스를 통과하는 동안 실제적으로는 3가지 코딩 패스 중 하나의 코딩 패스만을 통해 코딩되게 된다. 이것이 연산 효율 저하의 주된 원인으로써 JPEG 2000 Tier-1은 한 번의 코딩 과정 동안 메모리 참조를 3번 수행해야 하므로 연산시간이 증가된다. 예를 들어 64X64 크기와 7비트의 크기 비트 평면을 가지는 코드블록을 연산 할 경우라면 64*64*7*3의 연산 시간이 필요하게 된다. 그림 5는 JPEG2000 표준 알고리즘의 코딩 과정 예를 나타낸다[8]. 코딩 과정 중 1, 2, 3으로 표시된 부분은 각각 Pass 1, Pass 2, Pass 3으로 코딩되는 부분을 나타내고, c로 표시된 부분은 해당되는 pass를 제외한 다른 Pass로 코딩 되는 부분을 뜻한다.

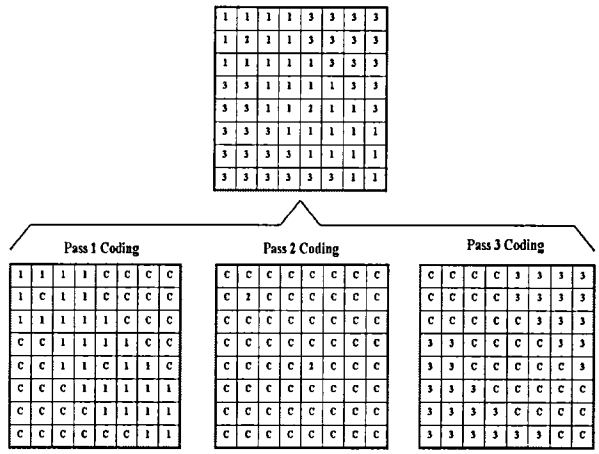

그림 5. JPEG2000 표준 알고리즘의 코딩 예시

Fig. 5. Example of JPEG2000 standard coding method

이와 같은 문제점은 고해상도 영상을 압축할 경우에 더욱 심각해지게 된다. 물론 EBCOT 과정을 병렬로 처리하여 고속화할 수 있지만 이것은 하드웨어 부담이 증가하는 요인이 된다. 표준 알고리즘에서 하나의 비트 평면을 코딩하기 위한 과정은 Pass 1, Pass 2, Pass 3의 과정을 순차적으로 수행하여야 하므로 총 3회의 스캔을 수행해야 한다. 하지만 기존 JPEG2000 표준 방법에서 비트 평면의 값과 Significant state, Delayed significant state의 상태에 따라서 코딩 패스가 결정되는 특성을 이용하여 사전에 코딩 패스를 결정하여 메모리 참조 횟수를 줄이는 것이 수정된 Context 추출 방법이다. 제안된 패스 결정 방법은 최초 코딩 샘플의 Significant 상태를 체크하게 된다. 현재 샘플이 중요한(Significant State = 1) 상태 값을 가지면 Magnitude Refinement pass로 결정되고, 그렇지 않다면 현재 샘플 주변의 이웃 샘플들의 주변 값을 체크하여 Significant 상태가 하나라도 있다면 Significant Propagation pass, 그렇지 않고 주변 상태 값이 전부 0라면 Clean-up pass로 결정된다. 코딩 패스가 결정된 후 이 정보를 이용해 코드블록의 비트 평면을 한꺼번에 코딩하게 된다. 이 같은 사전 코딩 패스 결정법은 패스 결정 시 메모리 참조 1회와 context 추출 시 메모리 참조 1회 총 2회의 메모리참조로 Coding 과정을 끝낼수 있다. 그림 6은 ion 에서 제안된 코딩 방법의 예를 나타낸다[8]. 제안된 알고리즘은 끝림 6에서 보듯이 사전에 코딩할 패스를 결정하는 Pass Decision 과정과 이 정보를 이용하여 샘플의 Context와 Decision을 출력

하는 Coding 과정으로 분류된다. 그림 6의 Coding과정에 나타낸 c는 Coding된 결과를 나타낸다. 이를 그림 5의 JPEG2000 표준 코딩 방법과 비교하면 Pass 1, Pass 2, Pass 3의 총 3회의 스캔이 필요한 과정을 2회의 스캔으로 동일한 Context와 Decision을 얻을 수 있다. 제안된 방법은 모든 코딩 패스(Pass 1, Pass 2, Pass 3)의 연산 시간이 모두 동일하다고 가정하였을 경우 기존 방법에 비해 약 33% 정도의 시간을 절약할 수 있게 된다. 이것은 표 1의 설명에도 언급하였듯이 Pass 1, Pass 2, Pass 3의 처리시간은 영상의 특성과, 시뮬레이션 조건에 따라 달라 질수 있으므로 이와 같이 나타낸 것이다.

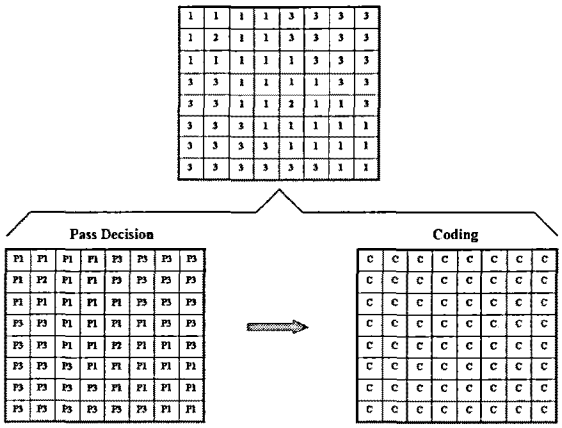

그림 6. 제안된 코딩 방법의 예시
Fig. 6. Example of proposed coding method

아래의 그림 7은 JPEG2000 표준 알고리즘과 본 논문에서 제안된 알고리즘과의 결과 비교를 나타낸 것이다. 그림 9의 4-level Lossy DWT 결과를 입력으로 사용하여 출력되는 Context(좌)와 Decision(우)이 차이가 없음을 확인하였다.

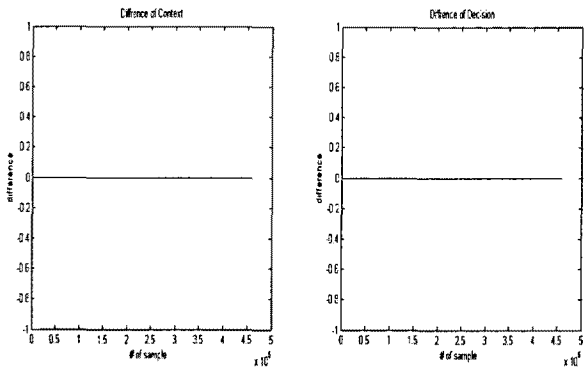

그림 7. 표준 방법과 제안된 코딩 방법의 결과 차이
Fig. 7. Difference of results between standard method and proposed method

EBCOT Tier-1의 기본처리 단위인 코드블록의 크기는 최대 64×64로 규정되어 있다. 이는 고해상도의 영상을 압축하게 될 경우 코드블록의 숫자는 자연스럽게 늘어나게 된다. 기존의 방법을 사용하게 되면 실제연산에 쓰이지 않는 낭비 시간이 누적되게 되고 이는 전체 시스템의 처리속도를 저하시키는 영향을 미치게 된다. 또한 JPEG2000은 알고리즘의 특성상 많은 양의 버퍼 메모리를 사용하게 되는데 고해상도 영상은 고용량의 외부 버퍼 메모리를 필요하게 되므로, 인코딩 과정 중에 더욱더 많은 메모리 참조 지연시간이 소모되게 된다. 즉 제안된 알고리즘은 이러한 낭비시간을 줄일 수 있으므로 구현성 측면에서도 이득을 얻게 된다. JPEG2000에서는 이와 같은 연산속도 뿐 아니라 압축률 또한 중요한 고려 사항 중 하나이다. JPEG2000에서 압축률을 실질적으로 결정하는 부분은 EBCOT TIER-2 과정이다. 본 논문은 압축률에는 영향을 미치지 않고 TIER-1 과정의 하드웨어 설계를 통한 연산시간 감소에 중점을 두었으므로 압축률의 향상이나 저하는 없다.

IV. 하드웨어 구조 및 시뮬레이션 결과

제안된 EBCOT Tier-1 시스템은 Verilog-HDL을 이용하여 구현되었다. 아래의 그림 8은 본 논문에서 제안된 시스템의 전체 블록 다이어그램을 나타내고 있다[8].

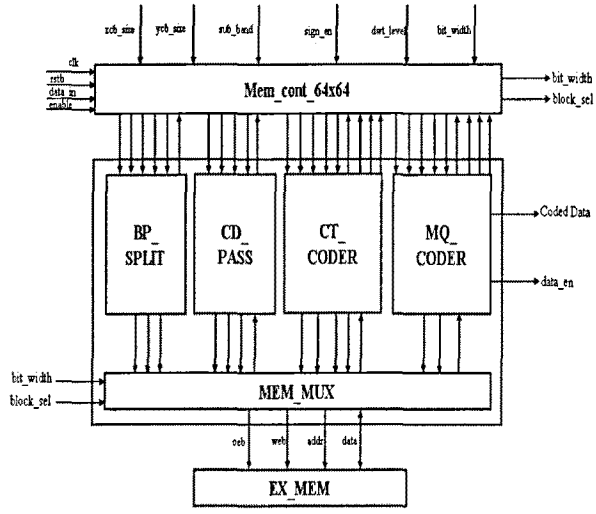

그림 8. 제안된 시스템의 블록 다이어그램
Fig. 8. Block Diagram of proposed system

총 6개의 세부 블록과 1개의 외부 메모리 블록으로 구성되어 있다. JPEG2000 내부의 다른 시스템과 인터페이스를 용이하게 하기 위하여 EBCOT 과정 중에 수행되는 모든 데이터는 외부 메모리에 영역을 할당하여 저장하였다. 블록 Mem_cont_64x64는 DWT블록에서 입력되는 코드블록을 라인 메모리에 저장하고 DWT블록에서 입력되는 제어신호를 통해 EBCOT Tier-1 전체의 동기 신호와 제어 신호를 생성한다. 블록 BP_SPLIT은 입력되는 DWT 계수를 2절에 설명되었던 EBCOT 스캔 패턴에 맞도록 외부 메모리에 저장하는 블록이다. 블록 Mem_cont_64x64에서 입력되는 영상서 입어신호를 통해 데이터를 스캔패턴에 맞도록 어드레스를 발생시키고 비트 평면 별로 분해하여 정렬, 저장하는 역할을 하게 된다. CD_PASS블록은 3절에서 언급한 바와 같이 EBCOT Tier-1

코딩에 사용될 세 가지 코딩 패스를 결정하게 된다. 최상위의 Magnitude 비트 평면에서는 코딩 패스를 결정하기 위한 Significant State와 Delayed Significant State를 메모리에 모두 0으로 리셋한다. 리셋된 Significant State를 참조하여 코딩 패스를 결정하게 되면 최상위의 Magnitude 비트 평면에는 Clean-up pass만이 수행되게 된다. 패스 결정이 끝난 코딩 패스 값은 외부 메모리에 저장하게 된다. 하위의 비트 평면을 처리하게 되면서 CT_CODER 블록에서 수행된 결과가 Significant State에 영향을 미치게 되므로 하나의 코드 블록이 상위 비트 평면부터 하위 비트 평면까지 모두 코딩될 때까지 Significant State를 유지하게 된다. CT_CODER 블록은 비트 평면 데이터와 Significant State, Delayed Significant State, CD_PASS 블록에서 결정된 코딩 패스 값을 조합하여 19종류의 Context와 Decision을 추출한다. 내부에는 ZC, SC, MRC, RLC의 회로가 설계되어 있다. 내부는 FSM(Finite State Machine)으로 코딩 패스에 따라 할당된 코딩을 수행하고 수행을 완료한 후 다른 블록과의 인터페이스 신호를 발생하게 된다. 블록 MQ_CODER는 CT_CODER에서 추출된 Context와 Decision을 입력받아 최종적인 코드스트림과 인터페이스를 위한 유효 데이터 판별 신호를 생성한다. MQ_CODER는 FSM으로 설계되어 시스템 제어 신호에 따라 산술 부호화를 수행하여 최종 데이터를 출력한다. 내부에는 확률 계산을 위한 ROM이 설계되어 있으며 내부의 연산은 곱셈기 대신 쉬프터와 가산기를 사용하여 하드웨어 복잡도를 낮추도록 하였다. 제안된 시스템의 처리결과의 적설성을를 확인하기 위하여 그림 9와 같은 Lena Image를 사용하여 4-level Lossy DWT를 수행한 결과를 사용하여 시뮬레이션을 수행하였다.

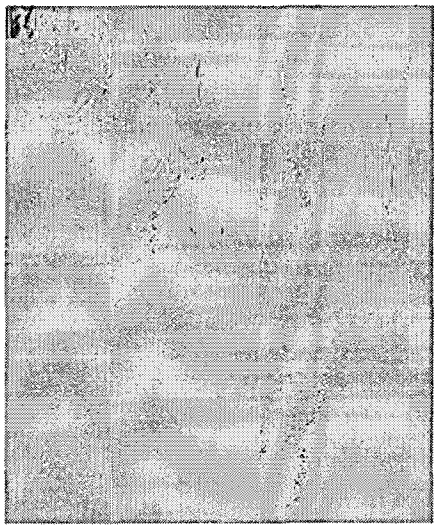

그림 9. 시뮬레이션을 위한 4-레벨 DWT 결과
Fig. 9. Result of 4-level DWT for simulation

그림 10과 그림 11은 본 논문에서 제안된 시스템의 정상 동작을 확인하기 위해 수행한 시뮬레이션 과정을 나타낸 것이다.

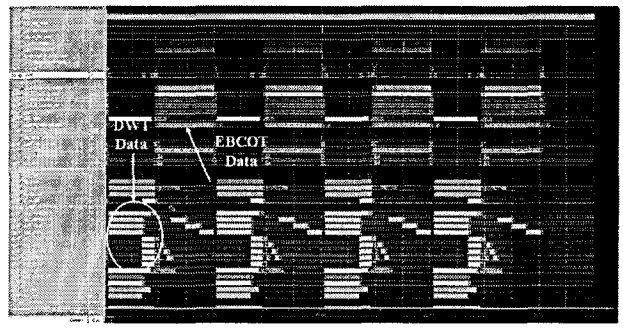

그림 10. EBCOT Tier-1 시스템의 시뮬레이션-1
Fig. 10. Simulation No.1 of EBCOT Tier-1 system

그림 10과 같이 기존에 제안된 Tiling 시스템과 2-D DWT 시스템과 연동하여 인터페이스 및 EBCOT Tier-1 코딩이 제대로 수행되는지 여부도 확인하였다[10-11]. 그림 11은 EBCOT Tier-1 과정의 시뮬레이션 결과를 좀 더 자세히 나타낸 것으로 4레벨의 DWT 데이터를 처리한 67EA의 결과 패킷이 생성된 것을 확인하였다.

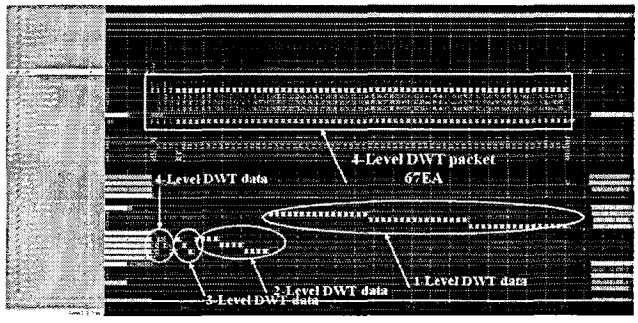

그림 11. EBCOT Tier-1 시스템의 시뮬레이션-2
Fig. 11. Simulation No.2 of EBCOT Tier-1 system

소프트웨어 시뮬레이션 코드와 그림 10, 그림 11의 하드웨어 시뮬레이션 결과의 비교를 통해 하드웨어 동작의 적절성에 대하여 검증하였다. 설계가 완료된 EBCOT Tier-1 시스템은 Synopsys Design Analyzer와 TSMC 0.25um ASIC 라이브러리를 사용하여 합성하였다.

표 2. 제안된 시스템의 합성 결과
Table. 2 Synthesis Results of proposed system

Module Name	Gate Counts
BP_SPLIT	1,208
CD_PASS	5,354
CT_CODER	11,857
MEM_cont_64x64	1,577
MEM_MUX	1,600
MQ_CODER	8,915
TOTAL	30,511

합성 결과는 위의 표 2와 같다. 총 Gate count는 2-input NAND 게이트를 기준으로 30,511개로 구현되었다. 또한 제안

된 시스템은 50MHz의 동작 조건을 만족하였다.표 3은 다른 시스템과의 gate counts 비교를 위한 것이다. 표 2의 gate counts는 데이터 패스만을 나타낸 것으로서 다른 시스템 [12-13]에 비하여 작게 구현되었음을 확인하였다. 하지만 본 논문의 구조는 정지 영상만을 처리하도록 하였고, 비교 논문들은 실시간 동작을 고려한 병렬 구조로 설계되었으므로 단순 비교는 어려울 것으로 판단된다.

표 3. 다른 시스템과 제안된 시스템의 게이트 카운트 비교
Table. 3 Comparison of gate counts between other systems and proposed system

Architecture Design	Gate Counts
[11]	122,204
[12]	152,136
Proposed	30,511

V. 결 론

본 논문은 칼라 영상 압축용 JPEG2000 인코더에 적용하기 위한 EBCOT Tier-1 코더의 하드웨어 구현에 관한 것이다. EBCOT Tier-1 엔트로피 코딩은 JPEG2000 인코딩 과정 중 가장 많은 연산량을 가지고 있다. 이를 극복하기 위한 방법으로 수정된 사전 코딩 패스 결정법을 통한 수정된 Context 추출 방법을 제안하고 하드웨어 구현을 통해 시스템을 고속화하였다. 본 논문에서 제안된 context 추출방법은 엔트로피 코딩의 종류를 결정하는 코딩 패스의 사전판단을 통해 연산시간의 낭비를 줄일 수 있으며 하드웨어 구현 시 메모리 참조 횟수를 3회에서 2회로 줄여 효율성을 높일 수 있는 장점을 가지고 있다. 구현된 산술 부호화기 MQ- Coder는 상태 다이어그램을 이용한 FSM 방식을 사용하였고 곱셈기 대신 가산기와 쉬프터로 구현하여 하드웨어를 소형화 하였다. 또한기존에 구현된 Tiling 시스템, DWT 시스템과 연동하여 동작의 적절성 여부를 검증하였다. 구현된 시스템은 총 30,511개의 게이트 카운트를 가지며 50MHz의 동작 주파수를 만족하였다. 본 논문에서 제안한 EBCOT Tier-1 코더는 기존에 구현된 Tiling 시스템, 2-D DWT 시스템과 결합하여 고해상도 영상 압축용 JPEG2000 Hardwired Encoder에 적용할 수 있다. 하드웨어로 구현된 JPEG 2000 Encoder는 소프트웨어 JPEG2000 Codec보다 기능면에서 특성화되고 고속처리가 가능해진다. 그래서 모바일 카메라, 디지털 카메라, 또는 PMP(Portable Multimedia Player)와 같은 각종 멀티미디어 휴대용 기기에 적용하여 압축효율을 높이고 좀더 좋은 화질을 획득 할 수 있다. 낮은 대역폭을 가지는 회선을 사용하더라도 빠르게 전송할 수 있고, 또한 화질 측면에서도 JPEG과 같은 압축률을 사용하더라도 보다 좋은 영상처리결과를 얻을 수 있다.

참 고 문 헌

[1] 김진만, 주동현, 김두영, "임베디드 시스템의 영상압축을 위한 분할정렬 알고리즘의 개선," 한국신호처리시스템학회 논문지, 제 6권 3호, pp. 321-328, 2005.7

[2] K. Andra, C. Chakrabarti and T. Achiarya, "A high -performance JPEG2000 Architecture," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol.13, No. 3, pp. 209-218, March 2003.

[3] C. Lian, K. Chen, H. Chen and L. Chen, "Analysis and Architecture Design of Block-Coding Engine for EBCOT in JPEG2000," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol.13, No. 3, pp. 219-230, March 2003.

[4] ITU-R T.800, Information technology - *JPEG2000 Image coding system : Core Coding System,:* 2002

[5] D. Taubman, ""High performance scalable image compression with EBCOT,"" *IEEE Transaction on Image Processing*, vol. 9, pp. 1158 - 1170, July 2000.

[6] M. Lee, J. Kim and B. Kang,"Hardware Implementation of EBCOT coder with ISO/IEC 15444-1 JPEG2000 standard", *IT-SoC 2005 Conference*, pp. 675-678, March 2003.

[7] IDEC, JPEG2000 정지영상 압축부호화 표준의 이해, 2002.

[8] 이성목, 송진근, 하주영, 이민우, 강봉순 "고해상도 정지영상 압축을 위한 효율적인 JPEG2000용 Context 추출 연구", 한국해양정보통신학회 추계학술대회 논문집, pp.97-100, 2007.10.

[9] 이성목, 조성대, 이민우, 강봉순, "고해상도 JPEG2000 Hardwired Encoder를 위한 EBCOT Tier-1 Coder의 구현", 한국신호처리시스템학회 하계학술대회 논문집, pp.161-164, 2008.6.

[10] 이성목, 조성대, 강봉순, "JPEG2000 Hardwired Encoder를 위한 칼라 2D DWT Processor의 구현," 한국신호처리시스템학회 논문지, 제9권 4호, pp. 321-328, 2008.10.

[11]장원우, 조성대, 강봉순, "JPEG2000을 위한 Tiling 시스템의 구현," 한국신호처리시스템학회 논문지, 제9권 3호, pp. 201-207, 2008.7.

[12]H. C. Fang, T. C.Wang, C. J. Lian, T. H. Chang, and L. C. Chen, "High speed memory efficient EBCOT architecture for JPEG2000," in Proc. *IEEE International Symposium Circuits and System*. pp. 736-739, May 2003.

[13]Y. Zhang, C. Xu, W. Wang, L. Chen, "Performance Analysis and Architecture Design for Parallel EBCOT Encoder of JPEG2000," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol.17, No. 10, pp. 1336-1347, March 2003.

이 성 목 (Sung-mok Lee)

2005년 2월 동아대학교 전기전자컴퓨터공학부 전자공학과(공학사)
2007년 2월 동아대학교 전기전자컴퓨터공학부 전자공학과(공학석사)

2007년 3월~현재 동아대학교 전자공학과 박사과정
관심분야 : VLSI algorithm/architecture design, image/video processing, and wireless communication.

장 원 우 (Won-woo Jang)

2005년 2월 동아대학교 전기전자컴퓨터공학부 전자공학과(공학사)
2007년 2월 동아대학교 전기전자컴퓨터공학부 전자공학과(공학석사)

2007년 3월~현재 동아대학교 전자공학과 박사과정
관심분야 : VLSI algorithm/architecture design, image/video processing, and wireless communication.

조 성 대 (Sung-dae Cho)

1996년 숭실대학교 전자계산학과(공학사)
2000년 미국 Rensselaer Polytechnic Institute 전자컴퓨터공학(공학석사)
2002년 미국 Rensselaer Polytechnic Institute 전자컴퓨터공학(공학박사)

2004년 RPI 영상처리센터 박사후 연구원
2004년~현재 삼성전자 DMC연구소 수석연구원
관심분야 : Multimedia image processing, color processing, computer vision, and data compression

강 봉 순 (Bong-soon Kang)

1985년 연세대학교 전자공학과(공학사)
1987년 미국 University of Pennsylvania 전기공학과(공학석사)
1990년 미국 Drexel University 전기 및 컴퓨터공학과(공학박사)

1989년~1999년 삼성전자 반도체 수석연구원
1999년~현재 동아대학교 전자공학과 교수
2006년~현재 멀티미디어 연구센터 소장
2006년~현재 2단계 BK21 사업팀장
관심분야 : VLSI algorithm/architecture design, image/video processing, and wireless communication.