(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0037968
(43) 공개일자 2019년04월08일

(51) 국제특허분류(Int. Cl.)
H04N 19/13 (2014.01) *H04N 19/103* (2014.01)
H04N 19/184 (2014.01)
(52) CPC특허분류
H04N 19/13 (2015.01)
H04N 19/103 (2015.01)
(21) 출원번호 10-2017-0127846
(22) 출원일자 2017년09월29일
심사청구일자 2017년09월29일

(71) 출원인
한밭대학교 산학협력단
대전광역시 유성구 동서대로 125 (덕명동)
(72) 발명자
류광기
대전광역시 유성구 덕명동 한밭대학교 N4동 407호
조현구
대전광역시 유성구 덕명동 한밭대학교 N4동 513호
(74) 대리인
추혁

전체 청구항 수 : 총 4 항

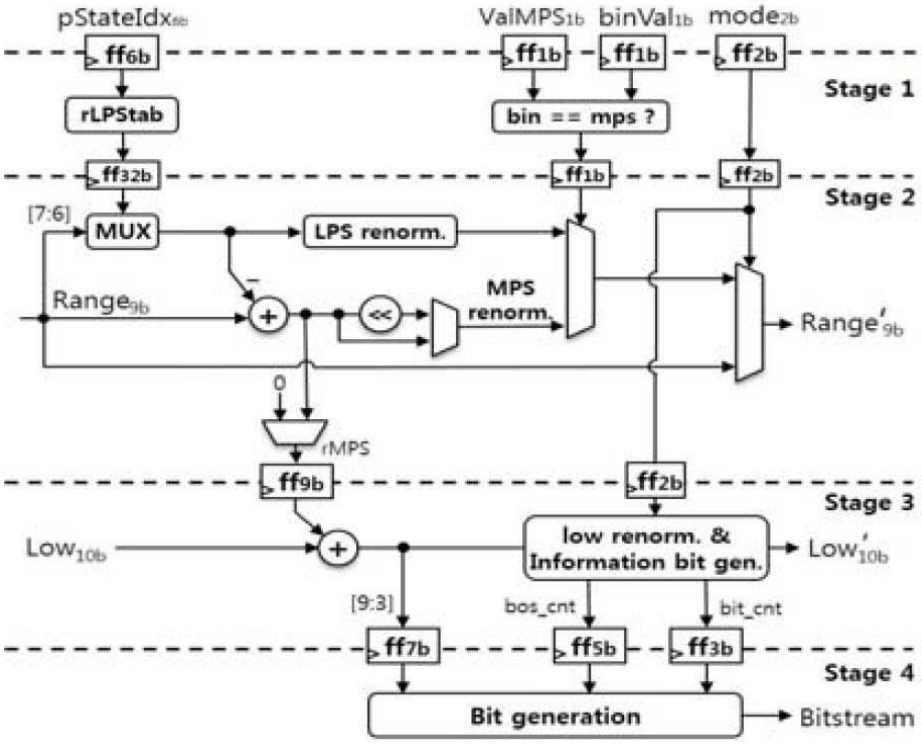
(54) 발명의 명칭 **UHD 영상의 고속 처리를 위한 CABAC 이진 산술 부호화기**

(57) 요 약

본 발명은 UHD 영상의 고속 처리를 위한 CABAC 이진 산술 부호화기에 관한 것으로서, 문맥 모델러에서 입력받은 문맥 모델 값(ValMPS)이 빈 값(binVal)과 같으면 현재 부호화하는 빈을 나올 확률이 가장 큰 심볼(MPS, Most Probable Symbol)로, 그리고 ValMPS가 빈 값과 같지 않으면 현재 부호화하는 빈을 나올 확률이 가장 적은 심볼

(뒷면에 계속)

대 표 도 - 도3



(LPS, Least Probable Symbol)로 결정하는, 심볼 확률 및 LPS 범위 후보 결정 스테이지, 현재 부호화하는 빈의 범위(Range)가 일정 범위보다 작아지면 재정규화를 수행하고, 재정규화 횟수(Cnt_RenormE)와 상기 범위의 최소 경계 값(Low) 계산에 필요한 MPS의 범위(rMPS)를 출력하는 범위 업데이트 스테이지, 상기 rMPS와 상기 Low의 값을 더하여 다음 호출을 위한 ivlLow 값을 구하며 ivlLow 값과 상기 Cnt_RenormE를 통하여 현재 부호화하는 빈에 대한 비트스트림 생성에 필요한 정보비트를 출력하는 최소 경계값 업데이트 스테이지 및 현재 부호화하는 빈에 대한 상기 정보비트를 통하여 비트스트림을 출력하는 비트스트림 생성 스테이지로 이루어짐으로써, 최대 지연 시간(Critical Path)을 최적으로 줄일 수 있다.

(52) CPC특허분류
H04N 19/184 (2015.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	R2418160007000 1002
부처명	미래창조과학부
연구관리전문기관	정보통신기술진흥센터
연구사업명	2016년 해외인재 스카우팅 사업
연구과제명	마이크로니들 센서 기반 패치형 원격 혈당측정기 개발
기 여 율	1/1
주관기관	연구사업관리팀
연구기간	2016.06.01 ~ 2016.12.31

명 세 서

청구범위

청구항 1

문맥 기반 적응적 산술 부호화(CABAC, Context-based Adaptive Binary Arithmetic Coding)에 있어서,

문맥 모델러에서 입력받은 문맥 모델 값(ValMPS)이 빈 값(binVal)과 같으면 현재 부호화하는 빈을 나올 확률이 가장 큰 심볼(MPS, Most Probable Symbol)로, 그리고 ValMPS가 빈 값과 같지 않으면 현재 부호화하는 빈을 나올 확률이 가장 적은 심볼(LPS, Least Probable Symbol)로 결정하는, 심볼 확률 및 LPS 범위 후보 결정 스테이지,

현재 부호화하는 빈의 범위(Range)가 일정 범위보다 작아지면 재정규화를 수행하고, 재정규화 횟수(Cnt_RenormE)와 상기 범위의 최소 경계 값(Low) 계산에 필요한 MPS의 범위(rMPS)를 출력하는 범위 업데이트 스테이지,

상기 rMPS와 상기 Low의 값을 더하여 다음 호출을 위한 ivlLow 값을 구하며 ivlLow 값과 상기 Cnt_RenormE를 통하여 현재 부호화하는 빈에 대한 비트스트림 생성에 필요한 정보비트를 출력하는 최소 경계값 업데이트 스테이지 및

현재 부호화하는 빈에 대한 상기 정보비트를 통하여 비트스트림을 출력하는 비트스트림 생성 스테이지로 이루어진, CABAC 이진 산술 부호화기.

청구항 2

청구항 1에서,

상기 범위 업데이트 스테이지는 현재 부호화되는 범위의 크기(ivlCurrRange)의 최상위 비트(MSB: Most Significant Bit)부터 첫 번째 1의 위치를 찾아 재정규화 횟수를 계산하고 반복 횟수만큼 왼쪽으로 시프트 하여 상기 재정규화를 수행하는, CABAC 이진 산술 부호화기.

청구항 3

청구항 1에서,

상기 비트 생성 스테이지는 상기 ivlLow의 상위 7비트(Low_data), 비트스트림 출력을 위한 준비 비트(bitOutstanding)의 횟수 및 출력 비트수를 입력받아 비트스트림을 출력하는, CABAC 이진 산술 부호화기.

청구항 4

청구항 1 내지 3 중 어느 한 항에 있어서,

멀티-빈 구조를 적용하여 여러 개의 빈을 동시에 처리하는, CABAC 이진 산술 부호화기.

발명의 설명

기 술 분 야

[0001] 본 발명은 CABAC 이진 산술 부호화기에 관한 것으로, 더욱 상세하게는 재정규화 과정을 분리시켜 기존의 반복적인 알고리즘을 병렬적으로 처리함으로써 최대 지연 시간(Critical Path)을 최적으로 줄일 수 있는 UHD 영상의 고속 처리를 위한 CABAC 이진 산술 부호화기에 관한 것이다.

배 경 기 술

[0002] 최근 영상 기술의 발달로 Full-HD를 넘어 4배 이상의 해상도를 갖는 UHD(Ultra High Definition, 초고해상도) 영상에 대한 사용자의 요구가 증가함에 따라 초고해상도 영상 서비스를 위한 기존 영상 압축 표준인 H.264/AVC 보다 높은 성능을 갖는 영상 압축 표준이 필요하게 되었다. 차세대 영상 압축 표준인 HEVC(High Efficiency

Video Coding, 고효율 비디오 코딩)는 저해상도인 176×144 영상부터 초고해상도인 4K, 8K 영상까지 다양한 영상의 압축을 지원하며, 이전 영상 압축 표준인 H.264/AVC와 비교하여 약 50% 이상 향상된 압축 효율을 갖지만 복잡도와 연산량이 증가하여 실시간 처리의 어려움을 보이고 있다. H.264/AVC 표준의 엔트로피 부호화 방법은 CABAC(Context-based Adaptive Binary Arithmetic Coding, 문맥 기반 적응적 산술 부호화)과 CAVLC(Context-based Adaptive Variable Length Coding, 문맥 기반 적응적 가변장 부호화) 두 가지를 모두 사용하였지만, HEVC에서는 압축 효율이 더 높은 CABAC만을 채택하여 사용하고 있다.

[0003] CABAC 부호화는 각 문맥 요소에 대한 확률 모델을 선택하는 문맥 기반 모델링(Context-based Modeling) 방법을 통해 적응적 이진 산술 부호화를 수행하여 부호화한다. CABAC 부호화기는 도 1과 같이 이진화기(Binarizer)(100) 문맥 모델러(Context Modeler)(200) 및 이진 산술 부호화기(BAE: Binary Arithmetic Encoder)(300)로 구성된다.

[0004] 도 1을 참조하여, CABAC에서의 이진화기(100)는 이진 값이 아닌 신택스 요소(Syntax Element)를 빈 스트링(Bin String)이라 불리는 이진 시퀀스로 처리한다. 이진 값을 가지는 신택스 요소일 경우 이진화 과정을 패스하게 된다. 문맥 모델러(200)는 부호화 블록의 주변 정보 값인 문맥(Context)을 이용하여 문맥 모델 확률을 추정한다. 이진 산술 부호화기(300)는 이진화된 값인 빈(Bin)과 문맥 모델러의 확률 값을 이용하여 범위(Range)를 설정하면서 부호화를 수행한다. 범위가 일정 크기(256) 미만으로 작아지게 되면 범위를 재설정하는 재정규화(Renormalization) 과정을 수행하며, 재정규화 과정에서 비트스트림(Bitstream)을 생성한다. 이진화기(100)를 거쳐 이진화된 빈은 정규 모드(Regular Mode), 우회 모드(Bypass Mode), 종결 모드(Terminate Mode)의 세 가지 모드로 이진 산술 부호화를 수행한다. 정규 모드는 문맥 모델러에서 생성된 확률 함수에 대해 적용될 초기 값(pStateIdx, ValMPS)을 가지고 부호화한다. 우회 모드는 문맥 모델러를 사용하지 않으며 동일한 확률로 빈을 부호화한다. 종료 모드는 슬라이스의 종료 여부를 판단하는 신택스 요소(end_of_slice_flag)의 빈을 부호화한다.

[0005] 구체적으로, 이진 산술 부호화기(300)는 문맥 모델러(200)에서 확률 값을 받아 현재 빈에 대하여 이진 산술 부호화를 수행한다. 도 2는 이진 산술 부호화의 과정을 보여준다.

[0006] 도 2를 참조하여, 입력되는 빈 값(binVal)은 '1, 0, 0, 0'이며, 초기 문맥 모델 값이 pStateIdx=0, ValMPS=0을 가질 경우의 이진 산술 부호화 과정은 다음과 같다. Range(현재 구간범위)와 Low(현재 구간범위의 최소 경계값)의 초기값은 510과 0이며, 첫 번째 입력되는 빈은 '1'이므로 MPS(Most Probable Symbol, 나올 확률이 가장 큰 심볼)와 LPS(Least Probable Symbol, 나올 확률이 가장 적은 심볼)를 결정하기 위해 ValMPS와 binVal를 비교한다. 현재 빈과 MPS의 값이 다르기 때문에 LPS가 선택되어 LPS의 범위인 240은 256보다 작기 때문에 재정규화 과정을 수행한다. 또한, Update CM(Context Model, 문맥 모델)을 통하여 ValMPS와 pStateIdx를 다음 빈에 대한 확률 모델 값으로 업데이트한다. CABAC은 재정규화 과정을 수행할 때마다 비트스트림을 출력하거나, 비트스트림 출력을 위한 준비 비트(BitOutStanding)를 업데이트한다. 이진 산술 부호화기(300)는 Range를 계산해야만 Low를 계산할 수 있고, Range와 Low가 업데이트되어야만 다음 빈을 부호화할 수 있다. 또한, 재정규화 과정은 정규 모드에서 최대 6번 반복될 수 있기 때문에 이진 산술 부호화 과정마다 소요되는 시간과 출력이 일정하지 않다.

[0007] 이와 같이, CABAC은 HEVC 표준에서 사용되는 엔트로피 코딩 방법으로 통계적 중복성을 제거하여 영상의 높은 압축률을 제공하지만, 그것에서의 이진 산술 부호화기는 단계 간의 데이터 의존 관계가 높아 병렬 처리가 어렵고 고속 동작을 어렵게 한다. 즉, CABAC은 하나의 빈을 부호화한 후 확률 모델을 업데이트하고 업데이트된 확률 모델로 다음 빈을 부호화함으로써 압축 효율을 향상시키지만, 현재 빈의 부호화가 끝나기 전에는 확률 모델이 업데이트되지 않아 다음 빈을 부호화할 수 없다.

[0008] 따라서, 본 출원에서는 연산 과정에서 지연되는 구간을 분리하여 최적으로 동작할 수 있는 4단 파이프라인 구조의 CABAC 이진 산술 부호화기를 제안한다.

선행기술문헌

특허문헌

[0009] (특허문헌 0001) 대한민국 등록특허공보 제10-0717052호(등록일자 2007.05.04)

(특허문헌 0002) 대한민국 등록특허공보 제10-0644713호(등록일자 2006.11.03)

(특허문헌 0003) 대한민국 등록특허공보 제10-0717055호(등록일자 2007.05.04)

발명의 내용

해결하려는 과제

[0010] 따라서, 본 발명은 상기한 종래 기술의 문제점을 해결하기 위해 이루어진 것으로서, 본 발명의 목적은 CABAC에서의 이진 산술 부호화기에서 재정규화 과정을 분리시켜 기존의 반복적인 알고리즘을 병렬적으로 처리함으로써 최대 지연 시간(Critical Path)을 최적으로 줄일 수 있는 4단계 파이프라인 구조로 이루어진 UHD 영상의 고속 처리를 위한 CABAC 이진 산술 부호화기를 제공하는 데 있다.

과제의 해결 수단

[0011] 상기와 같은 목적을 달성하기 위한 본 발명의 UHD 영상의 고속 처리를 위한 CABAC 이진 산술 부호화기는, 문맥 모델러에서 입력받은 문맥 모델 값(ValMPS)이 빈 값(binVal)과 같으면 현재 부호화하는 빈을 나올 확률이 가장 큰 심볼(MPS, Most Probable Symbol)로, 그리고 ValMPS가 빈 값과 같지 않으면 현재 부호화하는 빈을 나올 확률이 가장 적은 심볼(LPS, Least Probable Symbol)로 결정하는, 심볼 확률 및 LPS 범위 후보 결정 스테이지, 현재 부호화하는 빈의 범위(Range)가 일정 범위보다 작아지면 재정규화를 수행하고, 재정규화 횟수(Cnt_RenormE)와 상기 범위의 최소 경계 값(Low) 계산에 필요한 MPS의 범위(rMPS)를 출력하는 범위 업데이트 스테이지, 상기 rMPS와 상기 Low의 값을 더하여 다음 호출을 위한 ivlLow 값을 구하며 ivlLow 값과 상기 Cnt_RenormE를 통하여 현재 부호화하는 빈에 대한 비트스트림 생성에 필요한 정보비트를 출력하는 최소 경계값 업데이트 스테이지 및 현재 부호화하는 빈에 대한 상기 정보비트를 통하여 비트스트림을 출력하는 비트스트림 생성 스테이지로 이루어진다.

발명의 효과

[0012] 상술한 바와 같이, 본 발명에 의한 UHD 영상의 고속 처리를 위한 CABAC 이진 산술 부호화기는 다음과 같은 효과를 제공한다.

[0013] 본 발명에서 제안하는 CABAC 이진 산술 부호화기는 입력으로 들어오는 빈을 고속으로 처리하기 위하여 재정규화 과정을 분리시켜 동작하도록 설계한다. 기존의 반복적인 알고리즘을 병렬적으로 처리함으로써 최대 지연 시간(Critical Path)을 최적으로 줄일 수 있는 4단계의 파이프라인 구조로 설계하였다. 또한, 멀티-빈 구조를 적용하여 여러 개의 빈을 동시에 처리한다.

도면의 간단한 설명

[0014] 도 1은 기존의 CABAC 부호화기의 일반적인 구성을 나타낸 블록도이다.

도 2는 기존의 CABAC 부호화기의 일반적인 처리 과정을 나타낸 흐름도이다.

도 3은 본 발명의 바람직한 실시예에 따른 이진 산술 부호화기의 전체 구성을 나타낸 블록도이다.

도 4는 본 발명의 바람직한 실시예에 따른 이진 산술 부호화기의 범위 업데이트 스테이지에서 현재 범위의 크기를 구하는 과정을 나타낸 도식이다.

도 5a는 종래 이진 산술 부호화에서의 재정규화 알고리즘의 과정을 나타낸 흐름도이고, 도 5b는 본 발명의 바람직한 실시예에 따른 이진 산술 부호화에서의 재정규화 알고리즘의 과정을 나타낸 흐름도이다.

도 6은 본 발명의 바람직한 실시예에 따른 이진 산술 부호화기의 비트스트림 생성 스테이지의 구조를 나타낸 도식이다.

도 7은 본 발명의 바람직한 실시예에 따른 이진 산술 부호화기의 멀티-빈 처리를 위한 구조를 나타낸 도식이다.

발명을 실시하기 위한 구체적인 내용

[0015] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로

다른 다양한 형태로 구현될 수 있으며, 단지 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성요소를 지칭한다.

[0016] 이하 본 발명의 UHD 영상의 고속 처리를 위한 CABAC 이진 산술 부호화기에 대하여 첨부된 도면을 참조하여 상세히 설명하기로 한다.

[0017] 도 3은 본 발명의 바람직한 실시예에 따른 이진 산술 부호화기의 전체 구성을 나타낸 블록도이다.

[0018] 도 3에 도시된 바와 같이, 본 발명의 CABAC 이진 산술 부호화기는 재정규화 과정을 분리시켜 기존의 반복적인 알고리즘을 병렬적으로 처리함으로써 최대 지연 시간(Critical Path)을 최적으로 줄일 수 있도록, 심볼 확률 및 LPS 범위 후보 결정 스테이지(Stage 1), 범위 업데이트 스테이지(Stage 2), 최소 경계값 업데이트 스테이지(Stage 3) 및 비트스트림 생성 스테이지(Stage 4)의 4단계 파이프라인 구조로 이루어진다.

[0019] 종래 CABAC 이진 산술 부호화기에서는 재정규화에 의한 비트스트림 생성이 가변적이므로 메모리를 통하여 비트스트림을 출력해야 하는 반면, 본 발명의 본 발명의 CABAC 이진 산술 부호화기는 메모리를 사용하지 않고 유효한 비트스트림의 개수를 나타내는 비트와 비트스트림을 출력하고 하드웨어 면적을 감소시킨다.

[0020] 1. 심볼 확률 및 LPS 범위 후보 결정 스테이지

[0021] 심볼 확률 및 LPS 범위 후보 결정 스테이지(Stage 1)에서는 현재 빈에 대하여 나올 확률이 가장 큰 심볼(MPS, Most Probable Symbol)과 나올 확률이 가장 적은 심볼(LPS, Least Probable Symbol)을 식 (1)과 같이 결정한다. 즉, 문맥 모델러(도 1의 200)에서 입력받은 문맥 모델 값(ValMPS)이 빈 값(binVal)과 같으면 현재 부호화하는 빈을 MPS로, 그리고 ValMPS가 빈 값과 같지 않으면 현재 부호화하는 빈을 LPS로 결정한다.

[0022]
$$\begin{aligned} bin &= MPS \quad \text{if.}(ValMPS = bin) \\ bin &= LPS \quad \text{if.}(ValMPS \neq bin) \end{aligned} \quad (1)$$

[0023] 또한, 심볼 확률 및 LPS 범위 후보 결정 스테이지(Stage 1)에서는 표 1과 같이 문맥 모델러(도 1의 200)에서 입력받은 확률 값(pStateIdx)에 대하여 네 가지의 LPS 범위 후보(rLPS)를 출력한다.

[0024]

표 1

rLPS 표

pStateIdx	qRangeIdx			
	0	1	2	3
0	128	176	208	240
1	128	167	197	227
2	128	158	187	216
...				
61	6	7	9	10
62	6	7	8	9

[0025] 2. 범위 업데이트 스테이지

[0026] 범위 업데이트 스테이지(Stage 2)는 이진 산술 부호화의 범위(Range)가 일정 범위보다 작아지면 재정규화를 수행하고, 재정규화의 횟수(Cnt_RenormE)와 현재 범위의 최소 경계 값(Low) 계산에 필요한 MPS의 범위(rMPS)를 출력한다.

[0027] 심볼 확률 및 LPS 범위 후보 결정 스테이지(Stage 1)에서 입력받은 4개의 LPS 후보(rLPS)를 입력 Range의 7, 6번째 비트(qRangeIdx)로 LPS 범위의 크기(ivlLPSRange)를 결정한다. 현재 부호화하는 빈이 LPS일 경우와 MPS일 경우의 현재 부호화되는 범위의 크기(ivlCurrRange)는 식 (2)와 같다. 즉, 현재 부호화하는 빈이 LPS일 경우 전체 블록의 크기는 LPS 범위의 크기이고, 현재 부호화하는 빈이 MPS일 경우 현재 범위의 크기는 현재 범위의 크기에서 LPS 범위의 크기를 뺀 크기이다. 이를 그림으로 표현하면 도 4와 같다.

[0028]
$$\begin{aligned} ivlCurrRange &= ivlLPSRange && \text{if.}(bin = LPS) \\ ivlCurrRange &= ivlCurrRange - ivlLPSRange && \text{if.}(bin = MPS) \end{aligned} \quad (2)$$

[0029] 이진 산술 부호화의 정규 모드에서 기존의 알고리즘은 ivlCurrRange가 256이상이 될 때까지 최대 6번 좌시프트 연산을 반복적으로 수행하여 Critical Path를 발생시킨 반면, 본 출원에서 제안하는 구조에서는 Range 재정규화의 가변적인 연산을 해결하기 위해 ivlCurrRange의 최상위 비트(MSB: Most Significant Bit)부터 첫 번째 1의 위치를 찾아 재정규화 횟수를 계산하고 반복 횟수만큼 왼쪽으로 시프트 하여 Range 재정규화를 간단하게 수행한다. 도 5a는 종래 이진 산술 부호화에서의 재정규화 알고리즘의 과정을 나타낸 흐름도이고, 도 5b는 본 발명의 바람직한 실시예에 따른 이진 산술 부호화에서의 재정규화 알고리즘의 과정을 나타낸 흐름도이다.

[0030] 3. 최소 경계값 업데이트 스테이지

[0031] 최소 경계값 업데이트 스테이지(Stage 3)은 rMPS와 Low의 값을 더하여 다음 호출을 위한 ivlLow 값을 구하며 ivlLow 값과 재정규화 횟수를 통하여 현재 부호화하는 빈에 대한 비트스트림 생성에 필요한 정보비트를 출력한다. 현재 부호화하는 빈이 LPS일 경우와 MPS일 경우의 ivlLow는 식 (3)과 같다. 즉, 현재 부호화하는 빈이 LPS일 경우 ivlLow는 ivlLow에 rMPS를 더한 값이고, 현재 부호화하는 빈이 LPS일 경우 ivlLow는 ivlLow 값이다.

[0032]
$$\begin{aligned} ivlLow &= ivlLow + rMPS && \text{if.}(bin = LPS) \\ ivlLow &= ivlLow && \text{if.}(bin = MPS) \end{aligned} \quad (3)$$

[0033] CABAC 부호화기는 재정규화 횟수만큼 비트스트림을 생성하는데, 본 발명에서 제안하는 4단계 파이프라인 구조는 표 2와 같이 재정규화 횟수만큼 ivlLow를 좌시프트하고 최상위 비트를 0으로 만들어 재정규화(B)를 진행하며, 일부 조건의 경우 MSB를 변경하지 않고 유지한 채 재정규화(A)한다.

표 2

[0034]

ivlLow Index	Cnt_RenormE						
	0	1	2	3	4	5	6
[9]=1 or [0]=0	ivlLow	A	B	B	B	B	B
[9:7]=111 or [7]=0	ivlLow	B	A	B	B	B	B
[9:6]=1111 or [6]=0	ivlLow	B	B	A	B	B	B
[9:5]=11111 or [5]=0	ivlLow	B	B	B	A	B	B
[9:4]=111111 or [4]=0	ivlLow	B	B	B	B	A	B
[9:3]=1111111 or [3]=0	ivlLow	B	B	B	B	B	A
A : ivlLow << Cnt_RenormE B : (ivlLow << Cnt_RenormE) [MSB] <= 0							

[0035] 비트스트림에 필요한 정보비트의 생성은 ivlLow 값과 재정규화 횟수(Cnt_RenormE)에 따라서 ivlLow의 상위 재정규화 횟수 + 1 비트를 이용하여 비트스트림의 출력을 나타내는 bit_cnt와 비트스트림의 값이 결정되지 않은 비트의 개수를 나타내는 비트스트림 출력을 위한 준비 비트(bitOutstanding)의 횟수인 bos_cnt를 결정한다. 표 3은 재정규화 횟수에 따른 비트스트림 출력에 필요한 정보비트의 출력을 나타낸다.

표 3

[0036]

ivlLow [9:9-i]			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Cnt_RenormE(i)	1	bit	1	0	1	1												
		bos	0	1	0	0												
	2	bit	2	1	0	1	2	1	2	2								
		bos	0	1	2	1	0	1	0	0								
	3	bit	3	2	3	1	3	2	3	0	3	2	3	1	3	2	3	3
		bos	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0	0
	4	bit	4	3	4	2	4	3	4	1	4	3	4	2	4	3	4	0
		bos	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0	4

ivlLow [9:9-i]			16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
(i)	4	bit	4	3	4	2	4	3	4	1	4	3	4	2	4	3	4	4
		bos	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0	0
	5	...																
	6	...																

[0037] 4. 비트스트림 생성 스테이지

[0038] 비트스트림 생성 스테이지(Stage 4)는 현재 빈에 대한 비트스트림 생성 정보를 통하여 비트스트림을 출력한다. 비트 생성 스테이지(Stage 4)는 ivlLow의 상위 7bit인 Low_data, bitOutstanding의 횟수를 나타내는 bos_cnt, 출력 비트수를 나타내는 bit_cnt를 입력받아 비트스트림을 출력한다.

[0039] 기존의 CABAC 부호화기는 부호화되는 빈에 따라 출력하는 비트스트림의 개수는 가변적으로 정보비트를 저장할 메모리가 필요하지만, 본 출원에서 제안하는 4단계 파이프라인 구조는 가변적인 비트스트림의 개수를 나타내는 출력신호(valid_bit_cnt)를 생성하여 메모리에 저장하지 않고 비트스트림을 출력할 수 있도록 구성되어 하드웨어 면적을 줄일 수 있다. 도 6은 비트스트림 생성 스테이지(Stage 4)의 구조를 나타내며, 표 4는 비트스트림 출력을 위한 테이블을 나타낸다. 표 4의 ((n)~Low_data[6])의 n은 n번의 ~Low_data[6]의 비트 반복을 의미한다.

표 4

[0040]

bos_cnt	Bitstream
0	{Low_data[6:0], ((31)0)}
1	{Low_data[6], ((1)~Low_data[6]), Low_data[5:0], ((30)0)}
2	{Low_data[6], ((2)~Low_data[6]), Low_data[5:0], ((29)0)}
3	{Low_data[6], ((3)~Low_data[6]), Low_data[5:0], ((28)0)}
4	{Low_data[6], ((4)~Low_data[6]), Low_data[5:0], ((27)0)}
...	...
28	{Low_data[6], ((28)~Low_data[6]), Low_data[5:0], ((3)0)}
29	{Low_data[6], ((29)~Low_data[6]), Low_data[5:0], ((2)0)}
30	{Low_data[6], ((30)~Low_data[6]), Low_data[5:0], ((1)0)}
31	{Low_data[6], ((31)~Low_data[6]), Low_data[5:0]}

[0041] 5. 멀티-빈 아키텍처

[0042] 또한, 본 발명에서 제안하는 CABAC 이진 산술 부호화기는 멀티-빈 구조를 적용하여 여러 개의 빈을 동시에 처리한다. 이러한 멀티-빈 처리를 위한 구조는 도 7과 같으며, 클록 사이클 당 빈의 개수(BPCC, Bin Per Clock Cycle)가 1, 2, 3일 경우의 게이트 수, 최대 빈 처리량, 최대 동작 주파수는 표 5와 같다.

표 5

[0043]

BPCC	1	2	3
게이트 수	3.17K	5.7K	8.07K
최대 동작 주파수 (MHz)	1,530	1,110	769
최대 빈 처리량 (Mbin/s)	1,529	2,219	2,307

[0044] 이상에서 몇 가지 실시예를 들어 본 발명을 더욱 상세하게 설명하였으나, 본 발명은 반드시 이러한 실시예로 국한되는 것은 아니고 본 발명의 기술사상을 벗어나지 않는 범위 내에서 다양하게 변형 실시될 수 있다.

부호의 설명

[0045] 100 : 이진화기

200 : 문맥 모델러

300 : 이진 산술 부호화기

Stage 1 : 심볼 확률 및 LPS 범위 후보 결정 스테이지

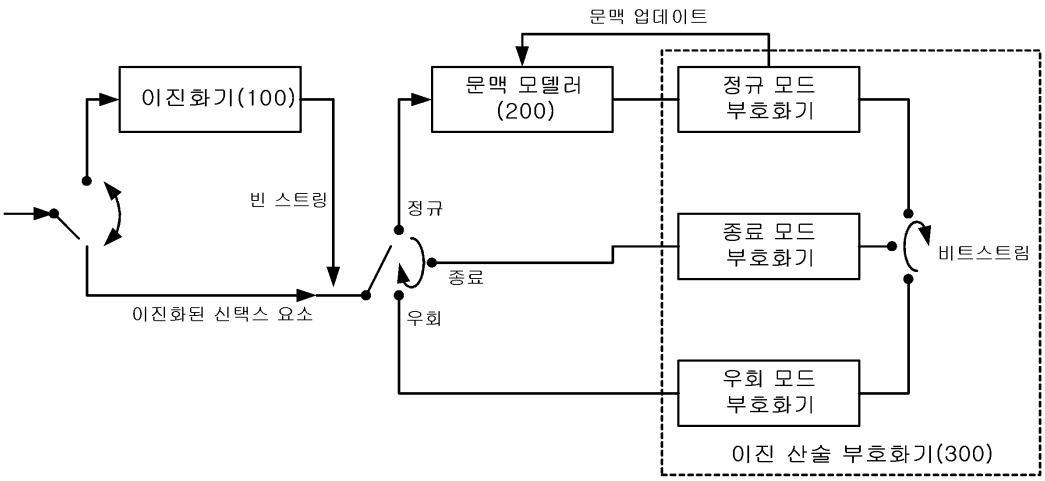
Stage 2 : 범위 업데이트 스테이지

Stage 3 : 최소 경계값 업데이트 스테이지

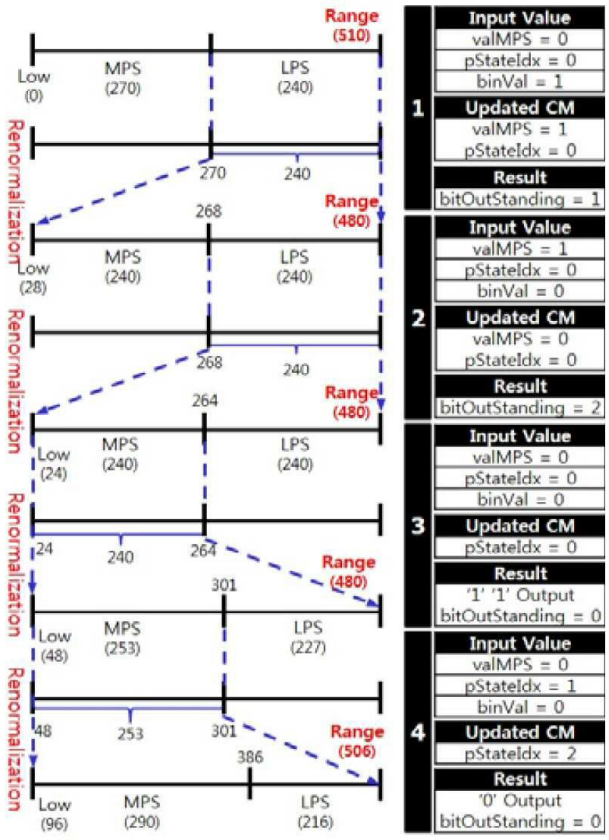
Stage 4 : 비트스트림 생성 스테이지

도면

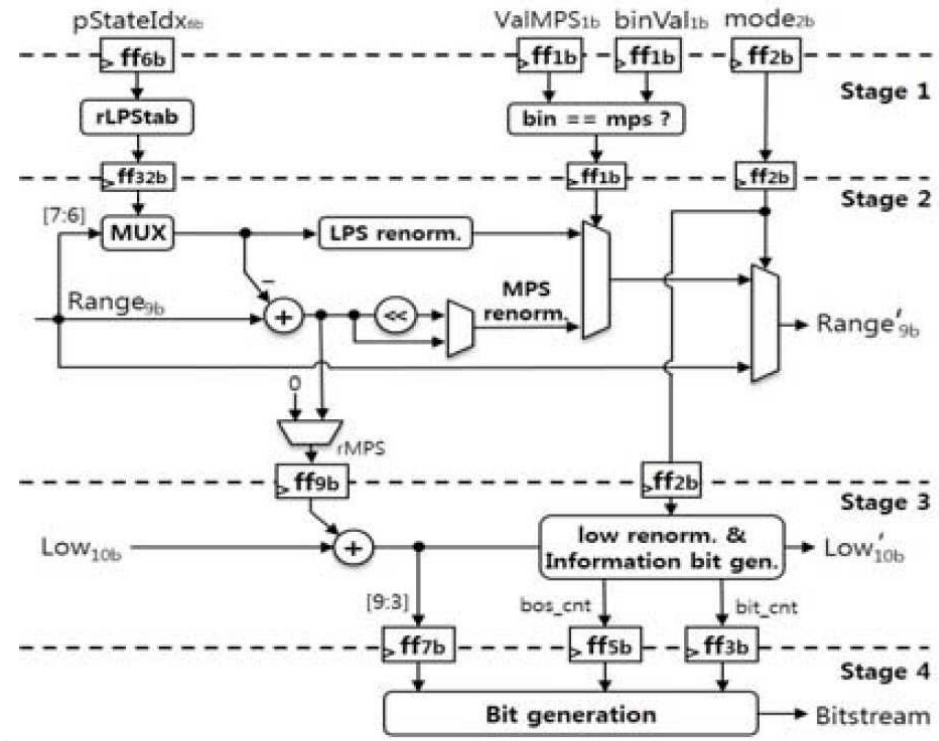
도면1



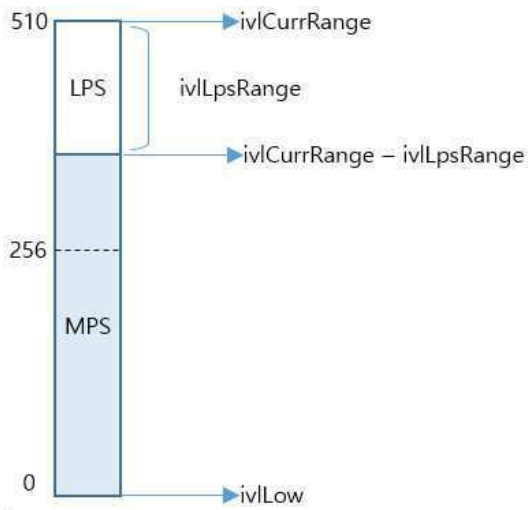
도면2



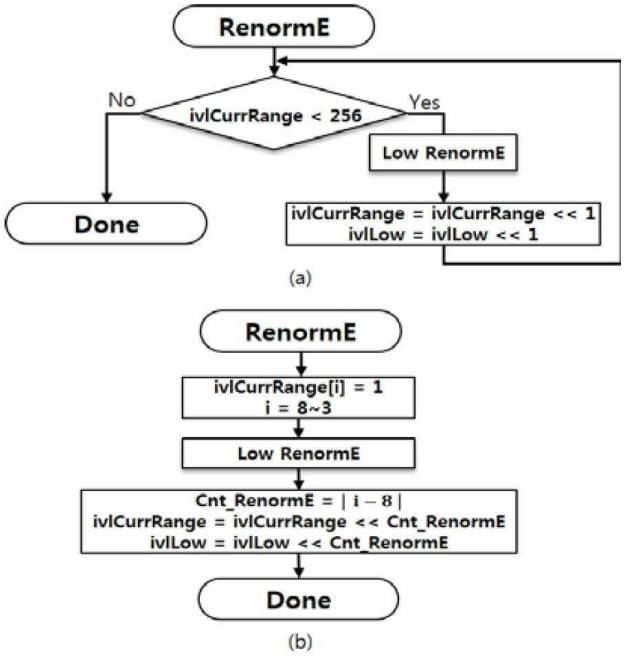
도면3


pStateIdx6b
ff6b
rLPStab
ff32b
[7:6]
MUX
LPS renorm.
Range9b
MPS renorm.
Range'9b
0
rMPS
ff9b
Low10b
ValMPS1b
binVal1b
mode2b
ff1b
ff1b
ff2b
bin == mps ?
ff1b
ff2b
ff2b
low renorm. & Information bit gen.
Low'10b
[9:3]
bos_cnt
bit_cnt
ff7b
ff5b
ff3b
Bit generation
Bitstream
Stage 1
Stage 2
Stage 3
Stage 4

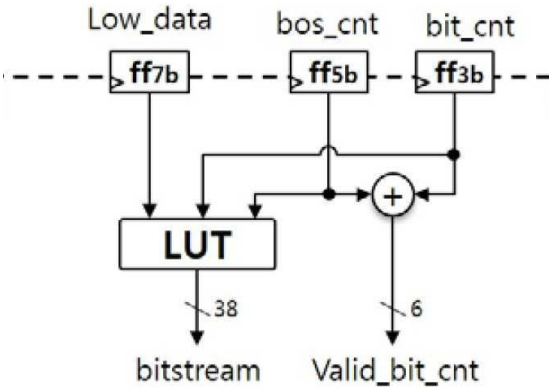
도면4


510
LPS
ivlCurrRange
ivlLpsRange
ivlCurrRange – ivlLpsRange
256
MPS
0
ivlLow

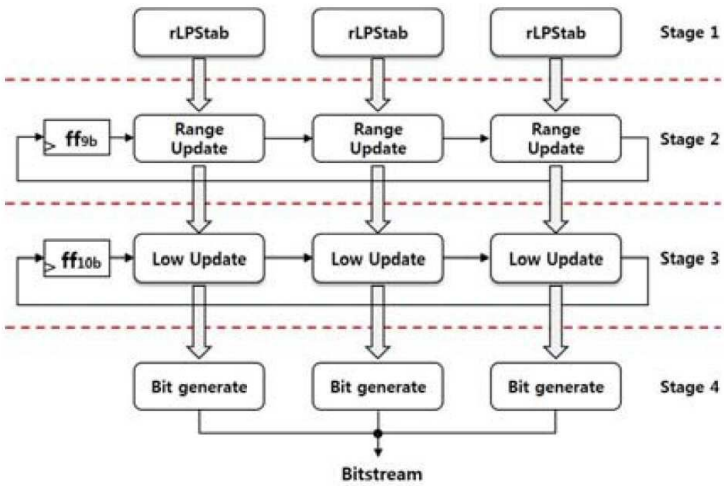
도면5


RenormE
No
ivlCurrRange < 256
Yes
Low RenormE
Done
ivlCurrRange = ivlCurrRange << 1
ivlLow = ivlLow << 1
(a)
RenormE
ivlCurrRange[i] = 1
i = 8~3
Low RenormE
Cnt_RenormE = | i − 8 |
ivlCurrRange = ivlCurrRange << Cnt_RenormE
ivlLow = ivlLow << Cnt_RenormE
Done
(b)

도면6


Low_data
bos_cnt
bit_cnt
ff7b
ff5b
ff3b
LUT
38
6
bitstream
Valid_bit_cnt

도면7


rLPStab
rLPStab
rLPStab
Stage 1
ff9b
Range Update
Range Update
Range Update
Stage 2
ff10b
Low Update
Low Update
Low Update
Stage 3
Bit generate
Bit generate
Bit generate
Stage 4
Bitstream