(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0112562
(43) 공개일자 2020년10월05일

(51) 국제특허분류(Int. Cl.)
H04N 19/625 (2014.01) *H04N 19/119* (2014.01)
H04N 19/13 (2014.01) *H04N 19/176* (2014.01)
H04N 19/82 (2014.01)
(52) CPC특허분류
H04N 19/625 (2015.01)
H04N 19/119 (2015.01)
(21) 출원번호 10-2019-0033228
(22) 출원일자 2019년03월22일
심사청구일자 없음

(71) 출원인
(주)휴맥스
경기도 용인시 처인구 영문로 2 (유방동)
조선대학교산학협력단
광주광역시 동구 필문대로 309 (서석동)
(72) 발명자
이범식
광주광역시 남구 제석로 80번길 5, 제일풍경채 102동 903호
쉬레스따 산딥
광주광역시 동구 지호로 16, 302호 (지산동), 61448
(74) 대리인
홍성진

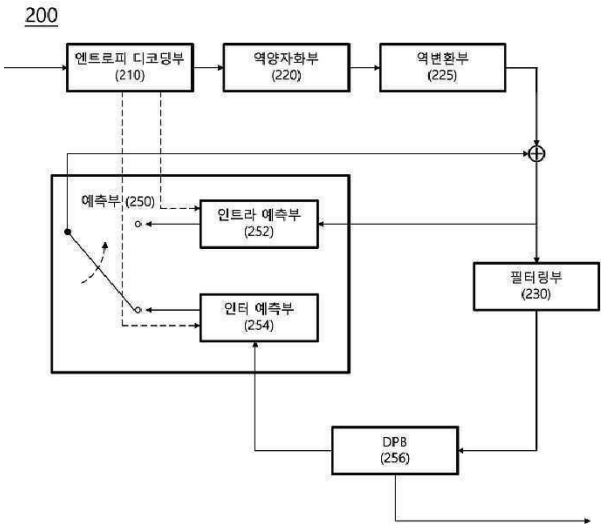
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **DCT-2, DST-7, DCT-8의 8-비트 정수 커널 생성 유도 방법**

(57) 요 약

비디오 신호를 인코딩하거나 디코딩하는 비디오 신호 처리 방법 및 장치가 개시된다.

대 표 도 - 도2



(52) CPC특허분류
H04N 19/13 (2015.01)
H04N 19/176 (2015.01)
H04N 19/82 (2015.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	2018-0-01481
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기술진흥센터
연구사업명	정보통신방송표준개발지원사업
연구과제명	초고해상도를 위한 Versatile Video Coding (VVC) 표준 개발
기 여 율	1/1
과제수행기관명	(주)휴맥스
연구기간	2018.07.01 ~ 2020.12.31

명 세 서

청구범위

청구항 1

비디오 신호의 처리 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 비디오 신호의 처리 방법 및 장치에 관한 것으로, 보다 상세하게는 비디오 신호를 인코딩하거나 디코딩하는 비디오 신호 처리 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 압축 부호화란 디지털화한 정보를 통신 회선을 통해 전송하거나, 저장 매체에 적합한 형태로 저장하기 위한 일련의 신호 처리 기술을 의미한다. 압축 부호화의 대상에는 음성, 영상, 문자 등의 대상이 존재하며, 특히 영상을 대상으로 압축 부호화를 수행하는 기술을 비디오 압축이라고 일컫는다. 비디오 신호에 대한 압축 부호화는 공간적인 상관관계, 시간적인 상관관계, 확률적인 상관관계 등을 고려하여 잉여 정보를 제거함으로써 이루어진다. 그러나 최근의 다양한 미디어 및 데이터 전송 매체의 발전으로 인해, 더욱 고효율의 비디오 신호 처리 방법 및 장치가 요구되고 있다.

발명의 내용

해결하려는 과제

[0003] 본 발명은 비디오 신호의 코딩 효율을 높이기 위한 목적을 가지고 있다.

과제의 해결 수단

[0004] 상기와 같은 과제를 해결하기 위해, 본 발명은 다양한 비디오 신호 처리 장치 및 비디오 신호 처리 방법을 제공한다.

발명의 효과

[0005] 본 발명의 실시예에 따르면 비디오 신호의 코딩 효율이 높아질 수 있다.

도면의 간단한 설명

[0006] 도 1은 본 발명의 실시예에 따른 비디오 신호 인코딩 장치의 개략적인 블록도이다.

도 2는 본 발명의 실시예에 따른 비디오 신호 디코딩 장치의 개략적인 블록도이다.

도 3은 픽쳐 내에서 코딩 트리 유닛이 코딩 유닛들로 분할되는 실시예를 도시한다.

도 4는 쿼드 트리 및 멀티-타입 트리의 분할을 시그널링하는 방법의 일 실시예를 도시한다.

그림 1. 두 개의 행 요소만으로 N=4 DST-7를 구성하는 방법

그림 2. 두 개의 행 요소만으로 N=8 DST-7를 구성하는 방법

그림 3. 두 개의 행 요소와 순열 행렬 3개를 이용한 완전한 DST-7 커널 행렬

그림 4. 4개의 행 요소만으로 N=16 DST-7을 구성하는 방법

그림 5. 4 개의 행 요소와 순열 행렬 4개를 이용한 완전한 N=16 DST-7 커널 행렬

그림 6. N=32의 경우 DST-7의 0, 1, 2, 3, 5, 6 행요소와 순열행렬 G, H, I, J, K 행렬을 이용하여 완전한 DST-7을 구성하는 예시

그림 7. N=32인 경우, 순열행렬 G

그림 8. N=32인 경우, 순열행렬 H

그림 9. N=32인 경우, 순열행렬 I

그림 10. N=32인 경우, 순열행렬 J

그림 11. N=32의 경우, 열행렬 K

그림 12. DST-7의 0번째 행과 순열행렬 G를 이용한 DST-7의 행성분 유도

그림 13. DST-7의 0번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

그림 14. DST-7의 1번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

그림 15. DST-7의 2번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

그림 16. DST-7의 3번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

그림 17. DST-7의 5번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

그림 18. DST-7의 6번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

그림 19. 행 그룹 구성을 효과적으로 하기 위해 첫 번째의 행의 마지막 수와 마지막 행의 첫번째 수를 변경한 예

그림 20. 행렬 U의 분해 형태

그림 21. 제안하는 커널 유도 생성 방법을 사용하였을 때 메모리 사용량

발명을 실시하기 위한 구체적인 내용

[0007] 본 명세서에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도, 관례 또는 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한 특정 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 발명의 설명 부분에서 그 의미를 기재할 것이다. 따라서 본 명세서에서 사용되는 용어는, 단순한 용어의 명칭이 아닌 그 용어가 가진 실질적인 의미와 본 명세서의 전반에 걸친 내용을 토대로 해석되어야 함을 밝혀두고자 한다.

[0008] 본 명세서에서 일부 용어들은 다음과 같이 해석될 수 있다. 코딩은 경우에 따라 인코딩 또는 디코딩으로 해석될 수 있다. 본 명세서에서 비디오 신호의 인코딩(부호화)을 수행하여 비디오 신호 비트스트림을 생성하는 장치는 인코딩 장치 혹은 인코더로 지칭되며, 비디오 신호 비트스트림의 디코딩(복호화)을 수행하여 비디오 신호를 복원하는 장치는 디코딩 장치 혹은 디코더로 지칭된다. 또한, 본 명세서에서 비디오 신호 처리 장치는 인코더 및 디코더를 모두 포함하는 개념의 용어로 사용된다. 정보(information)는 값(values), 파라미터(parameter), 계수(coefficients), 성분(elements) 등을 모두 포함하는 용어로서, 경우에 따라 의미는 달리 해석될 수 있으므로 본 발명은 이에 한정되지 아니한다. '유닛'은 영상 처리의 기본 단위 또는 픽처의 특정 위치를 지칭하는 의미로 사용되며, 루마(luma) 성분과 크로마(chroma) 성분을 모두 포함하는 이미지 영역을 가리킨다. 또한, '블록'은 루마 성분 및 크로마 성분들(즉, Cb 및 Cr) 중 특정 성분을 포함하는 이미지 영역을 가리킨다. 다만, 실시예에 따라서 '유닛', '블록', '파티션' 및 '영역' 등의 용어는 서로 혼용하여 사용될 수 있다. 또한, 본 명세서에서 유닛은 코딩 유닛, 예측 유닛, 변환 유닛을 모두 포함하는 개념으로 사용될 수 있다. 픽처는 필드 혹은 프레임을 가리키며, 실시예에 따라 상기 용어들은 서로 혼용하여 사용될 수 있다.

[0009] 도 1은 본 발명의 일 실시예에 따른 비디오 신호 인코딩 장치의 개략적인 블록도이다. 도 1을 참조하면, 본 발명의 인코딩 장치(100)는 변환부(110), 양자화부(115), 역양자화부(120), 역변환부(125), 필터링부(130), 예측부(150) 및 엔트로피 코딩부(160)를 포함한다.

[0010] 변환부(110)는 입력 받은 비디오 신호와 예측부(150)에서 생성된 예측 신호의 차이인 레지듀얼 신호를 변환하여 변환 계수 값을 획득한다. 예를 들어, 이산 코사인 변환(Discrete Cosine Transform, DCT), 이산 사인 변환(Discrete Sine Transform, DST) 또는 웨이블릿 변환(Wavelet Transform) 등이 사용될 수 있다. 이산 코사인

변환 및 이산 사인 변환은 입력된 픽쳐 신호를 블록 형태로 나누어 변환을 수행하게 된다. 변환에 있어서 변환 영역 내의 값들의 분포와 특성에 따라서 코딩 효율이 달라질 수 있다. 양자화부(115)는 변환부(110)에서 출력된 변환 계수 값을 양자화한다.

[0011] 코딩 효율을 높이기 위하여 픽쳐 신호를 그대로 코딩하는 것이 아니라, 예측부(150)를 통해 이미 코딩된 영역을 이용하여 픽쳐를 예측하고, 예측된 픽쳐에 원본 픽쳐와 예측 픽쳐 간의 레지듀얼 값을 더하여 복원 픽쳐를 획득하는 방법이 사용된다. 인코더와 디코더에서 미스매치가 발생되지 않도록 하기 위해, 인코더에서 예측을 수행할 때에는 디코더에서도 사용 가능한 정보를 사용해야 한다. 이를 위해, 인코더에서는 부호화한 현재 블록을 다시 복원하는 과정을 수행한다. 역양자화부(120)에서는 변환 계수 값을 역양자화하고, 역변환부(125)에서는 역양자화된 변환 계수값을 이용하여 레지듀얼 값을 복원한다. 한편, 필터링부(130)는 복원된 픽쳐의 품질 개선 및 부호화 효율 향상을 위한 필터링 연산을 수행한다. 예를 들어, 디블록킹 필터, 샘플 적응적 오프셋(Sample Adaptive Offset, SAO) 및 적응적 루프 필터 등이 포함될 수 있다. 필터링을 거친 픽쳐는 출력되거나 참조 픽쳐로 이용하기 위하여 복호 픽쳐 버퍼(Decoded Picture Buffer, DPB, 156)에 저장된다.

[0012] 예측부(150)는 인트라 예측부(152)와 인터 예측부(154)를 포함한다. 인트라 예측부(152)는 현재 픽쳐 내에서 인트라(intra) 예측을 수행하며, 인터 예측부(154)는 복호 픽쳐 버퍼(156)에 저장된 참조 픽쳐를 이용하여 현재 픽쳐를 예측하는 인터(inter) 예측을 수행한다. 인트라 예측부(152)는 현재 픽쳐 내의 복원된 샘플들로부터 인트라 예측을 수행하여, 인트라 부호화 정보를 엔트로피 코딩부(160)에 전달한다. 인트라 부호화 정보는 인트라 예측 모드, MPM(Most Probable Mode) 플래그, MPM 인덱스 중 적어도 하나를 포함할 수 있다. 인터 예측부(154)는 모션 추정부(154a) 및 모션 보상부(154b)를 포함하여 구성될 수 있다. 모션 추정부(154a)는 복원된 참조 픽쳐의 특정 영역을 참조하여 현재 영역의 모션 벡터값을 획득한다. 모션 추정부(154a)는 참조 영역에 대한 모션 정보(참조 픽쳐 인덱스, 모션 벡터 정보 등)를 엔트로피 코딩부(160)로 전달한다. 모션 보상부(154b)는 모션 추정부(154a)에서 전달된 모션 벡터값을 이용하여 모션 보상을 수행한다. 인터 예측부(154)는 참조 영역에 대한 모션 정보를 포함하는 인터 부호화 정보를 엔트로피 코딩부(160)에 전달한다.

[0013] 위와 같은 픽쳐 예측이 수행될 경우, 변환부(110)는 원본 픽쳐와 예측 픽쳐 간의 레지듀얼 값을 변환하여 변환 계수 값을 획득한다. 이때, 변환은 픽쳐 내에서 특정 블록 단위로 수행될 수 있으며, 특정 블록의 크기는 기 설정된 범위 내에서 가변할 수 있다. 양자화부(115)는 변환부(110)에서 생성된 변환 계수 값을 양자화하여 엔트로피 코딩부(160)로 전달한다.

[0014] 엔트로피 코딩부(160)는 양자화된 변환 계수, 인트라 부호화 정보, 및 인터 부호화 정보 등을 엔트로피 코딩하여 비디오 신호 비트스트림을 생성한다. 엔트로피 코딩부(160)에서는 가변 길이 코딩(Variable Length Coding, VLC) 방식과 산술 코딩(arithmetic coding) 방식 등이 사용될 수 있다. 가변 길이 코딩(VLC) 방식은 입력되는 심볼들을 연속적인 코드워드로 변환하는데, 코드워드의 길이는 가변적일 수 있다. 예를 들어, 자주 발생하는 심볼들을 짧은 코드워드로, 자주 발생하지 않은 심볼들은 긴 코드워드로 표현하는 것이다. 가변 길이 코딩 방식으로서 컨텍스트 기반 적응형 가변 길이 코딩(Context-based Adaptive Variable Length Coding, CAVLC) 방식이 사용될 수 있다. 산술 코딩은 연속적인 데이터 심볼들을 하나의 소수로 변환하는데, 산술 코딩은 각 심볼을 표현하기 위하여 필요한 최적의 소수 비트를 얻을 수 있다. 산술 코딩으로서 컨텍스트 기반 적응형 산술 부호화(Context-based Adaptive Binary Arithmetic Code, CABAC)가 이용될 수 있다.

[0015] 상기 생성된 비트스트림은 NAL(Network Abstraction Layer) 유닛을 기본 단위로 캡슐화 된다. NAL 유닛은 부호화된 정수 개의 코딩 트리 유닛(coding tree unit)을 포함한다. 비디오 디코더에서 비트스트림을 복호화하기 위해서는 먼저 비트스트림을 NAL 유닛 단위로 분리한 후, 분리 된 각각의 NAL 유닛을 복호화해야 한다. 한편, 비디오 신호 비트스트림의 복호화를 위해 필요한 정보들은 픽쳐 파라미터 세트(Picture Parameter Set, PPS), 시퀀스 파라미터 세트(Sequence Parameter Set, SPS), 비디오 파라미터 세트(Video Parameter Set, VPS) 등과 같은 상위 레벨 세트의 RBSP(Raw Byte Sequence Payload)를 통해 전송될 수 있다.

[0016] 한편, 도 1의 블록도는 본 발명의 일 실시예에 따른 인코딩 장치(100)를 나타낸 것으로서, 분리하여 표시된 블록들은 인코딩 장치(100)의 엘리먼트들을 논리적으로 구별하여 도시한 것이다. 따라서 전술한 인코딩 장치(100)의 엘리먼트들은 디바이스의 설계에 따라 하나의 칩으로 또는 복수의 칩으로 장착될 수 있다. 일 실시예에 따르면, 전술한 인코딩 장치(100)의 각 엘리먼트의 동작은 프로세서(미도시)에 의해 수행될 수 있다.

[0017] 도 2는 본 발명의 일 실시예에 따른 비디오 신호 디코딩 장치(200)의 개략적인 블록도이다. 도 2를 참조하면 본 발명의 디코딩 장치(200)는 엔트로피 디코딩부(210), 역양자화부(220), 역변환부(225), 필터링부(230) 및 예측부(250)를 포함한다.

[0018] 엔트로피 디코딩부(210)는 비디오 신호 비트스트림을 엔트로피 디코딩하여, 각 영역에 대한 변환 계수, 인트라 부호화 정보, 인터 부호화 정보 등을 추출한다. 역양자화부(220)는 엔트로피 디코딩된 변환 계수를 역양자화하고, 역변환부(225)는 역양자화된 변환 계수를 이용하여 레지듀얼 값을 복원한다. 비디오 신호 처리 장치(200)는 역변환부(225)에서 획득된 레지듀얼 값을 예측부(250)에서 획득된 예측값과 합산하여 원래의 화소값을 복원한다.

[0019] 한편, 필터링부(230)는 픽처에 대한 필터링을 수행하여 화질을 향상시킨다. 여기에는 블록 왜곡 현상을 감소시키기 위한 디블록킹 필터 및/또는 픽처 전체의 왜곡 제거를 위한 적응적 루프 필터 등이 포함될 수 있다. 필터링을 거친 픽처는 출력되거나 다음 픽처에 대한 참조 픽처로 이용하기 위하여 복호 픽처 버퍼(DPB, 256)에 저장된다.

[0020] 예측부(250)는 인트라 예측부(252) 및 인터 예측부(254)를 포함한다. 예측부(250)는 전술한 엔트로피 디코딩부(210)를 통해 복호화된 부호화 타입, 각 영역에 대한 변환 계수, 인트라/인터 부호화 정보 등을 활용하여 예측 픽처를 생성한다. 복호화가 수행되는 현재 블록을 복원하기 위해서, 현재 블록이 포함된 현재 픽처 또는 다른 픽처들의 복호화된 영역이 이용될 수 있다. 복원에 현재 픽처만을 이용하는, 즉 인트라 예측만을 수행하는 픽처(또는, 타일/슬라이스)를 인트라 픽처 또는 I 픽처(또는, 타일/슬라이스), 인트라 예측과 인터 예측을 모두 수행할 수 있는 픽처(또는, 타일/슬라이스)를 인터 픽처(또는, 타일/슬라이스)라고 한다. 인터 픽처(또는, 타일/슬라이스) 중 각 블록의 샘플값들을 예측하기 위하여 최대 하나의 모션 벡터 및 참조 픽처 인덱스를 이용하는 픽처(또는, 타일/슬라이스)를 예측 픽처(predictive picture) 또는 P 픽처(또는, 타일/슬라이스)라고 하며, 최대 두 개의 모션 벡터 및 참조 픽처 인덱스를 이용하는 픽처(또는, 타일/슬라이스)를 쌍예측 픽처(Bi-predictive picture) 또는 B 픽처(또는, 타일/슬라이스) 라고 한다. 다시 말해서, P 픽처(또는, 타일/슬라이스)는 각 블록을 예측하기 위해 최대 하나의 모션 정보 세트를 이용하고, B 픽처(또는, 타일/슬라이스)는 각 블록을 예측하기 위해 최대 두 개의 모션 정보 세트를 이용한다. 여기서, 모션 정보 세트는 하나 이상의 모션 벡터와 하나의 참조 픽처 인덱스를 포함한다.

[0021] 인트라 예측부(252)는 인트라 부호화 정보 및 현재 픽처 내의 복원된 샘플들을 이용하여 예측 블록을 생성한다. 전술한 바와 같이, 인트라 부호화 정보는 인트라 예측 모드, MPM(Most Probable Mode) 플래그, MPM 인덱스 중 적어도 하나를 포함할 수 있다. 인트라 예측부(252)는 현재 블록의 좌측 및/또는 상측에 위치한 복원된 픽셀들을 참조 픽셀들로 이용하여 현재 블록의 픽셀값들을 예측한다. 일 실시예에 따르면, 참조 픽셀들은 현재 블록의 좌측 경계에 인접한 픽셀들 및/또는 상측 경계에 인접한 픽셀들일 수 있다. 다른 실시예에 따르면, 참조 픽셀들은 현재 블록의 주변 블록의 픽셀들 중 현재 블록의 좌측 경계로부터 기 설정된 거리 이내에 인접한 픽셀들 및/또는 현재 블록의 상측 경계로부터 기 설정된 거리 이내에 인접한 픽셀들일 수 있다. 이때, 현재 블록의 주변 블록은 현재 블록에 인접한 좌측(L) 블록, 상측(A) 블록, 하좌측(Below Left, BL) 블록, 상우측(Above Right, AR) 블록 또는 상좌측(Above Left, AL) 블록 중 적어도 하나를 포함할 수 있다.

[0022] 인터 예측부(254)는 복호 픽처 버퍼(256)에 저장된 참조 픽처 및 인터 부호화 정보를 이용하여 예측 블록을 생성한다. 인터 부호화 정보는 참조 블록에 대한 현재 블록의 모션 정보(참조 픽처 인덱스, 모션 벡터 정보 등)을 포함할 수 있다. 인터 예측에는 L0 예측, L1 예측 및 쌍예측(Bi-prediction)이 있을 수 있다. L0 예측은 L0 픽처 리스트에 포함된 1개의 참조 픽처를 이용한 예측이고, L1 예측은 L1 픽처 리스트에 포함된 1개의 참조 픽처를 이용한 예측을 의미한다. 이를 위해서는 1세트의 모션 정보(예를 들어, 모션 벡터 및 참조 픽처 인덱스)가 필요할 수 있다. 쌍예측 방식에서는 최대 2개의 참조 영역을 이용할 수 있는데, 이 2개의 참조 영역은 동일한 참조 픽처에 존재할 수도 있고, 서로 다른 픽처에 각각 존재할 수도 있다. 즉, 쌍예측 방식에서는 최대 2세트의 모션 정보(예를 들어, 모션 벡터 및 참조 픽처 인덱스)가 이용될 수 있는데, 2개의 모션 벡터가 동일한 참조 픽처 인덱스에 대응될 수도 있고 서로 다른 참조 픽처 인덱스에 대응될 수도 있다. 이때, 참조 픽처들은 시간적으로 현재 픽처 이전이나 이후 모두에 표시(또는 출력)될 수 있다.

[0023] 인터 예측부(254)는 모션 벡터 및 참조 픽처 인덱스를 이용하여 현재 블록의 참조 블록을 획득할 수 있다. 상기 참조 블록은 참조 픽처 인덱스에 대응하는 참조 픽처 내에 존재한다. 또한, 모션 벡터에 의해서 특정된 블록의 픽셀값 또는 이의 보간(interpolation)된 값이 현재 블록의 예측값(predictor)으로 이용될 수 있다. 서브펠(sub-pel) 단위의 픽셀 정확도를 갖는 모션 예측을 위하여 이를 테면, 루마 신호에 대하여 8-탭 보간 필터가, 크로마 신호에 대하여 4-탭 보간 필터가 사용될 수 있다. 다만, 서브펠 단위의 모션 예측을 위한 보간 필터는 이에 한정되지 않는다. 이와 같이 인터 예측부(254)는 이전에 복원된 픽처로부터 현재 유닛의 텍스쳐를 모션 정보를 이용하여 예측하는 모션 보상(motion compensation)을 수행한다.

[0024] 상기 인트라 예측부(252) 또는 인터 예측부(254)로부터 출력된 예측값, 및 역변환부(225)로부터 출력된 레지듀얼 값이 더해져서 복원된 비디오 픽쳐가 생성된다. 즉, 비디오 신호 디코딩 장치(200)는 예측부(250)에서 생성된 예측 블록과 역변환부(225)로부터 획득된 레지듀얼을 이용하여 현재 블록을 복원한다.

[0025] 한편, 도 2의 블록도는 본 발명의 일 실시예에 따른 디코딩 장치(200)를 나타낸 것으로서, 분리하여 표시된 블록들은 디코딩 장치(200)의 엘리먼트들을 논리적으로 구별하여 도시한 것이다. 따라서 전술한 디코딩 장치(200)의 엘리먼트들은 디바이스의 설계에 따라 하나의 칩으로 또는 복수의 칩으로 장착될 수 있다. 일 실시예에 따르면, 전술한 디코딩 장치(200)의 각 엘리먼트의 동작은 프로세서(미도시)에 의해 수행될 수 있다.

[0026] 도 3은 픽쳐 내에서 코딩 트리 유닛(Coding Tree Unit, CTU)이 코딩 유닛들(Coding Units, CUs)로 분할되는 실시예를 도시한다. 비디오 신호의 코딩 과정에서, 픽쳐는 코딩 트리 유닛(CTU)들의 시퀀스로 분할될 수 있다. 코딩 트리 유닛은 루마(luma) 샘플들의 NXN 블록과, 이에 대응하는 크로마(chroma) 샘플들의 2개의 블록들로 구성된다. 코딩 트리 유닛은 복수의 코딩 유닛들로 분할될 수 있다. 코딩 유닛은 상기에서 설명한 비디오 신호의 처리 과정, 즉 인트라/인터 예측, 변환, 양자화 및/또는 엔트로피 코딩 등의 과정에서 픽쳐를 처리하기 위한 기본 단위를 가리킨다. 하나의 픽쳐 내에서 코딩 유닛의 크기 및 모양은 일정하지 않을 수 있다. 코딩 유닛은 정사각형 혹은 직사각형의 모양을 가질 수 있다. 직사각형 코딩 유닛(혹은, 직사각형 블록)은 수직 코딩 유닛(혹은, 수직 블록)과 수평 코딩 유닛(혹은, 수평 블록)을 포함한다. 본 명세서에서, 수직 블록은 높이가 너비보다 큰 블록이며, 수평 블록은 너비가 높이보다 큰 블록이다. 또한, 본 명세서에서 정사각형이 아닌(non-square) 블록은 직사각형 블록을 가리킬 수 있지만, 본 발명은 이에 한정되지 않는다.

[0027] 도 3을 참조하면, 코딩 트리 유닛은 먼저 쿼드 트리(Quad Tree, QT) 구조로 분할된다. 즉, 쿼드 트리 구조에서 2NX2N 크기를 가지는 하나의 노드는 NXN 크기를 가지는 네 개의 노드들로 분할될 수 있다. 본 명세서에서 쿼드 트리는 4진(quaternary) 트리로도 지칭될 수 있다. 쿼드 트리 분할은 재귀적으로 수행될 수 있으며, 모든 노드들이 동일한 깊이로 분할될 필요는 없다.

[0028] 한편, 전술한 쿼드 트리의 리프 노드(leaf node)는 멀티-타입 트리(Multi-Type Tree, MTT) 구조로 더욱 분할될 수 있다. 본 발명의 실시예에 따르면, 멀티 타입 트리 구조에서는 하나의 노드가 수평 혹은 수직 분할의 2진(binary, 바이너리) 혹은 3진(ternary, 터너리) 트리 구조로 분할될 수 있다. 즉, 멀티-타입 트리 구조에는 수직 바이너리 분할, 수평 바이너리 분할, 수직 터너리 분할 및 수평 터너리 분할의 4가지 분할 구조가 존재한다. 본 발명의 실시예에 따르면, 상기 각 트리 구조에서 노드의 너비 및 높이는 모두 2의 거듭제곱 값을 가질 수 있다. 예를 들어, 바이너리 트리(Binary Tree, BT) 구조에서, 2NX2N 크기의 노드는 수직 바이너리 분할에 의해 2개의 NX2N 노드들로 분할되고, 수평 바이너리 분할에 의해 2개의 2NXN 노드들로 분할될 수 있다. 또한, 터너리 트리(Ternary Tree, TT) 구조에서, 2NX2N 크기의 노드는 수직 터너리 분할에 의해 (N/2)X2N, NX2N 및 (N/2)X2N의 노드들로 분할되고, 수평 바이너리 분할에 의해 2NX(N/2), 2NXN 및 2NX(N/2)의 노드들로 분할될 수 있다. 이러한 멀티-타입 트리 분할은 재귀적으로 수행될 수 있다.

[0029] 멀티-타입 트리의 리프 노드는 코딩 유닛이 될 수 있다. 코딩 유닛이 최대 변환 길이에 비해 너무 크지 않으면, 해당 코딩 유닛은 더 이상의 분할 없이 예측 및 변환의 단위로 사용된다. 한편, 전술한 쿼드 트리 및 멀티-타입 트리에서 다음의 파라메터들 중 적어도 하나가 사전에 정의되거나 PPS, SPS, VPS 등과 같은 상위 레벨 세트의 RBSP를 통해 전송될 수 있다. 1) CTU 크기: 쿼드 트리의 루트 노드(root node) 크기, 2) 최소 QT 크기(MinQtSize): 허용된 최소 QT 리프 노드 크기, 3) 최대 BT 크기(MaxBtSize): 허용된 최대 BT 루트 노드 크기, 4) 최대 TT 크기(MaxTtSize): 허용된 최대 TT 루트 노드 크기, 5) 최대 MTT 깊이(MaxMttDepth): QT의 리프 노드로부터의 MTT 분할의 최대 허용 깊이, 6) 최소 BT 크기(MinBtSize): 허용된 최소 BT 리프 노드 크기, 7) 최소 TT 크기(MinTtSize): 허용된 최소 TT 리프 노드 크기.

[0030] 도 4는 쿼드 트리 및 멀티-타입 트리의 분할을 시그널링하는 방법의 일 실시예를 도시한다. 전술한 쿼드 트리 및 멀티-타입 트리의 분할을 시그널링하기 위해 기 설정된 플래그들이 사용될 수 있다. 도 4를 참조하면, 쿼드 트리 노드의 분할 여부를 지시하는 플래그 'qt_split_flag', 멀티-타입 트리 노드의 분할 여부를 지시하는 플래그 'mtt_split_flag', 멀티-타입 트리 노드의 분할 방향을 지시하는 플래그 'mtt_split_vertical_flag' 또는 멀티-타입 트리 노드의 분할 모양을 지시하는 플래그 'mtt_split_binary_flag' 중 적어도 하나가 사용될 수 있다.

[0031] 본 발명의 실시예에 따르면, 코딩 트리 유닛은 쿼드 트리의 루트 노드이며, 쿼드 트리 구조로 우선 분할될 수 있다. 쿼드 트리 구조에서는 각각의 노드 'QT_node' 별로 'qt_split_flag'가 시그널링된다. 'qt_split_flag'의 값이 1일 경우 해당 노드는 4개의 정사각형 노드들로 분할되며, 'qt_split_flag'의 값이 0일 경우 해당 노드는

쿼드 트리의 리프 노드 'QT_leaf_node'가 된다.

[0032] 각각의 쿼드 트리 리프 노드 'QT_leaf_node'는 멀티-타입 트리 구조로 더 분할될 수 있다. 멀티-타입 트리 구조에서는 각각의 노드 'MTT_node' 별로 'mtt_split_flag'가 시그널링된다. 'mtt_split_flag'의 값이 1일 경우 해당 노드는 복수의 직사각형 노드들로 분할되며, 'mtt_split_flag'의 값이 0일 경우 해당 노드는 멀티-타입 트리의 리프 노드 'MTT_leaf_node'가 된다. 멀티-타입 트리 노드 'MTT_node'가 복수의 직사각형 노드들로 분할될 경우(즉, 'mtt_split_flag'의 값이 1일 경우), 노드 'MTT_node'를 위한 'mtt_split_vertical_flag' 및 'mtt_split_binary_flag'가 추가로 시그널링될 수 있다. 'mtt_split_vertical_flag'의 값이 1일 경우 노드 'MTT_node'의 수직 분할이 지시되며, 'mtt_split_vertical_flag'의 값이 0일 경우 노드 'MTT_node'의 수평 분할이 지시된다. 또한, 'mtt_split_binary_flag'의 값이 1일 경우 노드 'MTT_node'는 2개의 직사각형 노드들로 분할되며, 'mtt_split_binary_flag'의 값이 0일 경우 노드 'MTT_node'는 3개의 직사각형 노드들로 분할된다.

[0034] 본 발명에서는 DST-7 변환 커널을 사용하여 다른 크기 및 다른 종류의 커널을 생성 유도하는 통합 변환 커널 생성 유도 방법과 이에 따라 생성된 DST-7, DCT-8 정수 변환 커널을 제안한다. DST-7, DST-8변환 커널은 (1)과 (2)와 같이 각각 정의하는 변환 커널이다.

$$\mathrm{DST}-7=[S_N^{VII}]_{n,k}=\frac{2}{\sqrt{2N+1}}\sin\left(\frac{\pi(2k+1)(n+1)}{2N+1}\right)$$

[0035] (1)

$$\mathrm{DCT}-8=[C_N^{VIII}]_{n,k}=\frac{2}{\sqrt{2(N-1)}}cos\left(\frac{\pi(2k+1)(2n+1)}{4N-2}\right)$$

[0036] (2)

[0037] 여기서 N은 변환 커널의 크기, n, k는 n번째 행, k번째 열을 나타내는 행과 열의 인덱스를 의미한다. N=4 이면 4x4 크기의 변환 커널이다. DST-7 변환 커널은 화면 내 예측 시 예측 잔차 신호의 특성에 매우 유리한 변환 커널이므로 HEVC 표준 코덱에서 화면 내 잔차 블록 4x4 크기에 적용하여 표준 변환 커널로 사용해 오고 있다. 그러나 이후 압축 성능의 향상을 위해 4x4 화면내 잔차 블록 뿐만 아니라 모든 여러 크기의 변환 블록에 사용하는 것을 고려하고 있다. 또한 DST-7 이외에 DCT-8도 역시 커널의 특성으로 인해 압축 성능 향상에 도움이 되므로 표준 변환 커널로서 사용될 수 있다. 그러나 이렇게 4x4, 8x8, … , NxN 등 다양한 변환 블록의 크기와 DST-7, DCT-8, DCT-2 등 많은 종류의 변환 커널이 사용될 경우 이로 인해 변환 커널의 관리에 어려움이 있을 뿐만 아니라 부호화기와 복호화기에 변환 커널의 저장으로 인한 메모리의 공간이 크게 소모될 수 있다. 본 발명에서는 화면내 부호화에 절대적으로 유리하여 압축 성능 향상에 크기 도움이 되는 DST-7변환 커널을 기반으로 다른 크기와 변환 커널과 변환 종류를 유도하는 통합 변환 커널 생성 및 유도 방법과 이를 이용하여 유도된 정수 변환 DSST-7, DCT-8 변환 커널에 대하여 설명한다.

[0038] 비디오 코덱에서는 변환 커널이 정수화하여 구현되기 때문에 본 발명에서는 DST-7 정수 변환 커널을 기반으로 기술한다. 본 발명의 실시예에서는 각 행렬 요소가 8비트로 표현되는 정수 변환 커널에 대한 예시를 보인다.

[0039] N=4 크기의 DST-7: (1)의 DST-7 실수 변환 커널에 상수를 곱하여 가장 가까운 정수로 매핑하면 (3)과 같은 정수 변환 커널이 된다.

$$\mathrm{DST}-7=\begin{bmatrix}29 & 55 & 74 & 84\\ 74 & 74 & 0 & -74\\ 84 & -29 & -74 & 55\\ 55 & -84 & 74 & -29\end{bmatrix}$$

[0040] (3)

[0041] N=8, N=16, N=32 와 같은 경우와 또 다른 변환 커널인 DCT-2, DCT-8의 커널의 경우에도 모든 블록 크기에 대하여 커널을 메모리에 저장하여 사용한다. 이 경우에 모든 변환 커널의 크기와 종류에 대하여 모든 커널 요소를 저장하여야 하기 때문에 이로 인한 메모리의 낭비가 발생한다. 이 문제를 해결하기 위하여 본 발명에서는 모든 변환 커널의 크기와 종류를 모두 저장하지 않고 일부분의 커널 요소의 저장만으로 모든 변환 커널의 크기와 종류를 유도 생성하는 방법과 이 방법을 기반으로 생성된 정수 변환 커널에 대하여 제안한다.

[0042] 먼저 (3)에서 정의된 N=4 DST-7 커널에서 첫 번째와 두 번째 행의 요소를 취하면 14 크기의 A, B 행렬이 된다.

$$\boldsymbol{A} = [29 \quad 55 \quad 74 \quad 84]$$

[0043] (4)

$$\mathbf{B} = [74 \quad 74 \quad 0 \quad -74]$$

[0044] (5)

[0045] 첫 번째와 두번째 행의 원소 14 행렬과 아래 순열 또는 치환 행렬 (permutation matrix) C, D행렬을 곱한다.

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

[0046] (6)

$$\mathbf{D} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

[0047] (7)

$$\mathbf{A} \times \mathbf{C} \quad \mathbf{A} \times \mathbf{D} \quad \mathbf{B} \times \mathbf{C} \quad \mathbf{B} \times \mathbf{D}$$

[0048] 그러면 아래와 같은 14 크기의 결과 행렬 , , , 가 각각 생성된다.

$$\mathbf{A} \times \mathbf{C} = [55 \quad -84 \quad 74 \quad -29]$$

[0049] (8)

$$\mathbf{A} \times \mathbf{D} = [84 \quad -29 \quad -74 \quad 55]$$

[0050] (9)

$$\mathbf{B} \times \mathbf{C} = [74 \quad 74 \quad 0 \quad -74]$$

[0051] (10)

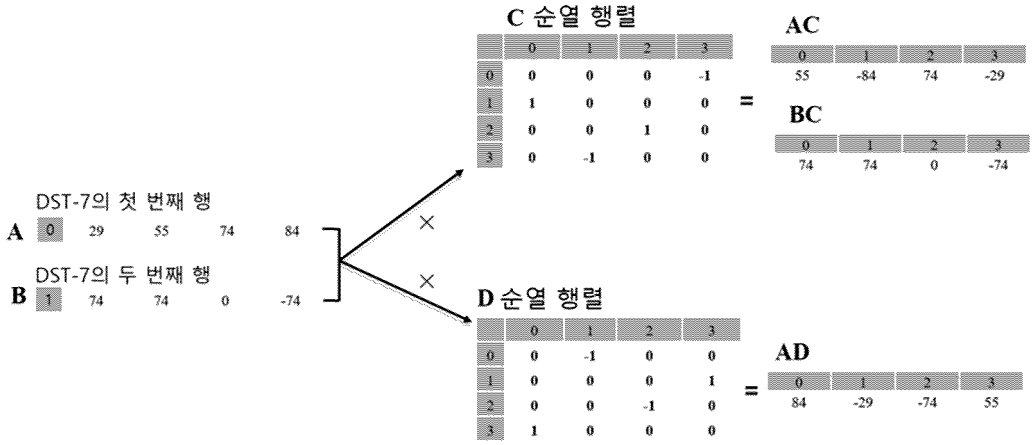
$$\mathbf{B} \times \mathbf{D} = [-74 \quad -74 \quad 0 \quad 74]$$

[0052] (11)

[0054] N=4 DST-7 행렬의 4개의 열 가운데서 2개의 행에 정의된 순열 행렬 C, D를 각각 곱하여 주면 44 전체 DST-7 행렬을 구성할 수 가 있다. N=4 DST-7 행렬의 각 행의 원소를 구하는 방법을 (12)와 같이 요약하였다.

$$\mathrm{DST}-7 = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} \begin{matrix} \boldsymbol{A} \\ \boldsymbol{B} \times \boldsymbol{C} \\ \boldsymbol{A} \times \boldsymbol{D} \\ \boldsymbol{A} \times \boldsymbol{C} \end{matrix}$$

[0055] (12)



[0058]

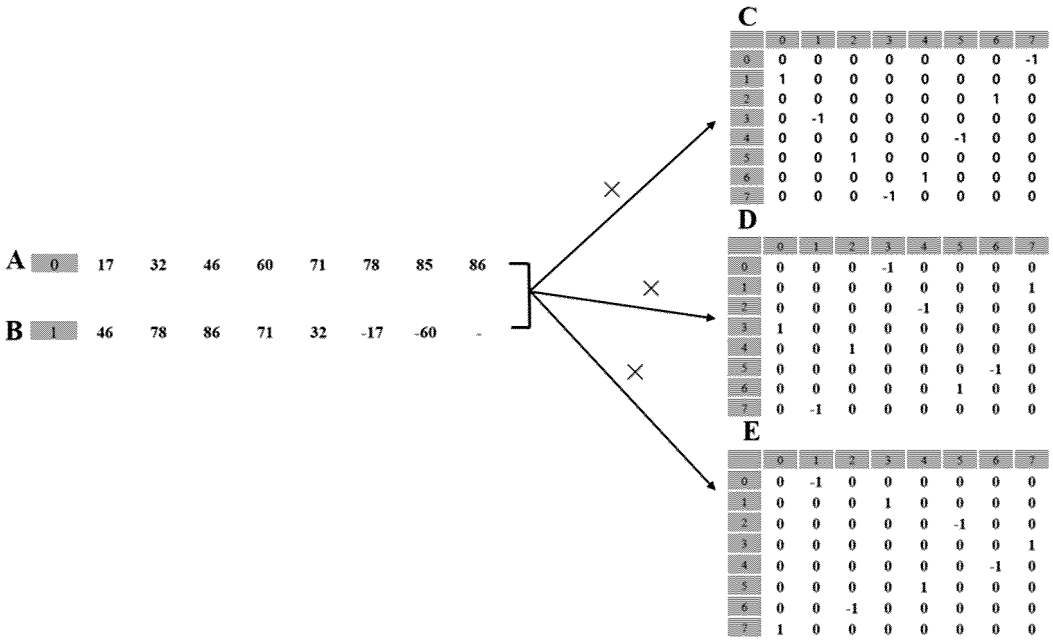
[0059] 그림 1. 두 개의 행 요소만으로 N=4 DST-7를 구성하는 방법

[0060] 그림 1은 N=4 DST-7에 대해 두 개의 행 요소만으로 완전한 N-4, DST-7을 구성하는 방법을 보여준다. 적은 수의 행렬의 행으로 완전한 DST-7 커널 행렬을 구성할 때 순열 행렬의 개수는 DST-7 변환 행렬의 크기와 관련이

$$N = 2^l$$

있다. 예를 들어 크기 N의 DST-7에 대해 이라고 할 때 순열 행렬의 개수는 이 된다.

[0061] N=8 DST-7에 대해서도 동일한 방법을 적용할 수 있다. 정수형으로 변환된 N=8 DST-7에 대하여 몇 개의 선택된 행만으로 전체 DST-7 변환 커널을 N=4일 때 동일한 방법을 사용하여 유도한다.



[0064]

[0065] 그림 2. 두 개의 행 요소만으로 N=8 DST-7를 구성하는 방법

[0067] N=8의 경우 순열 행렬은 그림 2에서처럼 **C**, **D**, **E**로 정의한다. 그림 2는 N=8, DST-7에 대해 첫 번째와 두 번째의 행 만으로 완전한 N=8 DST-7 커널 행렬을 구하는 방법을 보여준다.

		2		4				8	
0	65	127	185	237	280	314	338	350	A
7	127	-237	314	-350	338	-280	185	-65	AC
6	237	-350	280	-65	-185	338	-314	127	AD
4	350	-65	-338	127	314	-185	-280	237	AE
1	185	314	350	280	127	-65	-237	-338	B
5	314	-280	-65	338	-237	-127	350	-185	BC
2	280	338	127	-185	-350	-237	65	314	BD
3	338	185	-237	-314	65	350	127	-280	BE

[0069]

[0070] 그림 3. 두 개의 행 요소와 순열 행렬 3개를 이용한 완전한 DST-7 커널 행렬

$$l = 3 \quad 2^3 = 8$$

[0072] 그림 3에서 보는 것처럼 N=8 DST-7의 경우 DST-7의 첫 번째 두 번째 행과 ( )개의 순열 행렬 C, D, E를 이용하여 DST-7의 각 행의 위치에 배치하면 완전한 DST-7 행렬을 구할 수 있다.

[0073] N=16 DST-7에 대한 방법을 그림 4에서 보여준다.

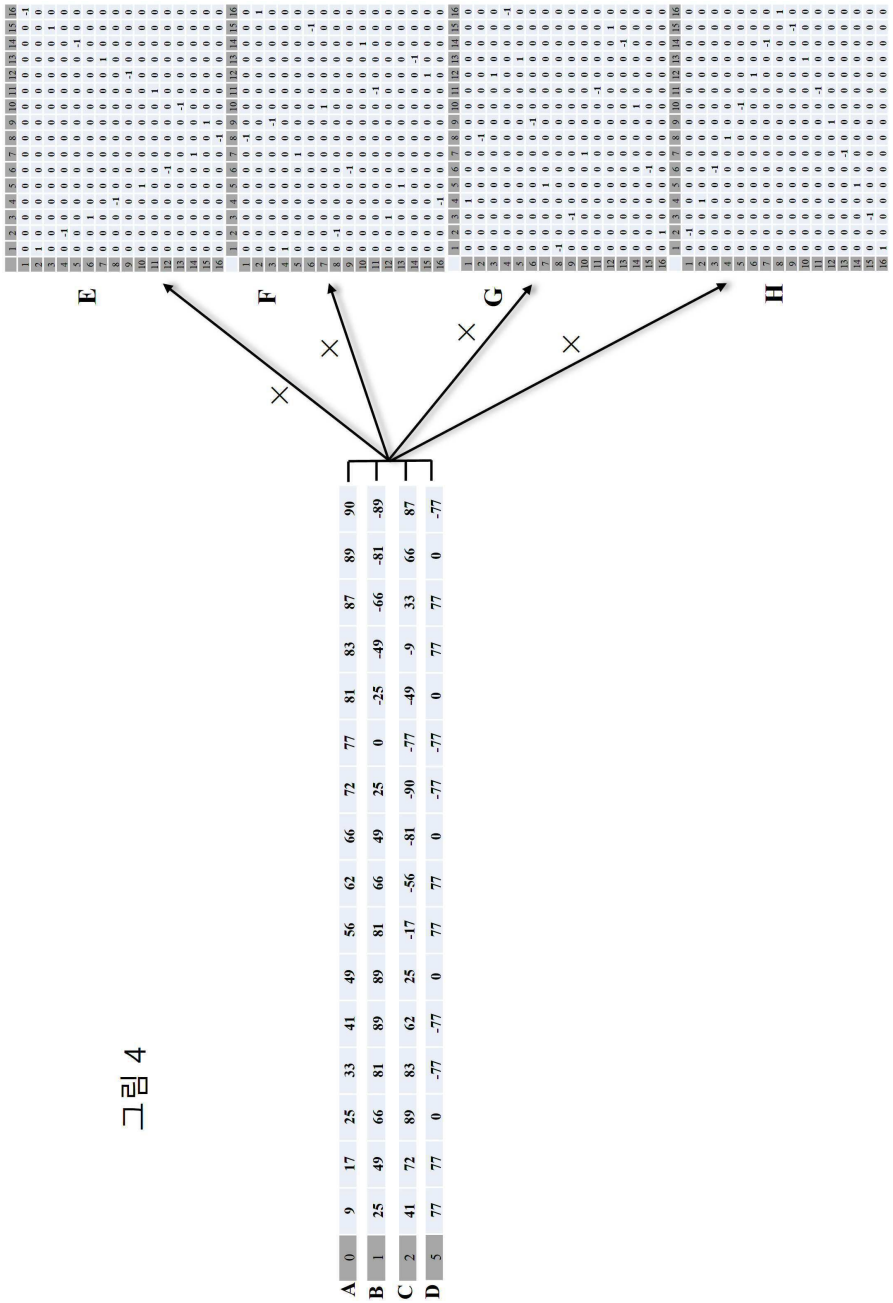


그림 4

[0074]

[0075] 그림 4. 4개의 행 요소만으로 N=16 DST-7을 구성하는 방법

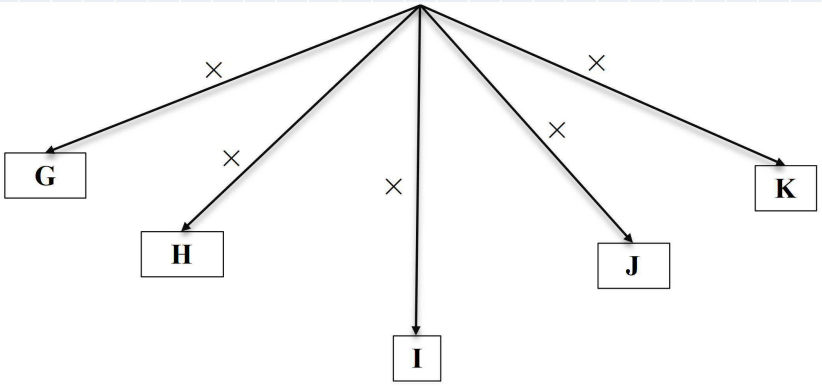
[0077] 그림 4는 N=16의 경우, 0, 1, 2, 5번 째 요소와 순열행렬 **E, F, G, H** 행렬 곱하여 완전한 N=16 DST-7 커널 행렬을 구성하는 방법을 보여준다. N=4, N=8의 경우와 마찬가지로 1x16의 행의 요소와 순열행렬의 곱에 대한 결과 행렬을 이용하여 N=16, DST-7 커널 행렬을 생성한다.

0	9	17	25	33	41	49	56	62	66	72	77	81	83	87	89	90
15	17	-33	49	-62	72	-81	87	-90	89	-83	77	-66	56	-41	25	-9
14	33	-62	81	-90	83	-66	41	-9	-25	56	-77	89	-87	72	-49	17
12	62	-90	66	-9	-56	89	-72	17	49	-87	77	-25	-41	83	-81	33
8	90	-9	-89	17	87	-25	-83	33	81	-41	-77	49	72	-56	-66	52
1	25	49	66	81	89	89	81	66	49	25	0	-25	-49	-66	-81	-89
13	49	-81	89	-66	25	25	-66	89	-81	49	0	-49	81	-89	66	-25
10	81	-66	-25	89	-49	-49	89	-25	-66	81	0	-81	66	25	-89	49
4	66	89	49	-25	-81	-81	-25	49	89	66	0	-66	-89	-49	25	81
7	89	25	-81	-49	66	66	-49	-81	25	89	0	-89	-25	81	49	-66
2	41	72	89	83	62	25	-17	-56	-81	-90	-77	-49	-9	33	66	87
11	72	-83	25	56	-90	49	33	-87	66	9	-77	81	-17	-62	89	-41
6	83	56	-49	-87	-9	81	62	-41	-89	-17	77	66	-33	-90	-25	72
3	56	87	81	41	-17	-66	-90	-72	-25	33	77	89	62	9	-49	-83
9	87	-41	-66	72	33	-89	9	83	-49	-62	77	25	-90	17	81	-56
5	77	77	0	-77	-77	0	77	77	0	-77	-77	0	77	77	0	-77

[0079]

[0080] 그림 5. 4개의 행 요소와 순열 행렬 4개를 이용한 완전한 N=16 DST-7 커널 행렬

0	4	9	13	17	21	26	30	34	38	42	45	50	53	56	60	63	66	68	72	74	77	78	80	82	84	85	86	88	88	89	90	90
1	13	26	38	50	60	68	77	82	86	89	90	88	85	80	74	66	56	45	34	21	9	-4	-17	-30	-42	-53	-63	-72	-78	-84	-88	-90
2	21	42	60	74	84	89	89	84	74	60	42	21	0	-21	-42	-60	-74	-84	-89	-89	-84	-74	-60	-42	-21	0	21	42	60	74	84	89
3	30	56	77	88	89	80	63	38	9	-21	-50	-72	-85	-90	-84	-68	-45	-17	13	42	66	82	90	86	74	53	26	-4	-34	-60	-78	-88
5	45	78	90	77	42	-4	-50	-80	-90	-74	-38	9	53	82	89	72	34	-13	-56	-84	-88	-68	-30	17	60	85	88	66	26	-21	-63	-86
6	53	85	85	53	0	-53	-85	-85	-53	0	53	85	85	53	0	-53	-85	-85	-53	0	53	85	85	53	0	-53	-85	-85	-53	0	53	85



[0082]

[0083] 그림 6. N=32의 경우 DST-7의 0, 1, 2, 3, 5, 6의 6개의 행 요소와 5개의 순열행렬 G, H, I, J, K 행렬을 이용하여 완전한 DST-7을 구성하는 예시

[0085] N=32의 경우 DST-7의 0, 1, 2, 3, 5, 6 행 요소와 순열행렬 G, H, I, J, K 행렬을 이용하여 완전한 DST-7을 구성한다. 그림 6은 N=32에 대한 실시예를 보여준다. 여기서 순열행렬 G, H, I, J, K는 아래와 같이 정의한다.

$G =$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
3	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0
5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
7	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
9	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
13	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[0087]

[0088] 그림 7. N=32인 경우, 순열행렬 G

$H =$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[0090]

[0091] 그림 8. N=32인 경우, 순열행렬 H

$\mathbf{I} =$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[0093]

[0094] 그림 9. N=32인 경우, 순열행렬 $\mathbf{I}$

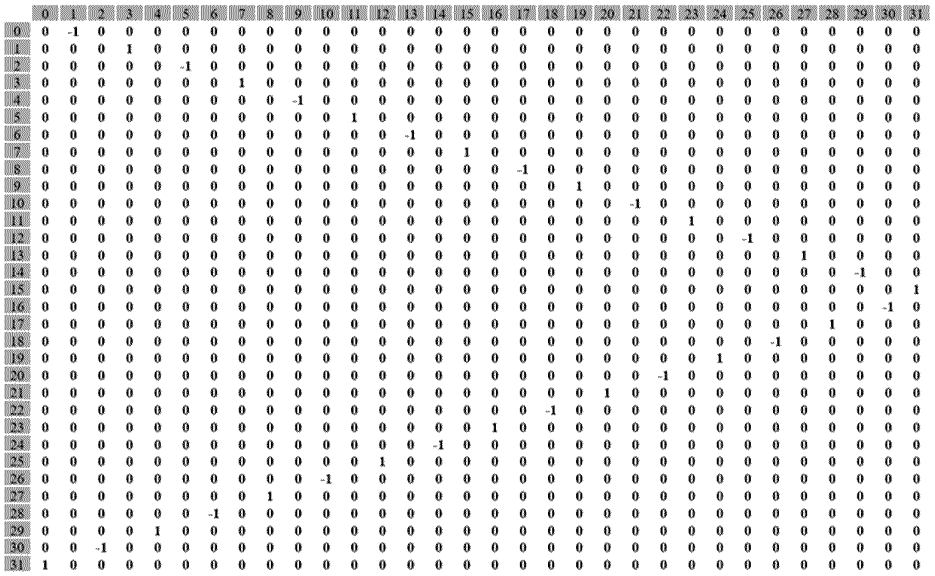
$\mathbf{J} =$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[0096]

[0097] 그림 10. N=32인 경우, 순열행렬 J

[0098] K =



[0099] 그림 11. N=32의 경우, 순열행렬 K

[0100] 위의 실시예 N=4, 8, 16, 32의 경우와 마찬가지로 N=64와 그 이상의 크기에 대해서도 동일한 방법을 이용하여 DST-7을 구성할 수 있다.

[0102] 또한 본 발명에서는 G, H, I, J, K로 표현되는 순열 행렬의 개수가 많아 행렬의 관리와 메모리 상의 문제점을 해결하기 위하여 순열 행렬의 개수를 줄이는 방법을 제안한다.

표 1

[0103] DST-7의 6개의 행으로 순열 행렬 G, H, I, J, K를 이용하여 32x32행렬을 생성하는 방법

DST-7의 행번호	DST-7의 행과 순열행렬 곱셈 후 생성되는 DST-7의 행번호				
	G	H	I	J	K
0	DST-7의 31번째 행 생성	DST-7의 30번째 행 생성	DST-7의 28번째 행 생성	DST-7의 24번째 행 생성	DST-7의 16번째 행 생성
1	DST-7의29번째 행 생성	DST-7의 26번째 행 생성	DST-7의 20번째 행 생성	DST-7의 8번째 행 생성	DST-7의 15번째 행 생성
2	DST-7의 27번째 행 생성	DST-7의 22번째 행 생성	DST-7의 12번째 행 생성	DST-7의 7번째 행 생성	DST-7의 17번째 행 생성
3	DST-7의 25번째 행 생성	DST-7의 18번째 행 생성	DST-7의 4번째 행 생성	DST-7의 23번째 행 생성	DST-7의 14번째 행 생성
5	DST-7의 21번째 행 생성	DST-7의 10번째 행 생성	DST-7의 11번째 행 생성	DST-7의 9번째 행 생성	DST-7의 13번째 행 생성
6	DST-7의 19번째 행 생성				

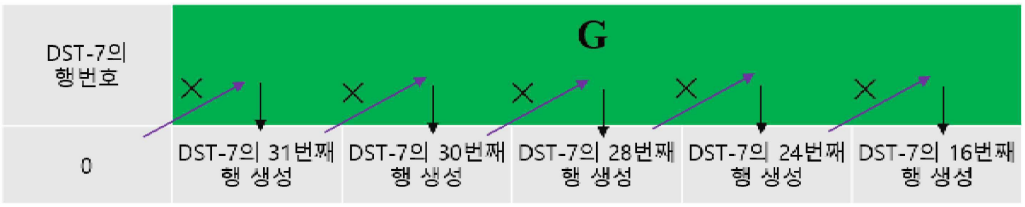
[0104] 표 1에서 보는 것처럼 N=32 DST-7의 0, 1, 2, 3, 5 6번째 행과 G,….,K 순열 행렬을 이용하여 N=32 DST-7의 각 행에 위치하는 원소들을 생성할 수 있다. 앞서 설명한 바와 같이 이 경우 G,…,K 순열 행렬을 모두 저장하고 관리해야 하는 단점이 존재하므로 본 발명에서는 G행렬 하나 만으로 32x32 DST-7 전체 행렬을 생성하는 방법을 제안한다.

표 2

[0105] DST-7의 6개의 행으로 순열 행렬 G를 이용하여 N=32, DST-7 변환 커널 행렬을 생성하는 방법

DST-7의 행번호	DST-7의 행과 순열행렬 곱셈 후 생성되는 DST-7의 행번호				
	G				
0	DST-7의 31번째 행 생성	DST-7의 30번째 행 생성	DST-7의 28번째 행 생성	DST-7의 24번째 행 생성	DST-7의 16번째 행 생성
1	DST-7의29번째 행 생성	DST-7의 26번째 행 생성	DST-7의 20번째 행 생성	DST-7의 8번째 행 생성	DST-7의 15번째 행 생성
2	DST-7의 27번째 행 생성	DST-7의 22번째 행 생성	DST-7의 12번째 행 생성	DST-7의 7번째 행 생성	DST-7의 17번째 행 생성
3	DST-7의 25번째 행 생성	DST-7의 18번째 행 생성	DST-7의 4번째 행 생성	DST-7의 23번째 행 생성	DST-7의 14번째 행 생성
5	DST-7의 21번째 행 생성	DST-7의 10번째 행 생성	DST-7의 11번째 행 생성	DST-7의 9번째 행 생성	DST-7의 13번째 행 생성
6	DST-7의 19번째 행 생성				

[0106] 표 2는 DST-7의 0, 1, 2, 3, 5, 6번째의 6개의 행과 G행렬 하나 만으로 N=32 전체 DST-7 행렬을 유도하는 방법을 나타낸다. 먼저 DST-7의 첫번째 행과 G행렬을 이용하는 방법을 사용하여 DST-7의 32, 30, 28, 24, 16번째 행을 유도하는 방법은 그림 12와 같다.



[0107] 그림 12. DST-7의 0번째 행과 순열행렬 G를 이용한 DST-7의 행렬 성분 유도

[0109] 1. 먼저 DST-7의 0번째 행과 순열 행렬 G를 곱하여 DST-7의 31번째 행 생성

[0110] 2. 생성된 DST-7의 31번째 행과 순열 행렬 G를 곱하여 DST-7의 30번째 행 생성

[0111] 3. 위 그림과 같은 방법으로 DST-7의 16번째 행까지 생성

[0112] 위 과정을 이용하여 DST-7의 0번째 행과 순열행렬 G를 이용하여 그림 13과 같이 DST-7행렬의 0, 31, 30, 28, 24, 16번째 행을 생성한다. 이 때 각 행의 첫번째 원소가 0보다 작으면 모든 행의 원소에 -1을 곱한다. 이 과정은 첫 번째 행의 마지막 원소의 절대값이 다른 행의 첫 번째 원소의 절대값과 같아 질 때까지 수행한다. 그림 13처럼 첫번째 행인 90와 마지막 행의 첫번째 원소인 90이 같은 값을 가지는 것을 알 수 있다.

0	4	9	13	17	21	26	30	34	38	42	45	50	53	56	60	63	66	68	72	74	77	78	80	82	84	85	86	88	88	89	90	90
31	9	-17	26	-34	42	-50	56	-63	68	-74	78	-82	85	-88	89	-90	90	-88	86	-84	80	-77	72	-66	60	-53	45	-38	30	-21	13	-4
30	17	-34	50	-63	74	-82	88	-90	88	-84	77	-66	53	-38	21	-4	-13	30	-45	60	-72	80	-86	90	-89	85	-78	68	-56	42	-26	9
28	34	-63	82	-90	84	-66	38	-4	-30	60	-80	90	-85	68	-42	9	26	-56	78	-89	86	-72	45	-13	-21	53	-77	88	-88	74	-50	17
24	63	-90	66	-4	-60	90	-68	9	56	-89	72	-13	-53	88	-74	17	50	-88	77	-21	-45	86	-78	26	42	-85	80	-30	-38	84	-82	34
16	90	-4	-90	9	89	-13	-88	17	88	-21	-86	26	85	-30	-84	34	82	-38	-80	42	78	-45	-77	50	74	-53	-72	56	68	-60	-66	63

[0115]

[0116] 그림 13. DST-7의 0번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행 원소들

[0118] 마찬가지로 표 2를 이용하면 DST-7의 0, 1, 3, 5, 6번째 행 중의 하나와 G행렬을 이용하면 전체 DST-7의 행렬을 유도할 수 있다. 이 때 생성되는 DST-7의 행의 원소들은 그림 14, 15, 16, 17, 18과 같다.

1	13	26	38	50	60	68	77	82	86	89	90	88	85	80	74	66	56	45	34	21	9	-4	-17	-30	-42	-53	-63	-72	-78	-84	-88	-90
29	26	-50	68	-82	89	-88	80	-66	45	-21	-4	30	-53	72	-84	90	-88	78	-63	42	-17	-9	34	-56	74	-85	90	-86	77	-60	38	-13
26	50	-82	88	-66	21	30	-72	90	-78	42	9	-56	85	-86	60	-13	-38	77	-90	74	-34	-17	63	-88	84	-53	4	45	-80	89	-68	26
20	82	-66	-30	90	-42	-56	86	-13	-77	74	17	-88	53	45	-89	26	68	-80	-4	84	-63	-34	90	-38	-60	85	-9	-78	72	21	-88	50
8	66	90	56	-13	-74	-88	-45	26	80	84	34	-38	-85	-78	-21	50	88	72	9	-60	-90	-63	4	68	89	53	-17	-77	-86	-42	30	82
15	90	13	-88	-26	84	38	-78	-50	72	60	-63	-68	53	77	-42	-82	30	86	-17	-89	4	90	9	-88	-21	85	34	-80	-45	74	56	-66

[0120]

[0121] 그림 14. DST-7의 1번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

2	21	42	60	74	84	89	89	84	74	60	42	21	0	-21	-42	-60	-74	-84	-89	-89	-84	-74	-60	-42	-21	0	21	42	60	74	84	89
27	42	-74	89	-84	60	-21	-21	60	-84	89	-74	42	0	-42	74	-89	84	-60	21	21	-60	84	-89	74	-42	0	42	-74	89	-84	60	-21
22	74	-84	21	60	-89	42	42	-89	60	21	-84	74	0	-74	84	-21	-60	89	-42	-42	89	-60	-21	84	-74	0	74	-84	21	60	-89	42
12	84	60	-42	-89	-21	74	74	-21	-89	-42	60	84	0	-84	-60	42	89	21	-74	-74	21	89	42	-60	-84	0	84	60	-42	-89	-21	74
7	60	89	74	21	-42	-84	-84	-42	21	74	89	60	0	-60	-89	-74	-21	42	84	84	42	-21	-74	-89	-60	0	60	89	74	21	-42	-84
17	89	-21	-84	42	74	-60	-60	74	42	-84	-21	89	0	-89	21	84	-42	-74	60	60	-74	-42	84	21	-89	0	89	-21	-84	42	74	-60

[0123]

[0124] 그림 15. DST-7의 2번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

3	30	56	77	88	89	80	63	38	9	-21	-50	-72	-85	-90	-84	-68	-45	-17	13	42	66	82	90	86	74	53	26	-4	-34	-60	-78	-88
25	56	-88	80	-38	-21	72	-90	68	-17	-42	82	-86	53	4	-60	89	-78	34	26	-74	90	-66	13	45	-84	85	-50	-9	63	-89	77	-30
18	88	-38	-72	68	42	-86	-4	89	-34	-74	66	45	-85	-9	89	-30	-77	63	50	-84	-13	90	-26	-78	60	53	-82	-17	90	-21	-80	56
4	38	68	86	89	74	45	9	-30	-63	-84	-90	-78	-53	-17	21	56	80	90	82	60	26	-13	-50	-77	-89	-85	-66	-34	4	42	72	88
23	68	-89	45	30	-84	78	-17	-56	90	-60	-13	77	-85	34	42	-88	72	-4	-66	89	-50	-26	82	-80	21	53	-90	63	9	-74	86	-38
14	88	30	-78	-56	60	77	-34	-88	4	89	26	-80	-53	63	74	-38	-86	9	90	21	-82	-50	66	72	-42	-85	13	90	17	-84	-45	68

[0126]

[0127] 그림 16. DST-7의 3번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

5	45	78	90	77	42	-4	-50	-80	-90	-74	-38	9	53	82	89	72	34	-13	-56	-84	-88	-68	-30	17	60	85	88	66	26	-21	-63	-86
21	78	-77	-4	80	-74	-9	82	-72	-13	84	-68	-17	85	-66	-21	86	-63	-26	88	-60	-30	88	-56	-34	89	-53	-38	90	-50	-42	90	-45
10	77	80	9	-72	-84	-17	66	86	26	-60	-88	-34	53	90	42	-45	-90	-50	38	89	56	-30	-88	-63	21	85	68	-13	-82	-74	4	78
11	80	72	-17	-86	-60	34	90	45	-50	-89	-30	63	85	13	-74	-78	4	82	68	-21	-88	-56	38	90	42	-53	-88	-26	66	84	9	-77
9	72	86	34	-45	-89	-63	13	78	82	21	-56	-90	-53	26	84	77	9	-66	-88	-42	38	88	68	-4	-74	-85	-30	50	90	60	-17	-80
13	86	45	-63	-78	21	90	26	-77	-66	42	88	4	-85	-50	60	80	-17	-90	-30	74	68	-38	-88	-9	84	53	-56	-82	13	89	34	-72

[0129]

[0130] 그림 17. DST-7의 5번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

6	53	85	85	53	0	-53	-85	-85	-53	0	53	85	85	53	0	-53	-85	-85	-53	0	53	85	85	53	0	-53	-85	-85	-53	0	53	85
19	85	-53	-53	85	0	-85	53	53	-85	0	85	-53	-53	85	0	-85	53	53	-85	0	85	-53	-53	85	0	-85	53	53	-85	0	85	-53

[0132]

[0133] 그림 18. DST-7의 6번째 행과 순열행렬 G를 이용하여 생성한 DST-7의 일부 행원소들

[0135] 그림 12 - 그림 18까지 설명한 방법은 1개의 순열행렬 G와 총 6개의 행 원소 만으로 N=32 DST-7을 구성할 수 있음을 보여준다. 이 때 그림 13 - 그림 18의 가장 상위에 있는 행만을 저장하면 그림 12의 유도 방식에 의해 총 N=32, DST-7을 유도 생성할 수 있다.

[0136] 이 방법에서 총 6개의 행들을 그룹을 지어 가장 상위에 있는 행들과 G 행렬을 이용하여 N=32 DST-7을 구성할 수 있는데, 이 때 그림 12에서 행 그룹을 구성할 때는 첫 번째 행의 마지막 원소가 다른 행의 첫 번째 원소가 절대값이 동일한 값을 가질 때 행 그룹 구성 과정이 정지한다. 예를 들어 그림 13에서처럼, 0번째 행의 마지막 값인 90이, 16번째 행의 첫 번째 값인 90과 동일하므로 행 그룹 구성이 16번 행에서 멈추게 되어 16번 행이 첫 번째 행 그룹의 마지막 행이 된다. 따라서 이 그룹의 첫 번째 행 만을 선택하여 저장하여 다른 다섯 개의 행을 유도할 수 있다.

[0138] 그러나 행 그룹 구성 시 첫 번째 행의 마지막 값이 다른 행의 첫 번째 값과 동일 할 때 그 행 그룹 구성의 정지

기준이 되므로, 행 그룹 만들기의 멈춤이 완전한 한 DST-7 커널을 만들지 못하도록 멈추는 경우가 존재한다. 예를 들어 그림 16에서, 3번째 행의 마지막 값이 -88이므로 이 절대값과 동일한 첫 번째 값을 가지는 행은 18번째 행이 된다. 이 경우 18번째 행에서 행 그룹 구성이 멈추게 되고 또 다른 행 그룹 구성이 시작된다. 이렇게 되면 결과적으로 유도 생성된 N=32, DST-7이 원 DST-7과 다른 값을 가지게 되며, 부호화 효율을 감소시킬 가능성이 존재한다.

[0139] 이 문제를 해결하기 위하여 본 발명에서는 그림 16의 세 번째 행 (행 번호 18)의 첫 번째 값을 88에서 89로 변경한다. 이렇게 하면 첫 번째 행 (행번 호 3)의 마지막 값인 -88이 세번째 행 (행번호 18)의 첫 번째 값인 88의 절대값이 달라지게 되므로 행 그룹 만들기가 세 번째 행에서 멈추게 되는 것을 방지 한다. 따라서 총 6개의 행이 하나의 행 그룹이 만들어 질 때까지 행 그룹 생성을 계속할 수 있다.

[0140]

-88	-30	-56	89	-38	68																										
-78	77	-80	72	86	-45																										
-60	-89	-21	42	-74	-84																										
-34	63	90	4	9	17																										
-4	-9	-17	-34	63	90																										
26	-50	-82	-66	-90	13																										
53	85	53	-85	53	-85																										
74	-84	60	-89	21	-42																										
86	45	-78	-77	-80	72																										
90	13	-26	-50	82	66																										
82	-66	90	-13	-26	-50																										
66	90	-13	26	-50	-82																										
42	-74	-84	60	89	21																										
13	26	50	82	-66	90																										
-17	34	63	90	-4	9																										
-45	-78	-77	80	72	-86																										
-68	88	-30	56	-89	-38																										
-84	-60	89	21	42	74																										
-90	4	-9	-17	34	63																										
-85	53	-85	-53	-85	-53																										
-72	-86	45	-78	77	-80																										
-50	82	66	-90	-13	26																										
-21	-42	-74	-84	-60	89																										
9	-17	-34	-63	90	4																										
38	68	88	-30	-56	-89																										
63	-90	-4	9	-17	-34																										
80	72	-86	45	78	77																										
89	-21	42	74	-84	60																										
89	-38	68	88	30	-56																										
77	80	-72	86	45	-78																										
56	-89	-38	68	-88	30																										
30	-56	89	38	68	88																										
3	25	18	4	23	14																										

[0141] 그림 19. 행 그룹 구성을 효과적으로 하기 위해 세 번째 행 (행번호 18)의 값을 88에서 89로 변경하고, 이에 따라서 DST-7 커널의 성질에 따라 그와 연관된 값이 변경한 예.

[0142] DST-7의 커널 성질에 따라 그림 19에서 첫 번째 행 (행번호 3)의 네 번째 열의 값이 88에서 89로 변경되게 된다. 본 발명에서는 각 행 그룹의 첫 번째 행 만을 저장하므로 그림 19에서 변경된 값의 행이 저장되게 된다.

[0144] 이러한 과정을 이용하여 얻어진 N=32, DST-7 행렬은 아래와 같다.

[0146] 본 명세서에서 (X)Y는 엘리먼트 X가 엘리먼트 Y로 대체됨을 의미한다.

[0147] 32-point DST7

[0148] DST7_32[m][n] = {

[0149] {4 9 13 17 21 26 30 34 38 42 45 50 53 56 60 63 66 68 72 74 77 78 80 82 84 85 86 88 88 89 90 90}

[0150] {13 26 38 50 60 68 77 82 86 89 90 88 85 80 74 66 56 45 34 21 9 -4 -17 -30 -42 -53 -63 -72 -78 -84 -88 -90}

[0151] {21 42 60 74 84 89 89 84 74 60 42 21 0 -21 -42 -60 -74 -84 -89 -89 -84 -74 -60 -42 -21 0 21 42 60 74 84 89}

[0152] {30 56 77 89 (88)89 80 63 38 9 -21 -50 -72 -85 -90 -84 -68 -45 -17 13 42 66 82 90 86 74 53 26 -4 -34 -60 -78 -88}

[0153] {38 68 86 88 74 45 9 -30 -63 -84 -90 -78 -53 -17 21 56 80 90 82 60 26 -13 -50 -77 -89 -85 -66 -34 4 42 72 (88)89}

[0154] {45 78 90 77 42 -4 -50 -80 -90 -74 -38 9 53 82 89 72 34 -13 -56 -84 -88 -68 -30 17 60 85 88 66 26 -21 -63 -86}

[0155] {53 85 85 53 0 -53 -85 -85 -53 0 53 85 85 53 0 -53 -85 -85 -53 0 53 85 85 53 0 -53 -85 -85 -53 0 53 85}

[0156] {60 89 74 21 -42 -84 -84 -42 21 74 89 60 0 -60 -89 -74 -21 42 84 84 42 -21 -74 -89 -60 0 60 89 74 21 -42 -84}

[0157] {66 90 56 -13 -74 -88 -45 26 80 84 34 -38 -85 -78 -21 50 88 72 9 -60 -90 -63 4 68 89 53 -17 -77 -86 -42 30 82}

[0158] {72 86 34 -45 -89 -63 13 78 82 21 -56 -90 -53 26 84 77 9 -66 -88 -42 38 88 68 -4 -74 -85 -30 50 90 60 -17 -80}

[0159] {77 80 9 -72 -84 -17 66 86 26 -60 -88 -34 53 90 42 -45 -90 -50 38 89 56 -30 -88 -63 21 85 68 -13 -82 -74 4 78}

[0160] {80 72 -17 -86 -60 34 90 45 -50 -89 -30 63 85 13 -74 -78 4 82 68 -21 -88 -56 38 90 42 -53 -88 -26 66 84 9 -77}

[0161] {84 60 -42 -89 -21 74 74 -21 -89 -42 60 84 0 -84 -60 42 89 21 -74 -74 21 89 42 -60 -84 0 84 60 -42 -89 -21 74}

[0162] {86 45 -63 -78 21 90 26 -77 -66 42 88 4 -85 -50 60 80 -17 -90 -30 74 68 -38 -88 -9 84 53 -56 -82 13 89 34 -72}

[0163] {88 30 -78 -56 60 77 -34 (-88)-89 4 89 26 -80 -53 63 74 -38 -86 9 90 21 -82 -50 66 72 -42 -85 13 90 17 -84 -45 68}

[0164] {90 13 -88 -26 84 38 -78 -50 72 60 -63 -68 53 77 -42 -82 30 86 -17 -89 4 90 9 -88 -21 85 34 -80 -45 74 56 -66}

[0165] {90 -4 -90 9 89 -13 -88 17 88 -21 -86 26 85 -30 -84 34 82 -38 -80 42 78 -45 -77 50 74 -53 -72 56 68 -60 -66 63}

[0166] {89 -21 -84 42 74 -60 -60 74 42 -84 -21 89 0 -89 21 84 -42 -74 60 60 -74 -42 84 21 -89 0 89 -21 -84 42 74 -60}

[0167] {(88)89 -38 -72 68 42 -86 -4 88 -34 -74 66 45 -85 -9 89 -30 -77 63 50 -84 -13 90 -26 -78 60 53 -82 -17 90 -21 -80 56}

[0168] {85 -53 -53 85 0 -85 53 53 -85 0 85 -53 -53 85 0 -85 53 53 -85 0 85 -53 -53 85 0 -85 53 53 -85 0 85 -53}

[0169] {82 -66 -30 90 -42 -56 86 -13 -77 74 17 -88 53 45 -89 26 68 -80 -4 84 -63 -34 90 -38 -60 85 -9 -78 72 21 -88 50}

[0170] {78 -77 -4 80 -74 -9 82 -72 -13 84 -68 -17 85 -66 -21 86 -63 -26 88 -60 -30 88 -56 -34 89 -53 -38 90 -50 -42 90 -45}

[0171] {74 -84 21 60 -89 42 42 -89 60 21 -84 74 0 -74 84 -21 -60 89 -42 -42 89 -60 -21 84 -74 0 74 -84 21 60 -89 42}

[0172] {68 -88 45 30 -84 78 -17 -56 90 -60 -13 77 -85 34 42 (-88)-89 72 -4 -66 89 -50 -26 82 -80 21 53 -90 63 9 -74 86 -38}

[0173] {63 -90 66 -4 -60 90 -68 9 56 -89 72 -13 -53 88 -74 17 50 -88 77 -21 -45 86 -78 26 42 -85 80 -30 -38 84 -82 34}

[0174] {56 (-88)-89 80 -38 -21 72 -90 68 -17 -42 82 -86 53 4 -60 88 -78 34 26 -74 90 -66 13 45 -84 85 -50 -9 63 -89 77 -30}

[0175] {50 -82 88 -66 21 30 -72 90 -78 42 9 -56 85 -86 60 -13 -38 77 -90 74 -34 -17 63 -88 84 -53 4 45 -80 89 -68 26}

[0176] {42 -74 89 -84 60 -21 -21 60 -84 89 -74 42 0 -42 74 -89 84 -60 21 21 -60 84 -89 74 -42 0 42 -74 89 -84 60 -21}

[0177] {34 -63 82 -90 84 -66 38 -4 -30 60 -80 90 -85 68 -42 9 26 -56 78 -89 86 -72 45 -13 -21 53 -77 88 -88 74 -50 17}

[0178] {26 -50 68 -82 89 -88 80 -66 45 -21 -4 30 -53 72 -84 90 -88 78 -63 42 -17 -9 34 -56 74 -85 90 -86 77 -60 38 -13}

[0179] {17 -34 50 -63 74 -82 88 -90 88 -84 77 -66 53 -38 21 -4 -13 30 -45 60 -72 80 -86 90 -89 85 -78 68 -56 42 -26 9}

[0180] {9 -17 26 -34 42 -50 56 -63 68 -74 78 -82 85 -88 89 -90 90 -88 86 -84 80 -77 72 -66 60 -53 45 -38 30 -21 13 -4}

[0181] }

[0183] 16-point DST7

[0184] DST7_16[m][n] = {

[0185] {9 17 25 33 41 49 56 62 66 72 77 81 83 87 89 90}

[0186] {25 49 66 81 89 89 81 66 49 25 0 -25 -49 -66 -81 -89}

[0187] {41 72 89 83 62 25 -17 -56 -81 -90 -77 -49 -9 33 66 87}

[0188] {56 87 81 41 -17 -66 -90 -72 -25 33 77 89 62 9 -49 -83}

[0189] {66 89 49 -25 -81 -81 -25 49 89 66 0 -66 -89 -49 25 81}

[0190] {77 77 0 -77 -77 0 77 77 0 -77 -77 0 77 77 0 -77}

[0191] {83 56 -49 -87 -9 81 62 -41 -89 -17 77 66 -33 -90 -25 72}

[0192] {89 25 -81 -49 66 66 -49 -81 25 89 0 -89 -25 81 49 -66}

[0193] {90 -9 -89 17 87 -25 -83 33 81 -41 -77 49 72 -56 -66 62}

[0194] {87 -41 -66 72 33 -89 9 83 -49 -62 77 25 -90 17 81 -56}

[0195] {81 -66 -25 89 -49 -49 89 -25 -66 81 0 -81 66 25 -89 49}

[0196] {72 -83 25 56 -90 49 33 -87 66 9 -77 81 -17 -62 89 -41}

[0197] {62 -90 66 -9 -56 89 -72 17 49 -87 77 -25 -41 83 -81 33}

[0198] {49 -81 89 -66 25 25 -66 89 -81 49 0 -49 81 -89 66 -25}

[0199] {33 -62 81 -90 83 -66 41 -9 -25 56 -77 89 -87 72 -49 17}

[0200] {17 -33 49 -62 72 -81 87 -90 89 -83 77 -66 56 -41 25 -9}

[0201] }

[0203] 8-point DST7

[0204] DST7_8[m][n] = {

[0205] {17 32 46 60 71 78 85 86}

[0206] {46 78 86 71 32 -17 -60 -85}

[0207] {71 85 32 -46 -86 -60 17 78}

[0208] {85 46 -60 -78 17 86 32 -71}

[0209] {86 -17 -85 32 78 -46 -71 60}

[0210] {78 -71 -17 85 -60 -32 86 -46}

[0211] {60 -86 71 -17 -46 85 -78 32}

[0212] {32 -60 78 -86 85 -71 46 -17}

[0213] }

[0215] 4-point DST7

[0216] DST7_4[m][n] = {

[0217] {29 55 74 84}

[0218] {74 74 0 -74}

[0219] {84 -29 -74 55}

[0220] {55 -84 74 -29}

[0222] 각 크기의 DST-7 행렬을 위의 방식을 이용하여 얻었다면 DCT-8은 아래와 같은 식으로 쉽게 유도하여 생성할 수 있다.

$$DCT-8 = S \times DST-7 \times F$$

[0223]

[0224] 여기서 F는 flipping 행렬, S는 부호 변환 (sign change)행렬이고 아래와 같이 정의한다.

[0225] $$F_{m,n} = \begin{cases} 1, & \text{if } n = N-1-m, \\ 0, & \text{otherwise,} \end{cases}$$

[0226] $$S_{m,n} = \begin{cases} (-1)^m, & \text{if } n = m, \\ 0, & \text{otherwise.} \end{cases}$$

[0227] 따라서 이러한 방식으로 유도한 N=4, N=8, N=16, N=32, DCT-8은 각각 아래와 같다.

[0228] 32-point DCT8

[0229] DCT8_32[m][n] = {

[0230] {90 90 89 88 88 86 85 84 82 80 78 77 74 72 68 66 63 60 56 53 50 45 42 38 34 30 26 21 17 13 9 4}

[0231] {90 88 84 78 72 63 53 42 30 17 4 -9 -21 -34 -45 -56 -66 -74 -80 -85 -88 -90 -89 -86 -82 -77 -68 -60 -50 -38 -26 -13}

[0232] {89 84 74 60 42 21 0 -21 -42 -60 -74 -84 -89 -89 -84 -74 -60 -42 -21 0 21 42 60 74 84 89 89 84 74 60 42 21}

[0233] {88 78 60 34 4 -26 -53 -74 -86 -90 -82 -66 -42 -13 17 45 68 84 90 85 72 50 21 -9 -38 -63 -80 -89 (-88)-89 -77 -56 -30}

[0234] {(88)89 72 42 4 -34 -66 -85 -89 -77 -50 -13 26 60 82 90 80 56 21 -17 -53 -78 -90 -84 -63 -30 9 45 74 88 86 68 38}

[0235] {86 63 21 -26 -66 -88 -85 -60 -17 30 68 88 84 56 13 -34 -72 -89 -82 -53 -9 38 74 90 80 50 4 -42 -77 -90 -78 -45}

[0236] {85 53 0 -53 -85 -85 -53 0 53 85 85 53 0 -53 -85 -85 -53 0 53 85 85 53 0 -53 -85 -85 -53 0 53 85 85 53}

[0237] {84 42 -21 -74 -89 -60 0 60 89 74 21 -42 -84 -84 -42 21 74 89 60 0 -60 -89 -74 -21 42 84 84 42 -21 -74 -89 -60}

[0238] {82 30 -42 -86 -77 -17 53 89 68 4 -63 -90 -60 9 72 88 50 -21 -78 -85 -38 34 84 80 26 -45 -88 -74 -13 56 90 66}

[0239] {80 17 -60 -90 -50 30 85 74 4 -68 -88 -38 42 88 66 -9 -77 -84 -26 53 90 56 -21 -82 -78 -13 63 89 45 -34 -86 -72}

[0240] {78 4 -74 -82 -13 68 85 21 -63 -88 -30 56 89 38 -50 -90 -45 42 90 53 -34 -88 -60 26 86 66 -17 -84 -72 9 80 77}

[0241] {77 -9 -84 -66 26 88 53 -42 -90 -38 56 88 21 -68 -82 -4 78 74 -13 -85 -63 30 89 50 -45 -90 -34 60 86 17 -72 -80}

[0242] {74 -21 -89 -42 60 84 0 -84 -60 42 89 21 -74 -74 21 89 42 -60 -84 0 84 60 -42 -89 -21 74 74 -21 -89 -42 60 84}

[0243] {72 -34 -89 -13 82 56 -53 -84 9 88 38 -68 -74 30 90 17 -80 -60 50 85 -4 -88 -42 66 77 -26 -90 -21 78 63 -45 -86}

[0244] {68 -45 -84 17 90 13 -85 -42 72 66 -50 -82 21 90 9 -86 -38 74 63 -53 -80 26 89 4 (-88)-89 -34 77 60 -56 -78 30 88}

[0245] {66 -56 -74 45 80 -34 -85 21 88 -9 -90 -4 89 17 -86 -30 82 42 -77 -53 68 63 -60 -72 50 78 -38 -84 26 88 -13 -90}

[0246] {63 -66 -60 68 56 -72 -53 74 50 -77 -45 78 42 -80 -38 82 34 -84 -30 85 26 -86 -21 88 17 -88 -13 89 9 -90 -4 90}

[0247] {60 -74 -42 84 21 -89 0 89 -21 -84 42 74 -60 -60 74 42 -84 -21 89 0 -89 21 84 -42 -74 60 60 -74 -42 84 21 -89}

[0248] {56 -80 -21 90 -17 -82 53 60 -78 -26 90 -13 -84 50 63 -77 -30 89 -9 -85 45 66 -74 -34 88 -4 -86 42 68 -72 -38 (88)89}

[0249] {53 -85 0 85 -53 -53 85 0 -85 53 53 -85 0 85 -53 -53 85 0 -85 53 53 -85 0 85 -53 -53 85 0 -85 53 53 -85}

[0250] {50 -88 21 72 -78 -9 85 -60 -38 90 -34 -63 84 -4 -80 68 26 -89 45 53 -88 17 74 -77 -13 86 -56 -42 90 -30 -66 82}

[0251] {45 -90 42 50 -90 38 53 -89 34 56 -88 30 60 -88 26 63 -86 21 66 -85 17 68 -84 13 72 -82 9 74 -80 4 77 -78}

[0252] {42 -89 60 21 -84 74 0 -74 84 -21 -60 89 -42 -42 89 -60 -21 84 -74 0 74 -84 21 60 -89 42 42 -89 60 21 -84 74}

[0253] {38 -86 74 -9 -63 90 -53 -21 80 -82 26 50 -89 66 4 -72(88)89 -42 -34 85 -77 13 60 -90 56 17 -78 84 -30 -45 88 -68}

[0254] {34 -82 84 -38 -30 80 -85 42 26 -78 86 -45 -21 77 -88 50 17 -74 88 -53 -13 72 -89 56 9 -68 90 -60 -4 66 -90 63}

[0255] {30 -77 89 -63 9 50 -85 84 -45 -13 66 -90 74 -26 -34 78 -88 60 -4 -53 86 -82 42 17 -68 90 -72 21 38 -80 (88)89 -56}

[0256] {26 -68 89 -80 45 4 -53 84 -88 63 -17 -34 74 -90 77 -38 -13 60 -86 85 -56 9 42 -78 90 -72 30 21 -66 88 -82 50}

[0257] {21 -60 84 -89 74 -42 0 42 -74 89 -84 60 -21 -21 60 -84 89 -74 42 0 -42 74 -89 84 -60 21 21 -60 84 -89 74 -42}

[0258] {17 -50 74 -88 88 -77 53 -21 -13 45 -72 86 -89 78 -56 26 9 -42 68 -85 90 -80 60 -30 -4 38 -66 84 -90 82 -63 34}

[0259] {13 -38 60 -77 86 -90 85 -74 56 -34 9 17 -42 63 -78 88 -90 84 -72 53 -30 4 21 -45 66 -80 88 -89 82 -68 50 -26}

[0260] {9 -26 42 -56 68 -78 85 -89 90 -86 80 -72 60 -45 30 -13 -4 21 -38 53 -66 77 -84 88 -90 88 -82 74 -63 50 -34 17}

[0261] {4 -13 21 -30 38 -45 53 -60 66 -72 77 -80 84 -86 88 -90 90 -89 88 -85 82 -78 74 -68 63 -56 50 -42 34 -26 17 -9}

[0262] }

[0265] 16-point DCT8

[0266] DCT8_16[m][n] = {

[0267] {90 89 87 83 81 77 72 66 62 56 49 41 33 25 17 9}

[0268] {89 81 66 49 25 0 -25 -49 -66 -81 -89 -89 -81 -66 -49 -25}

[0269] {87 66 33 -9 -49 -77 -90 -81 -56 -17 25 62 83 89 72 41}

[0270] {83 49 -9 -62 -89 -77 -33 25 72 90 66 17 -41 -81 -87 -56}

[0271] {81 25 -49 -89 -66 0 66 89 49 -25 -81 -81 -25 49 89 66}

[0272] {77 0 -77 -77 0 77 77 0 -77 -77 0 77 77 0 -77 -77}

[0273] {72 -25 -90 -33 66 77 -17 -89 -41 62 81 -9 -87 -49 56 83}

[0274] {66 -49 -81 25 89 0 -89 -25 81 49 -66 -66 49 81 -25 -89}

[0275] {62 -66 -56 72 49 -77 -41 81 33 -83 -25 87 17 -89 -9 90}

[0276] {56 -81 -17 90 -25 -77 62 49 -83 -9 89 -33 -72 66 41 -87}

[0277] {49 -89 25 66 -81 0 81 -66 -25 89 -49 -49 89 -25 -66 81}

[0278] {41 -89 62 17 -81 77 -9 -66 87 -33 -49 90 -56 -25 83 -72}

[0279] {33 -81 83 -41 -25 77 -87 49 17 -72 89 -56 -9 66 -90 62}

[0280] {25 -66 89 -81 49 0 -49 81 -89 66 -25 -25 66 -89 81 -49}

[0281] {17 -49 72 -87 89 -77 56 -25 -9 41 -66 83 -90 81 -62 33}

[0282] {9 -25 41 -56 66 -77 83 -89 90 -87 81 -72 62 -49 33 -17}

[0283] }

[0285] 8-point DCT8

[0286] DCT8_8[m][n] = {

[0287] {86 85 78 71 60 46 32 17}

[0288] {85 60 17 -32 -71 -86 -78 -46}

[0289] {78 17 -60 -86 -46 32 85 71}

[0290] {71 -32 -86 -17 78 60 -46 -85}

[0291] {60 -71 -46 78 32 -85 -17 86}

[0292] {46 -86 32 60 -85 17 71 -78}

[0293] {32 -78 85 -46 -17 71 -86 60}

[0294] {17 -46 71 -85 86 -78 60 -32}

[0295] }

[0297] 4-point DCT8

[0298] DCT8_4[m][n] = {

[0299] {84 74 55 29}

[0300] {74 0 -74 -74}

[0301] {55 -74 -29 84}

[0302] {29 -74 84 -55}

[0303] }

[0305] DCT-2는 DST-3 커널과의 관계를 이용하여 얻을 수 있다. DCT-2와 DST-3의 관계는 아래와 같다.

[0306] $$DCT-2 = F \times DST-3 \times S$$

[0307] 여기서 F는 flipping 행렬, S는 부호 변환 (sign change)행렬이고 아래와 같이 정의한다.

[0308] $$F_{m,n} = \begin{cases} 1, & \text{if } n = N-1-m, \\ 0, & \text{otherwise,} \end{cases}$$

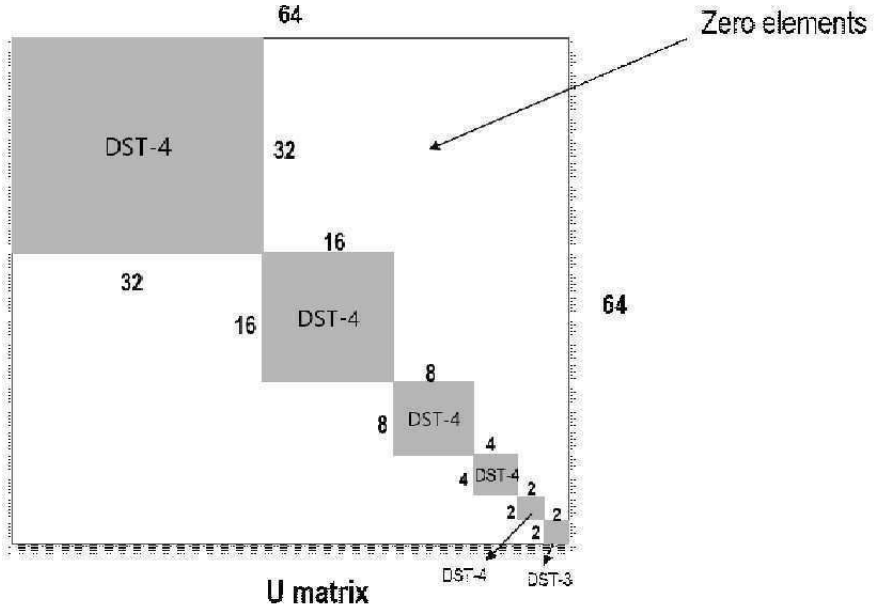
[0309] $$S_{m,n} = \begin{cases} (-1)^m, & \text{if } n = m, \\ 0, & \text{otherwise.} \end{cases}$$

[0310] 위 관계식에서 일단 DST-3 커널을 유도한다면 DCT-2는 F, S 행렬을 이용하여 쉽게 유도 생성할 수 있다. 이를

위해 DST-3가 여러 개의 행렬들로 분해 할 수 있는 성질을 이용한다. N=64, DST-3는 아래와 같은 행렬들로 분해 할 수 있다.

[0311] $$DST-3_{64} = \mathbf{UABCDE}$$

[0312] 여기서 U는 통합 행렬 (unified matrix), A, B, C, D, E는 1, 0, -1로만 구성된 행렬이다. U행렬은 그림 20과 같은 형태를 가지고 있다.



[0314]

[0315] 그림 20. 행렬 U의 분해 형태

[0316] 본 발명에서는 행렬 U를 기반으로 각 크기의 DST-3을 아래와 같이 구할 수 있다.

[0317] $$DST-3_4 = T[U_{64} \times A_{64}]_4$$

[0318] $$DST-3_8 = T[U_{64} \times M_{64}]_8$$

[0319] $$DST-3_{16} = T[U_{64} \times N_{64}]_{16}$$

[0320] $$DST-3_{32} = T[U_{64} \times O_{64}]_{32}$$

[0321] $$DST-3_{64} = U_{64} \times P_{64}$$

[0322] 여기서 $T[\ ]_N$ 은 매트릭스의 오른쪽 아래 부분 NN 크기를 취하는 함수 이다. M, N, O, P 행렬은 아래 식을 이용하여 얻을 수 있다.

[0323] $$M = A \times B$$

[0324] $$N = A \times B \times C$$

[0325] $$O = A \times B \times C \times D$$

[0326] $$P = A \times B \times C \times D \times E$$

[0327] 이러한 방법을 이용하여 얻은 N=2, N=4, N=8, N=16, N=32, N=64 DCT-2 커널은 아래와 같이 정의 한다.

[0328] 2-point DCT2

[0329] DCT2_2[m][n] = {

[0330] {64 64}

[0331] {64 -64}

[0332] }

[0334] 4-point DCT2

[0335] DCT2_4[m][n] = {

[0336] {64 64 64 64}

[0337] {83 36 -36 -83}

[0338] {64 -64 -64 64}

[0339] {36 -83 83 -36}

[0340] }

[0342] 8-point DCT2

[0343] DCT2_8[m][n] = {

[0344] {64 64 64 64 64 64 64 64}

[0345] {89 75 50 18 -18 -50 -75 -89}

[0346] {83 36 -36 -83 -83 -36 36 83}

[0347] {75 -18 -89 -50 50 89 18 -75}

[0348] {64 -64 -64 64 64 -64 -64 64}

[0349] {50 -89 18 75 -75 -18 89 -50}

[0350] {36 -83 83 -36 -36 83 -83 36}

[0351] {18 -50 75 -89 89 -75 50 -18}

[0352] }

[0354] 16-point DCT2

[0355] DCT2_16[m][n] = {

[0356] {64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}

[0357] {90 87 80 70 57 43 25 9 -9 -25 -43 -57 -70 -80 -87 -90}

[0358] {89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89}

[0359] {87 57 9 -43 -80 -90 -70 -25 25 70 90 80 43 -9 -57 -87}

[0360] {83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83}

[0361] {80 9 -70 -87 -25 57 90 43 -43 -90 -57 25 87 70 -9 -80}

[0362] {75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75}

[0363] {70 -43 -87 9 90 25 -80 -57 57 80 -25 -90 -9 87 43 -70}

[0364] {64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64}

[0365] {57 -80 -25 90 -9 -87 43 70 -70 -43 87 9 -90 25 80 -57}

[0366] {50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50}

[0367] {43 -90 57 25 -87 70 9 -80 80 -9 -70 87 -25 -57 90 -43}

[0368] {36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36}

[0369] {25 -70 90 -80 43 9 -57 87 -87 57 -9 -43 80 -90 70 -25}

[0370] {18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18}

[0371] {9 -25 43 -57 70 -80 87 -90 90 -87 80 -70 57 -43 25 -9}

[0372] }

[0374] 32-point DCT2

[0375] DCT2_32[m][n] = {

[0376] {64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}

[0377] {90 90 88 85 82 78 73 67 61 54 46 38 31 22 13 4 -4 -13 -22 -31 -38 -46 -54 -61 -67 -73 -78 -82 -85 -88 -90 -90}

[0378] {90 87 80 70 57 43 25 9 -9 -25 -43 -57 -70 -80 -87 -90 -90 -87 -80 -70 -57 -43 -25 -9 9 25 43 57 70 80 87 90}

[0379] {90 82 67 46 22 -4 -31 -54 -73 -85 -90 -88 -78 -61 -38 -13 13 38 61 78 88 90 85 73 54 31 4 -22 -46 -67 -82 -90}

[0380] {89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89 89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89}

[0381] {88 67 31 -13 -54 -82 -90 -78 -46 -4 38 73 90 85 61 22 -22 -61 -85 -90 -73 -38 4 46 78 90 82 54 13 -31 -67 -88}

[0382] {87 57 9 -43 -80 -90 -70 -25 25 70 90 80 43 -9 -57 -87 -87 -57 -9 43 80 90 70 25 -25 -70 -90 -80 -43 9 57 87}

[0383] {85 46 -13 -67 -90 -73 -22 38 82 88 54 -4 -61 -90 -78 -31 31 78 90 61 4 -54 -88 -82 -38 22 73 90 67 13 -46 -85}

[0384] {83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83}

[0385] {82 22 -54 -90 -61 13 78 85 31 -46 -90 -67 4 73 88 38 -38 -88 -73 -4 67 90 46 -31 -85 -78 -13 61 90 54 -22 -82}

[0386] {80 9 -70 -87 -25 57 90 43 -43 -90 -57 25 87 70 -9 -80 -80 -9 70 87 25 -57 -90 -43 43 90 57 -25 -87 -70 9 80}

[0387] {78 -4 -82 -73 13 85 67 -22 -88 -61 31 90 54 -38 -90 -46 46 90 38 -54 -90 -31 61 88 22 -67 -85 -13 73 82 4 -78}

[0388] {75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75 75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75}

[0389] {73 -31 -90 -22 78 67 -38 -90 -13 82 61 -46 -88 -4 85 54 -54 -85 4 88 46 -61 -82 13 90 38 -67 -78 22 90 31 -73}

[0390] {70 -43 -87 9 90 25 -80 -57 57 80 -25 -90 -9 87 43 -70 -70 43 87 -9 -90 -25 80

57 -57 -80 25 90 9 -87 -43 70}

[0391] {67 -54 -78 38 85 -22 -90 4 90 13 -88 -31 82 46 -73 -61 61 73 -46 -82 31 88 -13 -90 -4 90 22 -85 -38 78 54 -67}

[0392] {64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64}

[0393] {61 -73 -46 82 31 -88 -13 90 -4 -90 22 85 -38 -78 54 67 -67 -54 78 38 -85 -22 90 4 -90 13 88 -31 -82 46 73 -61}

[0394] {57 -80 -25 90 -9 -87 43 70 -70 -43 87 9 -90 25 80 -57 -57 80 25 -90 9 87 -43 -70 70 43 -87 -9 90 -25 -80 57}

[0395] {54 -85 -4 88 -46 -61 82 13 -90 38 67 -78 -22 90 -31 -73 73 31 -90 22 78 -67 -38 90 -13 -82 61 46 -88 4 85 -54}

[0396] {50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50 50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50}

[0397] {46 -90 38 54 -90 31 61 -88 22 67 -85 13 73 -82 4 78 -78 -4 82 -73 -13 85 -67 -22 88 -61 -31 90 -54 -38 90 -46}

[0398] {43 -90 57 25 -87 70 9 -80 80 -9 -70 87 -25 -57 90 -43 -43 90 -57 -25 87 -70 -9 80 -80 9 70 -87 25 57 -90 43}

[0399] {38 -88 73 -4 -67 90 -46 -31 85 -78 13 61 -90 54 22 -82 82 -22 -54 90 -61 -13 78 -85 31 46 -90 67 4 -73 88 -38}

[0400] {36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36}

[0401] {31 -78 90 -61 4 54 -88 82 -38 -22 73 -90 67 -13 -46 85 -85 46 13 -67 90 -73 22 38 -82 88 -54 -4 61 -90 78 -31}

[0402] {25 -70 90 -80 43 9 -57 87 -87 57 -9 -43 80 -90 70 -25 -25 70 -90 80 -43 -9 57 -87 87 -57 9 43 -80 90 -70 25}

[0403] {22 -61 85 -90 73 -38 -4 46 -78 90 -82 54 -13 -31 67 -88 88 -67 31 13 -54 82 -90 78 -46 4 38 -73 90 -85 61 -22}

[0404] {18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18 18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18}

[0405] {13 -38 61 -78 88 -90 85 -73 54 -31 4 22 -46 67 -82 90 -90 82 -67 46 -22 -4 31 -54 73 -85 90 -88 78 -61 38 -13}

[0406] {9 -25 43 -57 70 -80 87 -90 90 -87 80 -70 57 -43 25 -9 -9 25 -43 57 -70 80 -87 90 -90 87 -80 70 -57 43 -25 9}

[0407] {4 -13 22 -31 38 -46 54 -61 67 -73 78 -82 85 -88 90 -90 90 -90 88 -85 82 -78 73 -67 61 -54 46 -38 31 -22 13 -4}

[0408] }

[0410] 64-point DCT2

[0411] DCT2_64[m][n] ={

[0412] {64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}

64 64 64 64 64 64 64 64 64 64 64 64 64}

[0413] {91 90 90 90 88 87 86 84 83 81 79 77 73 71 69 65 62 59 56 52 48 44 41 37 33 28 24 20 15 11 7 2 -2 -7 -11 -15 -20 -24 -28 -33 -37 -41 -44 -48 -52 -56 -59 -62 -65 -69 -71 -73 -77 -79 -81 -83 -84 -86 -87 -88 -90 -90 -90 -91}

[0414] {90 90 88 85 82 78 73 67 61 54 46 38 31 22 13 4 -4 -13 -22 -31 -38 -46 -54 -61 -67 -73 -78 -82 -85 -88 -90 -90 -90 -90 -88 -85 -82 -78 -73 -67 -61 -54 -46 -38 -31 -22 -13 -4 4 13 22 31 38 46 54 61 67 73 78 82 85 88 90 90}

[0415] {90 88 84 79 71 62 52 41 28 15 2 -11 -24 -37 -48 -59 -69 -77 -83 -87 -90 -91 -90 -86 -81 -73 -65 -56 -44 -33 -20 -7 7 20 33 44 56 65 73 81 86 90 91 90 87 83 77 69 59 48 37 24 11 -2 -15 -28 -41 -52 -62 -71 -79 -84 -88 -90}

[0416] {90 87 80 70 57 43 25 9 -9 -25 -43 -57 -70 -80 -87 -90 -90 -87 -80 -70 -57 -43 -25 -9 9 25 43 57 70 80 87 90 90 87 80 70 57 43 25 9 -9 -25 -43 -57 -70 -80 -87 -90 -90 -87 -80 -70 -57 -43 -25 -9 9 25 43 57 70 80 87 90}

[0417] {90 84 73 59 41 20 -2 -24 -44 -62 -77 -86 -90 -90 -83 -71 -56 -37 -15 7 28 48 65 79 87 91 88 81 69 52 33 11 -11 -33 -52 -69 -81 -88 -91 -87 -79 -65 -48 -28 -7 15 37 56 71 83 90 90 86 77 62 44 24 2 -20 -41 -59 -73 -84 -90}

[0418] {90 82 67 46 22 -4 -31 -54 -73 -85 -90 -88 -78 -61 -38 -13 13 38 61 78 88 90 85 73 54 31 4 -22 -46 -67 -82 -90 -90 -82 -67 -46 -22 4 31 54 73 85 90 88 78 61 38 13 -13 -38 -61 -78 -88 -90 -85 -73 -54 -31 -4 22 46 67 82 90}

[0419] {90 79 59 33 2 -28 -56 -77 -88 -90 -81 -62 -37 -7 24 52 73 87 90 83 65 41 11 -20 -48 -71 -86 -91 -84 -69 -44 -15 15 44 69 84 91 86 71 48 20 -11 -41 -65 -83 -90 -87 -73 -52 -24 7 37 62 81 90 88 77 56 28 -2 -33 -59 -79 -90}

[0420] {89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89 89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89 89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89 89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89}

[0421] {88 71 41 2 -37 -69 -87 -90 -73 -44 -7 33 65 86 90 77 48 11 -28 -62 -84 -90 -79 -52 -15 24 59 83 91 81 56 20 -20 -56 -81 -91 -83 -59 -24 15 52 79 90 84 62 28 -11 -48 -77 -90 -86 -65 -33 7 44 73 90 87 69 37 -2 -41 -71 -88}

[0422] {88 67 31 -13 -54 -82 -90 -78 -46 -4 38 73 90 85 61 22 -22 -61 -85 -90 -73 -38 4 46 78 90 82 54 13 -31 -67 -88 -88 -67 -31 13 54 82 90 78 46 4 -38 -73 -90 -85 -61 -22 22 61 85 90 73 38 -4 -46 -78 -90 -82 -54 -13 31 67 88}

[0423] {87 62 20 -28 -69 -90 -84 -56 -11 37 73 90 81 48 2 -44 -79 -91 -77 -41 7 52 83 90 71 33 -15 -59 -86 -88 -65 -24 24 65 88 86 59 15 -33 -71 -90 -83 -52 -7 41 77 91 79 44 -2 -48 -81 -90 -73 -37 11 56 84 90 69 28 -20 -62 -87}

[0424] {87 57 9 -43 -80 -90 -70 -25 25 70 90 80 43 -9 -57 -87 -87 -57 -9 43 80 90 70 25 -25 -70 -90 -80 -43 9 57 87 87 57 9 -43 -80 -90 -70 -25 25 70 90 80 43 -9 -57 -87 -87 -57 -9 43 80 90 70 25 -25 -70 -90 -80 -43 9 57 87}

[0425] {86 52 -2 -56 -87 -84 -48 7 59 88 83 44 -11 -62 -90 -81 -41 15 65 90 79 37 -20 -69 -90 -77 -33 24 71 91 73 28 -28 -73 -91 -71 -24 33 77 90 69 20 -37 -79 -90 -65 -15 41 81 90 62 11 -44 -83 -88 -59 -7 48 84 87 56 2 -52 -86}

[0426] {85 46 -13 -67 -90 -73 -22 38 82 88 54 -4 -61 -90 -78 -31 31 78 90 61 4 -54 -88 -82 -38 22 73 90 67 13 -46 -85 -85 -46 13 67 90 73 22 -38 -82 -88 -54 4 61 90 78 31 -31 -78 -90 -61 -4 54 88 82 38 -22 -73 -90 -67 -13 46 85}

[0427] {84 41 -24 -77 -90 -56 7 65 91 69 11 -52 -88 -79 -28 37 83 86 44 -20 -73 -90 -59

2 62 90 71 15 -48 -87 -81 -33 33 81 87 48 -15 -71 -90 -62 -2 59 90 73 20 -44 -86 -83 -37 28 79 88 52 -11 -69 -91 -65 -7 56 90 77 24 -41 -84}

[0428] {83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83}

[0429] {83 28 -44 -88 -73 -11 59 91 62 -7 -71 -90 -48 24 81 84 33 -41 -87 -77 -15 56 90 65 -2 -69 -90 -52 20 79 86 37 -37 -86 -79 -20 52 90 69 2 -65 -90 -56 15 77 87 41 -33 -84 -81 -24 48 90 71 7 -62 -91 -59 11 73 88 44 -28 -83}

[0430] {82 22 -54 -90 -61 13 78 85 31 -46 -90 -67 4 73 88 38 -38 -88 -73 -4 67 90 46 -31 -85 -78 -13 61 90 54 -22 -82 -82 -22 54 90 61 -13 -78 -85 -31 46 90 67 -4 -73 -88 -38 38 88 73 4 -67 -90 -46 31 85 78 13 -61 -90 -54 22 82}

[0431] {81 15 -62 -90 -44 37 88 69 -7 -77 -84 -24 56 91 52 -28 -86 -73 -2 71 87 33 -48 -90 -59 20 83 79 11 -65 -90 -41 41 90 65 -11 -79 -83 -20 59 90 48 -33 -87 -71 2 73 86 28 -52 -91 -56 24 84 77 7 -69 -88 -37 44 90 62 -15 -81}

[0432] {80 9 -70 -87 -25 57 90 43 -43 -90 -57 25 87 70 -9 -80 -80 -9 70 87 25 -57 -90 -43 43 90 57 -25 -87 -70 9 80 80 9 -70 -87 -25 57 90 43 -43 -90 -57 25 87 70 -9 -80 -80 -9 70 87 25 -57 -90 -43 43 90 57 -25 -87 -70 9 80}

[0433] {79 2 -77 -81 -7 73 83 11 -71 -84 -15 69 86 20 -65 -87 -24 62 88 28 -59 -90 -33 56 90 37 -52 -90 -41 48 91 44 -44 -91 -48 41 90 52 -37 -90 -56 33 90 59 -28 -88 -62 24 87 65 -20 -86 -69 15 84 71 -11 -83 -73 7 81 77 -2 -79}

[0434] {78 -4 -82 -73 13 85 67 -22 -88 -61 31 90 54 -38 -90 -46 46 90 38 -54 -90 -31 61 88 22 -67 -85 -13 73 82 4 -78 -78 4 82 73 -13 -85 -67 22 88 61 -31 -90 -54 38 90 46 -46 -90 -38 54 90 31 -61 -88 -22 67 85 13 -73 -82 -4 78}

[0435] {77 -11 -86 -62 33 90 44 -52 -90 -24 69 83 2 -81 -71 20 88 56 -41 -91 -37 59 87 15 -73 -79 7 84 65 -28 -90 -48 48 90 28 -65 -84 -7 79 73 -15 -87 -59 37 91 41 -56 -88 -20 71 81 -2 -83 -69 24 90 52 -44 -90 -33 62 86 11 -77}

[0436] {75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75 75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75 75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75 75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75}

[0437] {73 -24 -90 -37 65 81 -11 -88 -48 56 86 2 -84 -59 44 90 15 -79 -69 33 91 28 -71 -77 20 90 41 -62 -83 7 87 52 -52 -87 -7 83 62 -41 -90 -20 77 71 -28 -91 -33 69 79 -15 -90 -44 59 84 -2 -86 -56 48 88 11 -81 -65 37 90 24 -73}

[0438] {73 -31 -90 -22 78 67 -38 -90 -13 82 61 -46 -88 -4 85 54 -54 -85 4 88 46 -61 -82 13 90 38 -67 -78 22 90 31 -73 -73 31 90 22 -78 -67 38 90 13 -82 -61 46 88 4 -85 -54 54 85 -4 -88 -46 61 82 -13 -90 -38 67 78 -22 -90 -31 73}

[0439] {71 -37 -90 -7 86 48 -62 -79 24 91 20 -81 -59 52 84 -11 -90 -33 73 69 -41 -88 -2 87 44 -65 -77 28 90 15 -83 -56 56 83 -15 -90 -28 77 65 -44 -87 2 88 41 -69 -73 33 90 11 -84 -52 59 81 -20 -91 -24 79 62 -48 -86 7 90 37 -71}

[0440] {70 -43 -87 9 90 25 -80 -57 57 80 -25 -90 -9 87 43 -70 -70 43 87 -9 -90 -25 80 57 -57 -80 25 90 9 -87 -43 70 70 -43 -87 9 90 25 -80 -57 57 80 -25 -90 -9 87 43 -70 -70 43 87 -9 -90 -25 80 57 -57 -80 25 90 9 -87 -43 70}

[0441] {69 -48 -83 24 90 2 -90 -28 81 52 -65 -71 44 84 -20 -90 -7 88 33 -79 -56 62 73 -41 -86 15 91 11 -87 -37 77 59 -59 -77 37 87 -11 -91 -15 86 41 -73 -62 56 79 -33 -88 7 90 20 -84 -44 71 65 -52 -81 28 90 -2 -90 -24 83 48 -69}

[0442] {67 -54 -78 38 85 -22 -90 4 90 13 -88 -31 82 46 -73 -61 61 73 -46 -82 31 88 -13 -90 -4 90 22 -85 -38 78 54 -67 -67 54 78 -38 -85 22 90 -4 -90 -13 88 31 -82 -46 73 61 -61 -73 46 82 -31 -88 13 90 4 -90 -22 85 38 -78 -54 67}

[0443] {65 -59 -71 52 77 -44 -81 37 84 -28 -87 20 90 -11 -90 2 91 7 -90 -15 88 24 -86 -33 83 41 -79 -48 73 56 -69 -62 62 69 -56 -73 48 79 -41 -83 33 86 -24 -88 15 90 -7 -91 -2 90 11 -90 -20 87 28 -84 -37 81 44 -77 -52 71 59 -65}

[0444] {64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64}

[0445] {62 -69 -56 73 48 -79 -41 83 33 -86 -24 88 15 -90 -7 91 -2 -90 11 90 -20 -87 28 84 -37 -81 44 77 -52 -71 59 65 -65 -59 71 52 -77 -44 81 37 -84 -28 87 20 -90 -11 90 2 -91 7 90 -15 -88 24 86 -33 -83 41 79 -48 -73 56 69 -62}

[0446] {61 -73 -46 82 31 -88 -13 90 -4 -90 22 85 -38 -78 54 67 -67 -54 78 38 -85 -22 90 4 -90 13 88 -31 -82 46 73 -61 -61 73 46 -82 -31 88 13 -90 4 90 -22 -85 38 78 -54 -67 67 54 -78 -38 85 22 -90 -4 90 -13 -88 31 82 -46 -73 61}

[0447] {59 -77 -37 87 11 -91 15 86 -41 -73 62 56 -79 -33 88 7 -90 20 84 -44 -71 65 52 -81 -28 90 2 -90 24 83 -48 -69 69 48 -83 -24 90 -2 -90 28 81 -52 -65 71 44 -84 -20 90 -7 -88 33 79 -56 -62 73 41 -86 -15 91 -11 -87 37 77 -59}

[0448] {57 -80 -25 90 -9 -87 43 70 -70 -43 87 9 -90 25 80 -57 -57 80 25 -90 9 87 -43 -70 70 43 -87 -9 90 -25 -80 57 57 -80 -25 90 -9 -87 43 70 -70 -43 87 9 -90 25 80 -57 -57 80 25 -90 9 87 -43 -70 70 43 -87 -9 90 -25 -80 57}

[0449] {56 -83 -15 90 -28 -77 65 44 -87 -2 88 -41 -69 73 33 -90 11 84 -52 -59 81 20 -91 24 79 -62 -48 86 7 -90 37 71 -71 -37 90 -7 -86 48 62 -79 -24 91 -20 -81 59 52 -84 -11 90 -33 -73 69 41 -88 2 87 -44 -65 77 28 -90 15 83 -56}

[0450] {54 -85 -4 88 -46 -61 82 13 -90 38 67 -78 -22 90 -31 -73 73 31 -90 22 78 -67 -38 90 -13 -82 61 46 -88 4 85 -54 -54 85 4 -88 46 61 -82 -13 90 -38 -67 78 22 -90 31 73 -73 -31 90 -22 -78 67 38 -90 13 82 -61 -46 88 -4 -85 54}

[0451] {52 -87 7 83 -62 -41 90 -20 -77 71 28 -91 33 69 -79 -15 90 -44 -59 84 2 -86 56 48 -88 11 81 -65 -37 90 -24 -73 73 24 -90 37 65 -81 -11 88 -48 -56 86 -2 -84 59 44 -90 15 79 -69 -33 91 -28 -71 77 20 -90 41 62 -83 -7 87 -52}

[0452] {50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50 50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50 50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50 50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50}

[0453] {48 -90 28 65 -84 7 79 -73 -15 87 -59 -37 91 -41 -56 88 -20 -71 81 2 -83 69 24 -90 52 44 -90 33 62 -86 11 77 -77 -11 86 -62 -33 90 -44 -52 90 -24 -69 83 -2 -81 71 20 -88 56 41 -91 37 59 -87 15 73 -79 -7 84 -65 -28 90 -48}

[0454] {46 -90 38 54 -90 31 61 -88 22 67 -85 13 73 -82 4 78 -78 -4 82 -73 -13 85 -67 -22 88 -61 -31 90 -54 -38 90 -46 -46 90 -38 -54 90 -31 -61 88 -22 -67 85 -13 -73 82 -4 -78 78 4 -82 73 13 -85 67 22 -88 61 31 -90 54 38 -90 46}

[0455] {44 -91 48 41 -90 52 37 -90 56 33 -90 59 28 -88 62 24 -87 65 20 -86 69 15 -84 71 11 -83 73 7 -81 77 2 -79 79 -2 -77 81 -7 -73 83 -11 -71 84 -15 -69 86 -20 -65 87 -24 -62 88 -28 -59 90 -33 -56 90 -37 -52 90 -41 -48 91 -44}

[0456] {43 -90 57 25 -87 70 9 -80 80 -9 -70 87 -25 -57 90 -43 -43 90 -57 -25 87 -70 -9 80 -80 9 70 -87 25 57 -90 43 43 -90 57 25 -87 70 9 -80 80 -9 -70 87 -25 -57 90

-43 -43 90 -57 -25 87 -70 -9 80 -80 9 70 -87 25 57 -90 43}

[0457] {41 -90 65 11 -79 83 -20 -59 90 -48 -33 87 -71 -2 73 -86 28 52 -91 56 24 -84 77 -7 -69 88 -37 -44 90 -62 -15 81 -81 15 62 -90 44 37 -88 69 7 -77 84 -24 -56 91 -52 -28 86 -73 2 71 -87 33 48 -90 59 20 -83 79 -11 -65 90 -41}

[0458] {38 -88 73 -4 -67 90 -46 -31 85 -78 13 61 -90 54 22 -82 82 -22 -54 90 -61 -13 78 -85 31 46 -90 67 4 -73 88 -38 -38 88 -73 4 67 -90 46 31 -85 78 -13 -61 90 -54 -22 82 -82 22 54 -90 61 13 -78 85 -31 -46 90 -67 -4 73 -88 38}

[0459] {37 -86 79 -20 -52 90 -69 2 65 -90 56 15 -77 87 -41 -33 84 -81 24 48 -90 71 -7 -62 91 -59 -11 73 -88 44 28 -83 83 -28 -44 88 -73 11 59 -91 62 7 -71 90 -48 -24 81 -84 33 41 -87 77 -15 -56 90 -65 -2 69 -90 52 20 -79 86 -37}

[0460] {36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36}

[0461] {33 -81 87 -48 -15 71 -90 62 -2 -59 90 -73 20 44 -86 83 -37 -28 79 -88 52 11 -69 91 -65 7 56 -90 77 -24 -41 84 -84 41 24 -77 90 -56 -7 65 -91 69 -11 -52 88 -79 28 37 -83 86 -44 -20 73 -90 59 2 -62 90 -71 15 48 -87 81 -33}

[0462] {31 -78 90 -61 4 54 -88 82 -38 -22 73 -90 67 -13 -46 85 -85 46 13 -67 90 -73 22 38 -82 88 -54 -4 61 -90 78 -31 -31 78 -90 61 -4 -54 88 -82 38 22 -73 90 -67 13 46 -85 85 -46 -13 67 -90 73 -22 -38 82 -88 54 4 -61 90 -78 31}

[0463] {28 -73 91 -71 24 33 -77 90 -69 20 37 -79 90 -65 15 41 -81 90 -62 11 44 -83 88 -59 7 48 -84 87 -56 2 52 -86 86 -52 -2 56 -87 84 -48 -7 59 -88 83 -44 -11 62 -90 81 -41 -15 65 -90 79 -37 -20 69 -90 77 -33 -24 71 -91 73 -28}

[0464] {25 -70 90 -80 43 9 -57 87 -87 57 -9 -43 80 -90 70 -25 -25 70 -90 80 -43 -9 57 -87 87 -57 9 43 -80 90 -70 25 25 -70 90 -80 43 9 -57 87 -87 57 -9 -43 80 -90 70 -25 -25 70 -90 80 -43 -9 57 -87 87 -57 9 43 -80 90 -70 25}

[0465] {24 -65 88 -86 59 -15 -33 71 -90 83 -52 7 41 -77 91 -79 44 2 -48 81 -90 73 -37 -11 56 -84 90 -69 28 20 -62 87 -87 62 -20 -28 69 -90 84 -56 11 37 -73 90 -81 48 -2 -44 79 -91 77 -41 -7 52 -83 90 -71 33 15 -59 86 -88 65 -24}

[0466] {22 -61 85 -90 73 -38 -4 46 -78 90 -82 54 -13 -31 67 -88 88 -67 31 13 -54 82 -90 78 -46 4 38 -73 90 -85 61 -22 -22 61 -85 90 -73 38 4 -46 78 -90 82 -54 13 31 -67 88 -88 67 -31 -13 54 -82 90 -78 46 -4 -38 73 -90 85 -61 22}

[0467] {20 -56 81 -91 83 -59 24 15 -52 79 -90 84 -62 28 11 -48 77 -90 86 -65 33 7 -44 73 -90 87 -69 37 2 -41 71 -88 88 -71 41 -2 -37 69 -87 90 -73 44 -7 -33 65 -86 90 -77 48 -11 -28 62 -84 90 -79 52 -15 -24 59 -83 91 -81 56 -20}

[0468] {18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18 18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18 18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18 18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18}

[0469] {15 -44 69 -84 91 -86 71 -48 20 11 -41 65 -83 90 -87 73 -52 24 7 -37 62 -81 90 -88 77 -56 28 2 -33 59 -79 90 -90 79 -59 33 -2 -28 56 -77 88 -90 81 -62 37 -7 -24 52 -73 87 -90 83 -65 41 -11 -20 48 -71 86 -91 84 -69 44 -15}

[0470] {13 -38 61 -78 88 -90 85 -73 54 -31 4 22 -46 67 -82 90 -90 82 -67 46 -22 -4 31 -54 73 -85 90 -88 78 -61 38 -13 -13 38 -61 78 -88 90 -85 73 -54 31 -4 -22 46 -67 82 -90 90 -82 67 -46 22 4 -31 54 -73 85 -90 88 -78 61 -38 13}

[0471] {11 -33 52 -69 81 -88 91 -87 79 -65 48 -28 7 15 -37 56 -71 83 -90 90 -86 77 -62

44 -24 2 20 -41 59 -73 84 -90 90 -84 73 -59 41 -20 -2 24 -44 62 -77 86 -90 90 -83 71 -56 37 -15 -7 28 -48 65 -79 87 -91 88 -81 69 -52 33 -11}

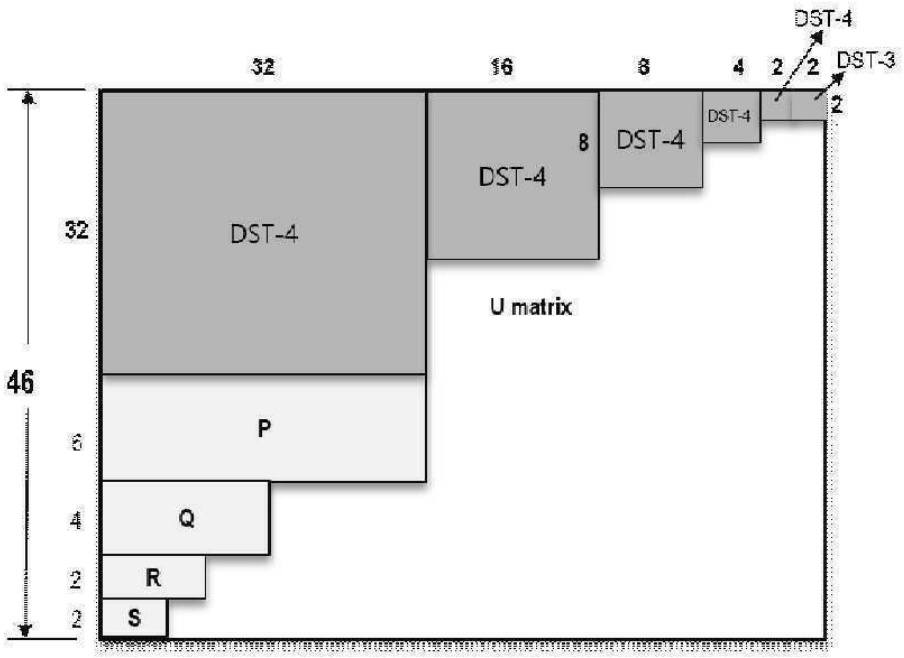
[0472] {9 -25 43 -57 70 -80 87 -90 90 -87 80 -70 57 -43 25 -9 -9 25 -43 57 -70 80 -87 90 -90 87 -80 70 -57 43 -25 9 9 -25 43 -57 70 -80 87 -90 90 -87 80 -70 57 -43 25 -9 -9 25 -43 57 -70 80 -87 90 -90 87 -80 70 -57 43 -25 9}

[0473] {7 -20 33 -44 56 -65 73 -81 86 -90 91 -90 87 -83 77 -69 59 -48 37 -24 11 2 -15 28 -41 52 -62 71 -79 84 -88 90 -90 88 -84 79 -71 62 -52 41 -28 15 -2 -11 24 -37 48 -59 69 -77 83 -87 90 -91 90 -86 81 -73 65 -56 44 -33 20 -7}

[0474] {4 -13 22 -31 38 -46 54 -61 67 -73 78 -82 85 -88 90 -90 90 -90 88 -85 82 -78 73 -67 61 -54 46 -38 31 -22 13 -4 -4 13 -22 31 -38 46 -54 61 -67 73 -78 82 -85 88 -90 90 -90 90 -88 85 -82 78 -73 67 -61 54 -46 38 -31 22 -13 4}

[0475] {2 -7 11 -15 20 -24 28 -33 37 -41 44 -48 52 -56 59 -62 65 -69 71 -73 77 -79 81 -83 84 -86 87 -88 90 -90 90 -91 91 -90 90 -90 88 -87 86 -84 83 -81 79 -77 73 -71 69 -65 62 -59 56 -52 48 -44 41 -37 33 -28 24 -20 15 -11 7 -2}

[0476] }



[0479]

[0480] 그림 21. 제안하는 커널 유도 생성 방법을 사용하였을 때 메모리 사용량

[0481] 본 발명에서 제안하는 커널 유도 생성 방법을 사용하였을 때, 8비트 행렬 요소를 사용할 경우 그림 21과 같이 총 1648 바이트의 정적 메모리를 사용할 수 있다. 이 것으로부터 N=64, N=32, N=16, N=8, N=4, N=2 DCT-2와 N=32, N=16, N=8, N=4 DST-7, N=32, N=16, N=8, N=4 DCT-8을 생성할 수 있으며 기존 모든 커널을 저장하여 사용하였을 때 대비 약 30% 수준의 메모리 사용량이다.

[0483] 상술한 본 발명의 실시예들은 다양한 수단을 통해 구현될 수 있다. 예를 들어, 본 발명의 실시예들은 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다.

[0484] 하드웨어에 의한 구현의 경우, 본 발명의 실시예들에 따른 방법은 하나 또는 그 이상의 ASICs(Application Specific Integrated Circuits), DSPs(Digital Signal Processors), DSPDs(Digital Signal Processing

Devices), PLDs(Programmable Logic Devices), FPGAs(Field Programmable Gate Arrays), 프로세서, 컨트롤러, 마이크로 컨트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

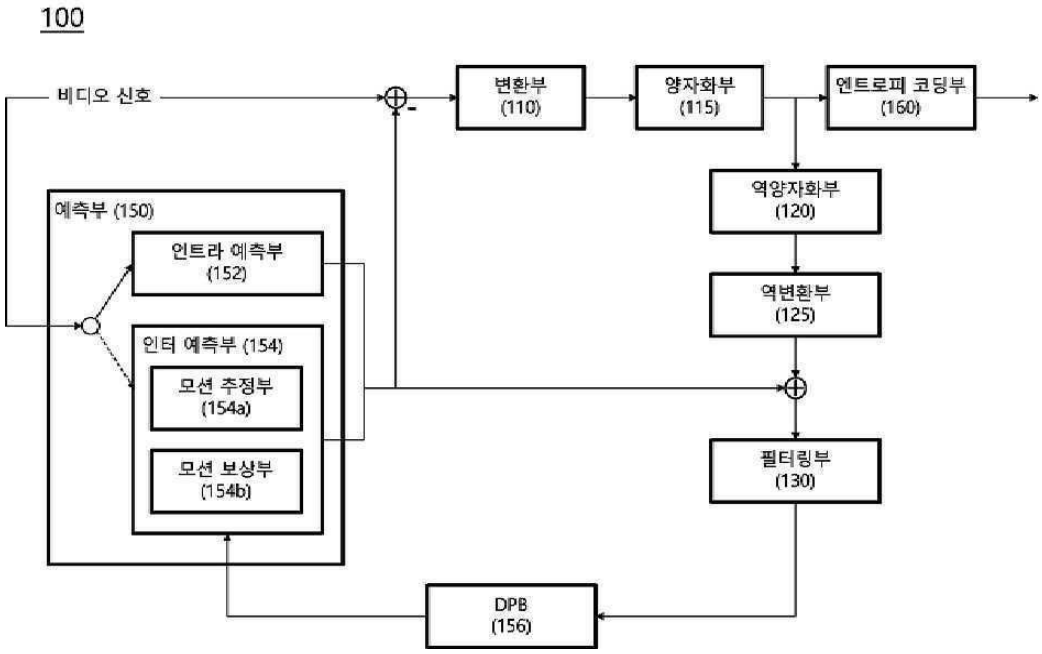
[0485] 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명의 실시예들에 따른 방법은 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차 또는 함수 등의 형태로 구현될 수 있다. 소프트웨어 코드는 메모리에 저장되어 프로세서에 의해 구동될 수 있다. 상기 메모리는 프로세서의 내부 또는 외부에 위치할 수 있으며, 이미 공지된 다양한 수단에 의해 프로세서와 데이터를 주고 받을 수 있다.

[0486] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예 들은 모든 면에서 예시적인 것이며 한정적인 것이 아는 것으로 해석해야 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

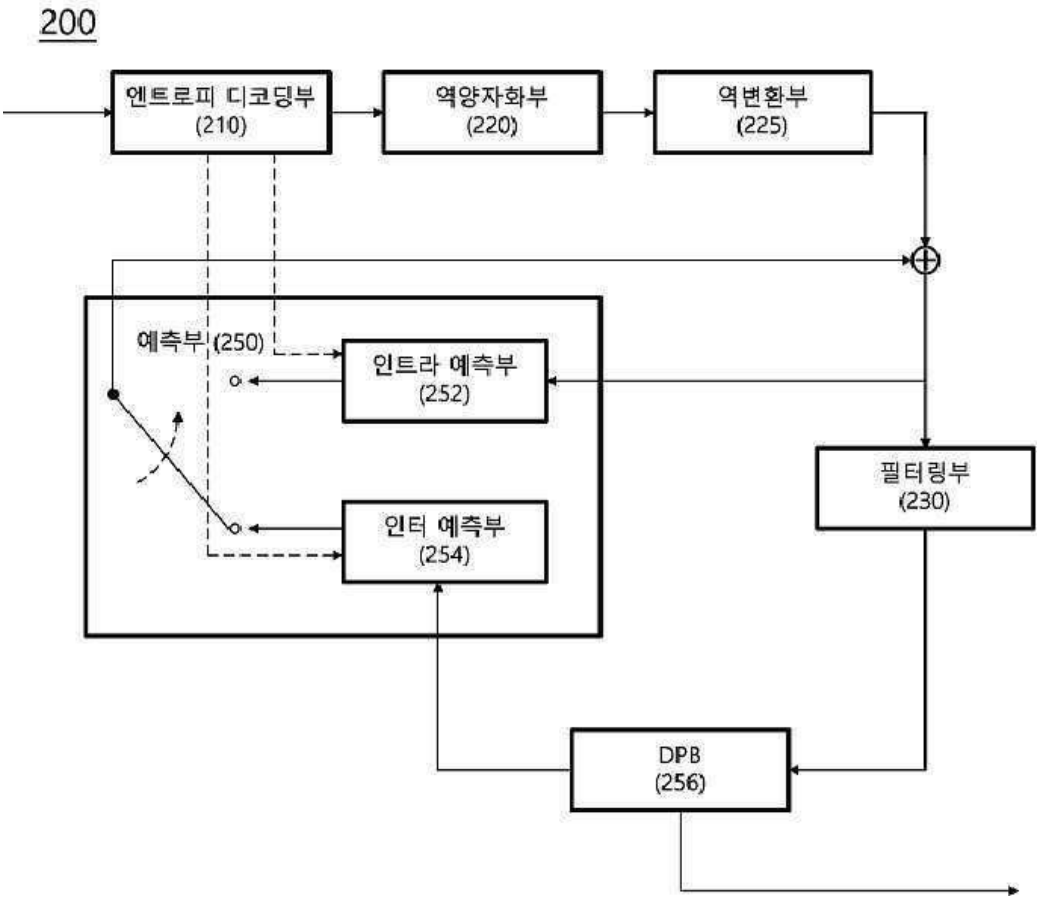
[0487] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

도면

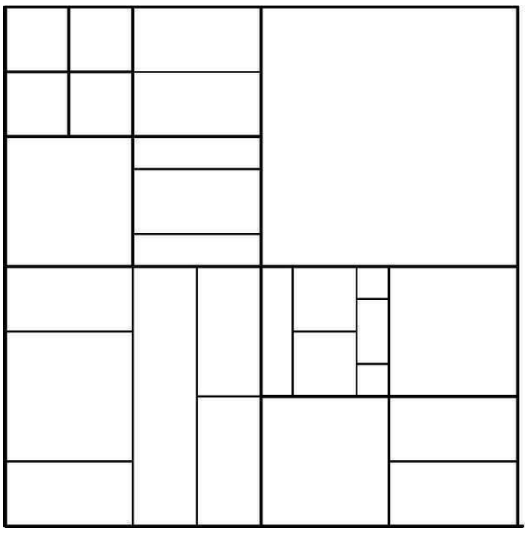
도면1



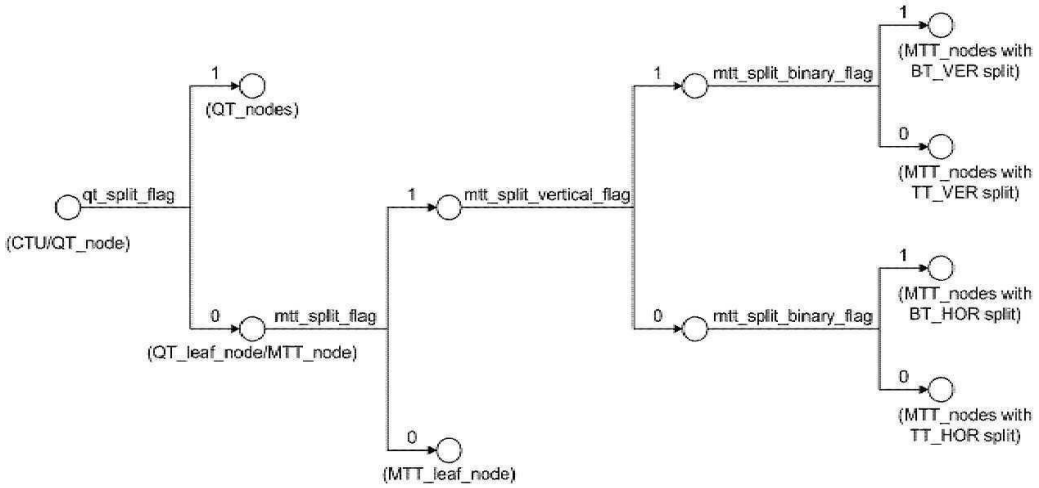
도면2


200
엔트로피 디코딩부
(210)
역양자화부
(220)
역변환부
(225)
예측부 (250)
인트라 예측부
(252)
인터 예측부
(254)
필터링부
(230)
DPB
(256)

도면3



도면4


qt_split_flag
(CTU/QT_node)
1
(QT_nodes)
0
mtt_split_flag
(QT_leaf_node/MTT_node)
1
mtt_split_vertical_flag
0
(MTT_leaf_node)
1
mtt_split_binary_flag
0
mtt_split_binary_flag
1
(MTT_nodes with BT_VER split)
0
(MTT_nodes with TT_VER split)
1
(MTT_nodes with BT_HOR split)
0
(MTT_nodes with TT_HOR split)