

# 통합 비디오 코덱을 위한 4x4/8x8 DCT와 양자화 회로의 고성능 구조

이 선 영<sup>†</sup> · 조 경 순<sup>††</sup>

## 요 약

본 논문은 통합 비디오 코덱에 적용할 수 있는 DCT와 양자화 회로에 대한 고성능 구조를 제안한다. 제안된 구조는 JPEG, MPEG-1/2/4, H.264, VC-1과 같은 동영상 압축 표준들에 사용되는 모든 변환과 양자화에 적용할 수 있다. 통합 DCT 회로 구조를 위해 8x8 DCT의 변환행렬을 재배치하는 순열행렬을 정의하였고 4x4 DCT의 변환행렬과 통합하기 위해 8x8 변환행렬을 4개의 4x4 변환행렬로 나누었다. 8x8 DCT는 재배치와 분할된 변환행렬을 기반으로 4x4 DCT 연산을 반복하여 수행된다. 구현된 회로는 사용자가 변환 계수를 입력하기 때문에 앞으로 등장할 어떤 종류의 DCT 변환에도 매우 쉽게 확장할 수 있다. DCT 회로의 곱셈기들은 회로 크기를 최소화하기 위해 양자화 회로에서 사용되는 곱셈기들과 공유하였다. 이때, 양자화 회로는 회로 구현에 필요한 자원과 처리 시간의 증가 없이 DCT 회로와 통합된다. 제안된 DCT와 양자화 회로는 RTL로 구현하였고 FPGA가 탑재된 보드에서 동작을 검증하였다.

키워드 : DCT, 양자화, 통합 비디오 코덱, 회로구조

## High-Performance Architecture of 4x4/8x8 DCT and Quantization Circuit for Unified Video CODEC

Seonyoung Lee<sup>†</sup> · Kyeongsoon Cho<sup>††</sup>

## ABSTRACT

This paper proposes the new high-performance circuit architecture of the transform and quantization for unified video CODEC. The proposed architecture can be applied to all kinds of transforms and quantizations for the video compression standards such as JPEG, MPEG-1/2/4, H.264 and VC-1. We defined the permutation matrices to reorder the transform matrix of the 8x8 DCT and partitioned the reordered 8x8 transform matrix into four 4x4 sub-matrices. The 8x8 DCT is performed by repeating the 4x4 DCT's based on the reordered and partitioned transform matrices. Since our circuit accepts the transform coefficients from the users, it can be extended very easily to cover any kind of DCT-based transforms for future standards. The multipliers in the DCT circuit are shared by the quantization circuit in order to minimize the circuit size. The quantization circuit is merged into the DCT circuit without any significant increase of circuit resources and processing time. We described the proposed DCT and quantization circuit at RTL, and verified its operation on FPGA board.

Keywords : DCT, Quantization, Unified Video CODEC, Circuit Architecture

## 1. 서 론

고해상도 카메라, 휴대폰, 스마트폰, DMB (digital multimedia broadcasting) 등과 같은 대부분의 휴대용 멀티미디어 제품들은 한 제품 안에 여러 종류의 멀티미디어 코

덱 (encoder/decoder, CODEC)들이 내장되어 있다. 이 기능들을 구현하기 위해 기기 내에 여러 종류의 하드웨어 비디오 코덱을 사용하는 것은 제품의 단가를 높이게 된다. 이 문제를 해결하는 것은 여러 표준을 지원할 수 있는 단일 칩을 사용하는 것이다. 기존에 연구된 DCT (discrete cosine transform)는 곱셈기와 덧셈기를 이용하는 구조, CORDIC (coordinate rotation digital computer) 기반 구조, 열 기반 구조와 메모리 기반 구조로 나눌 수 있다.[1] 그러나 기존의 방법들은 전용 구조만을 지원하여 현재 존재하는 모든 DCT 및 양자화를 지원할 수 없다. 본 논문에서는 통합 비디오 코덱에 적용할 수 있는 고성능 DCT와 양자화 회로 구조에

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단(No.20090067004)과 2011년도 한국외국어대학교 교내학술연구비의 지원에 의한 것임.

† 정 회 원 : 전자부품연구원 융합신호SoC연구센터 선임연구원

†† 정 회 원 : 한국외국어대학교 전자공학과 정교수

논문접수: 2011년 1월 13일

수정일: 1차 2011년 2월 11일

심사완료: 2011년 2월 23일

대해 설명한다. 본 논문에서 제안한 고성능 4x4/8x8 DCT, 양자화 회로 구조는 JPEG[2], MPEG-1/2, MPEG-4[3], H.264[4], VC-1[5]를 지원할 수 있는 통합 비디오 코덱에 적용 가능하다.

제안된 DCT 회로는 4개의 PE (processor element)와 다용도 버퍼를 갖는다. 각 PE는 1D (one-dimensional) 4x4 DCT를 처리한다. 1D 8x8 DCT는 1D 4x4 DCT를 반복 수행하여 처리한다. 다용도 버퍼들은 2D (two-dimensional) DCT를 위해 치환 버퍼 또는 데이터를 재 정렬을 위한 순열 버퍼로 사용한다. 본 논문의 회로는 사용자들이 변환 계수들을 변경하여 DCT에 기반을 둔 변환을 처리할 수 있다. 사용자가 변환 계수들을 외부에서 바꾸는 것을 통해 제안한 회로는 DCT 기반의 모든 변환을 처리할 수 있다. 제안된 구조는 앞으로 표준화될 모든 종류의 DCT 변환들을 구현하는데 쉽게 적용될 수 있다. 변환 회로의 곱셈기들은 DCT와 양자화 회로에서 서로 공유된다. 양자화 동작은 변환 동작을 하는 동안 여유 클럭 사이클을 이용하기 때문에 양자화 동작만을 위해 별도의 클럭 사이클을 사용하지 않아도 된다. 제안된 DCT와 양자화 회로 구조는 Verilog HDL (hardware description language)을 이용하여 RTL (register transfer level)로 기술하였고 Xilinx Virtex4 LX60 FPGA에서 동작을 검증하였다. RTL 회로는 130nm 표준 셀 라이브러리를 이용하여 게이트 수준 회로로 합성하였으며 다른 접근 방법과 비교하여 우수한 성능을 보였다.

## 2. 재정렬과 분할을 이용한 변환 행렬

대부분의 영상 압축 표준들은 4x4와 8x8 DCT를 필요로 한다. 본 논문에서는 1D 4x4 DCT의 결과를 이용하여 1D 8x8 DCT를 처리하였다. 4x4 DCT는 순열 행렬을 이용하여 표현 가능하다. H.264나 VC-1의 4x4 DCT 계수들은 (1)과 같이 3 가지 ( $h, i, j$ ) 계수들로 분류할 수 있다. 4x4 DCT의 계수 행렬은 (2)에 정의한 것과 같이 2개의 순열 행렬로 재정렬할 수 있다.[6] 이와 같은 순열 행렬들은 다음과 같은 특성을 만족한다.  $P_{4c}P_{4c}^T = P_{4c}^TP_{4c} = I_4$ ,  $P_{4r}P_{4r}^T = P_{4r}^TP_{4r} = I_4$ . (3)과 같이 정의된 4x4 DCT  $D_{4p}$ 는 (4)와 같이 표현할 수 있다. 4x4 순방향 DCT 동작은 (5)와 같이 표현된다. 이때  $X_{4p}$ 는 입력되는 4x4 행렬 데이터이다.  $W_{4p} = P_{4r}X_{4p}P_{4r}$ ,  $Z_{4p} = D_{4p}W_{4p}D_{4p}^T$ 로 정의하면 (6)과 같은 식으로 정리된다.

$$C_{4p} = \begin{pmatrix} h & h & h & h \\ i & j & -j & -i \\ h & -h & -h & h \\ j & -i & i & -j \end{pmatrix} \quad (1)$$

$$P_{4c} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad P_{4r} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$P_{4c}C_{4p}P_{4r} = D_{4p} \quad (3)$$

$$D_{4p} = \begin{pmatrix} h & h & h & h \\ h & -h & -h & h \\ i & j & -j & -i \\ j & -i & i & -j \end{pmatrix} \quad (4)$$

$$Y_{4p} = C_{4p}X_{4p}C_{4p}^T = (P_{4c}D_{4p}P_{4r})X_{4p}(P_{4c}D_{4p}P_{4r})^T \quad (5)$$

$$Y_{4p} = P_{4c}(D_{4p}W_{4p}D_{4p}^T)P_{4c} = P_{4c}Z_{4p}P_{4c} \quad (6)$$

JPEG, MPEG-1/2/4, VC-1의 8x8 DCT 계수들은 (7)과 같이 7개의 다른 계수들 ( $a, b, c, d, e, f, g$ )로 분류할 수 있다.[7] 본 논문에서는 8x8 DCT에 사용하기 위한 (8), (9)와 같은 순열 행렬을 제안하였다. 제안된 대칭 단위 행렬들은  $P_{8c}P_{8c}^T = P_{8c}^TP_{8c} = I_8$ 와  $P_{8r}P_{8r}^T = P_{8r}^TP_{8r} = I_8$ 을 만족한다. 8x8 DCT  $C_{8p}$ 의 계수 행렬은 4x4 DCT와 같은 방법으로 순열 행렬을 이용하여 재정렬될 수 있고  $D_{8p}$ 는 (10)과 같이 얻어진다.  $C_{8p}$ 는 8x8 DCT의 원래 변환 행렬이다. 2D 8x8 DCT는 (11)과 같이 나타낼 수 있다. 여기서  $W_{8p} = P_{8r}X_{8p}P_{8r}$ 이고  $Z_{8p} = D_{8p}W_{8p}D_{8p}^T$ 이다.  $X_{8p}$ 는 8x8 입력 데이터 행렬이다. (10)에 의해 정의된  $D_{8p}$ 는 (12)와 같이 표현할 수 있다. 그 후에 (13)에 나타난 것과 같이  $D_{8p}$ 는 (14)의  $D_{00}, D_{01}, D_{10}, D_{11}$ 으로 이루어지는 4개의 행렬로 나뉜다. (4)의  $D_{4p}$ 와  $D_{00}$ 은 매우 유사한 구조를 갖는 것을 발견할 수 있다. 이를 통해 4x4 DCT와 8x8 DCT 사이에 동작들을 공유할 수 있다. 1D 8x8 순방향 DCT는 (15)와 같이 4x4 행렬들을 이용하여 나타낼 수 있다. 8x8 행렬들의 곱셈 연산은 4x4 행렬의 곱셈 결과들의 합으로 처리하게 된다. 제안된 회로에서 중간 결과인  $D_{00}W_{00}, D_{10}W_{00}, D_{00}W_{01}, D_{10}W_{01}$ 은 버퍼에 저장되고  $D_{01}W_{10}, D_{11}W_{10}, D_{01}W_{11}, D_{11}W_{11}$ 에 더해지게 된다.

$$C_{8p} = \begin{pmatrix} a & a & a & a & a & a & a & a \\ b & c & d & e & -e & -d & -c & -b \\ f & -g & g & -f & -f & g & -g & f \\ c & -e & -b & -d & d & b & e & -c \\ a & -a & -a & a & a & -a & -a & a \\ d & -b & e & c & -c & -e & b & -d \\ -g & -f & f & g & g & f & -f & -g \\ e & -d & c & -b & b & -c & d & -e \end{pmatrix} \quad (7)$$

$$P_{8c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (8)$$

$$P_{8r} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (9)$$

$$P_{8c}C_{8p}P_{8r} = D_{8p} \quad (10)$$

$$Y_{8p} = C_{8p}X_{8p}C_{8p}^T = P_{8c}(D_{8p}W_{8p}D_{8p}^T)P_{8c} = P_{8c}Z_{8p}P_{8c} \quad (11)$$

$$D_{8p} = \begin{pmatrix} a & a & a & a & a & a & a & a \\ f & g & g & f & -f & -g & -g & -f \\ b & c & -c & -b & -e & -d & d & e \\ c & -e & e & -c & d & b & -b & -d \\ a & -a & -a & a & a & -a & -a & a \\ g & -f & -f & g & -g & f & f & -g \\ d & -b & b & -d & -c & -e & e & c \\ e & -d & d & -e & b & -c & c & -b \end{pmatrix} \quad (12)$$

$$D_{8p} = \begin{bmatrix} D_{00} & D_{01} \\ D_{10} & D_{11} \end{bmatrix} \quad (13)$$

$$D_{00} = \begin{pmatrix} a & a & a & a \\ f & g & g & f \\ b & c & -c & -b \\ c & -e & e & -c \end{pmatrix} \quad D_{01} = \begin{pmatrix} a & a & a & a \\ -f & -g & -g & -f \\ -e & -d & d & e \\ d & b & -b & -d \end{pmatrix}$$

$$D_{10} = \begin{pmatrix} a & -a & -a & a \\ g & -f & -f & g \\ d & -b & b & -d \\ e & -d & d & -e \end{pmatrix} \quad D_{11} = \begin{pmatrix} a & -a & -a & a \\ -g & f & f & -g \\ -c & -e & e & c \\ b & -c & c & -b \end{pmatrix} \quad (14)$$

$$D_{8p}W_{8p} = \begin{bmatrix} D_{00} & D_{01} \\ D_{10} & D_{11} \end{bmatrix} \begin{bmatrix} W_{00} & W_{01} \\ W_{10} & W_{11} \end{bmatrix} = \begin{bmatrix} D_{00}W_{00} + D_{01}W_{10} & D_{00}W_{01} + D_{01}W_{11} \\ D_{10}W_{00} + D_{11}W_{10} & D_{10}W_{01} + D_{11}W_{11} \end{bmatrix} \quad (15)$$

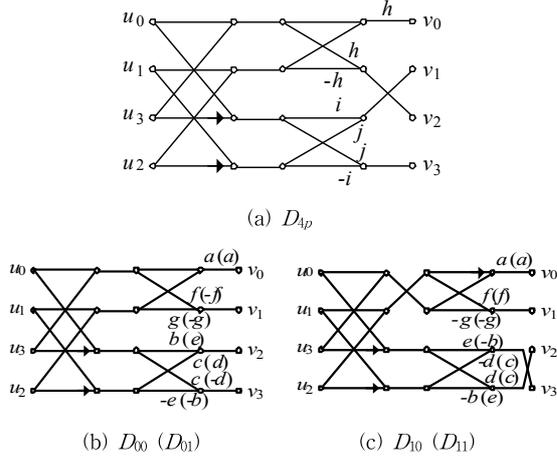
### 3. 제안한 회로 구조

본 논문은 순환 행렬과 행렬 분할을 이용하여 하나의 회로에서 4x4 DCT와 8x8 DCT를 처리할 수 있는 새로운 회로 구조를 제안한다. (그림 1)은 재 정렬된 4x4 변환 행렬의 신호 흐름을 나타낸 그래프이다. 4x4 DCT는  $D_{4p}$ 를 사용하고 8x8 DCT는  $D_{00}$ ,  $D_{01}$ ,  $D_{10}$ ,  $D_{11}$ 를 사용한다. 모든 그래프들은 유사한 구조를 갖고 있고 계수 값들만 차이가 있다. 4x4 변환 행렬  $C_{4p}$ 는 3개의 다른 계수( $h, i, j$ )를 갖고 8x8 변환 행렬  $C_{8p}$ 는 7개의 다른 계수( $a, b, c, d, e, f, g$ )를 갖는다. 본 논문에서는 모든 재 정렬된 4x4 변환 행렬을 다루기 위한 PE 회로를 도출하였다.

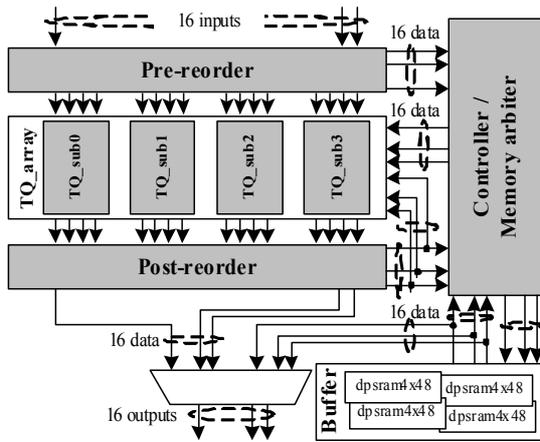
(그림 2)와 같이 제안된 회로는 Pre-reorder, TQ\_array, Post-reorder, Buffer, Controller/Memory arbiter 블록으로 구성된다. Pre-reorder 블록은 16 pixels/cycle의 데이터를 받고 4x4 DCT를 위해  $W_{4p}$ 와 8x8 DCT를 위해  $W_{00} \sim W_{11}$ 를 생성하는 순환 행렬 연산을 처리한다. 이 연산에서 데이터를 재 정렬하는 것은 단순한 동작이기 때문에 이 블록의 회로 크기는 무시할 만큼 작다. TQ\_array 블록은 재 정렬된 1D 4x4 DCT (4x4 DCT의  $D_{4p}W_{4p}$ , 8x8 DCT의  $D_{00}W_{00} \sim D_{11}W_{11}$ )를 수행한다. 양자화 회로는 TQ\_array 블록에 포함된다. TQ\_array 블록은 4개의 TQ\_array 단위 연산 블록 (TQ\_sub0~TQ\_sub3)으로 구성된다. 각 단위 연산 블록은 4 pixels/cycle을 처리하기 때문에 TQ\_array 블록은 16 pixels/cycle을 처리하게 된다. Post-reorder 블록은 TQ\_array 블록으로부터 16 pixels/cycle을 받고 Buffer 블록에 저장한다. Post-reorder 블록의 다른 역할은 4x4 DCT의  $Y_{4p}$ 와 8x8 DCT의  $Y_{8p}$ 를 생성하기 위해 순환 행렬의 동작을 처리하는 것이다. Buffer 블록은 (15)를 처리하기 위해 첫 번째 64개 중간 결과들 ( $D_{00}W_{00}$ ,  $D_{10}W_{00}$ ,  $D_{00}W_{01}$ ,  $D_{10}W_{01}$ )을 저장하고 두 번째 16개 중간 결과들 ( $D_{01}W_{10}$ ,  $D_{11}W_{10}$ ,  $D_{01}W_{11}$ ,  $D_{11}W_{11}$ )을 더한다. 이것은 또한 첫 번째 1D DCT 결과들을 저장하고 두 번째 1D DCT에 진치된 값들을 전달하기 위한 전치 버퍼 (transpose buffer)로 사용된다. Controller/Memory arbiter 블록은 4x4 또는 8x8 DCT 중에서 수행하고자 하는 DCT 종류를 선택하고 Buffer 블록의 SRAM들을 제어한다.

(그림 3)은 TQ\_array 단위 연산 블록의 구조를 나타낸 것이다. 각 단위 연산기는 8개 가산기, 7개 곱셈기, 7개 부호 변환기 (SC), 7개 멀티플렉서를 갖는다. DCT 계수들은 외부로부터 단위 연산기에 공급된다. 8x8 DCT의 경우 7개 계수들 ( $a \sim g$ )이 공급되고 4x4 DCT의 경우 3개의 계수들 ( $a = f = g = h$ ,  $b = e = i$ ,  $c = d = j$ )이 공급된다. (그림 4)의

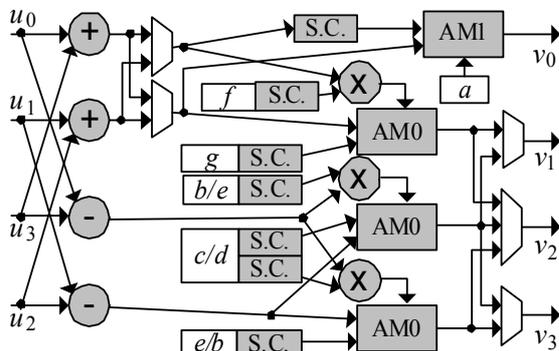
AM0와 AM1은 DCT와 양자화 동작의 기본 회로이다. T\_coeff (Q\_coeff)는 DCT (양자화) 동작을 위한 계수들이다. T\_din0과 T\_din1 (Q\_din0과 Q\_din1)는 DCT (양자화) 동작의 입력 데이터이다. DCT 동작은 T\_coeff, T\_din0 and T\_din1을 이용하여 처리된다. 양자화 동작은 Q\_din0 또는 Q\_din1과 Q\_coeff를 곱하여 처리된다.



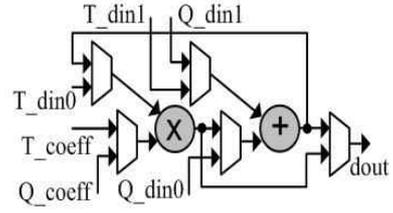
(그림 1)  $D_{4p}$ ,  $D_{00}$ ,  $D_{01}$ ,  $D_{10}$ ,  $D_{11}$ 의 신호 흐름 그래프



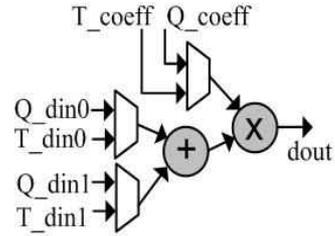
(그림 2) 제안된 DCT와 양자화의 회로 구조



(그림 3) TQ\_array 단위 연산기의 구조



(a) AM0



(b) AM1

(그림 4) AM0와 AM1 연산기의 구조

<표 1>은 1D 8x8 DCT와 양자화 동작을 처리하기 위한 TQ\_array 블록의 처리 단계를 나타낸다. 1D 8x8 DCT를 위해 총 13개 사이클이 필요하다. TQ\_array 블록의 입력들은  $W_{00}$ ,  $W_{01}$ ,  $W_{10}$ ,  $W_{11}$ 의 4개 행렬로 표현되고 여기에  $D_{00}$ ,  $D_{01}$ ,  $D_{10}$ ,  $D_{11}$ 의 4개 재 정렬 변환 행렬들이 곱해진다. TQ\_array 블록은 매 사이클 마다 16개 입력 데이터를 처리한다. <표 1>에서 보여진 것과 같이 6, 8, 10, 12 사이클에서 DCT 동작을 하는데 가산기만 사용된다. 이들 사이클 동안 곱셈기는 양자화 동작에 사용된다. 이와 같은 방법으로 추가적인 시간을 사용하지 않고 양자화 동작을 처리하였다.

<표 1> 1D 8x8 DCT와 양자화 동작을 위한 스케줄링

Cycle	1	2	3	4	5	6
DCT Inputs	$W_{00}$	$W_{00}$	$W_{01}$	$W_{01}$	$W_{10}$	$D_{00}W_{00}$
DCT Coeff.	$D_{00}$	$D_{10}$	$D_{00}$	$D_{10}$	$D_{01}$	
DCT Results		$D_{00}W_{00}$	$D_{10}W_{00}$	$D_{00}W_{01}$	$D_{10}W_{01}$	$D_{01}W_{10}$
Using	DCT	+ X	+ X	+ X	+ X	+
	Q					X

	7	8	9	10	11	12	13
	$W_{10}$	$D_{10}W_{00}$	$W_{11}$	$D_{00}W_{01}$	$W_{11}$	$D_{10}W_{01}$	
	$D_{11}$		$D_{01}$		$D_{11}$		
	$D_{00}W_{00}^+$ $D_{01}W_{10}$	$D_{11}W_{10}$	$D_{10}W_{00}^+$ $D_{11}W_{10}$	$D_{01}W_{11}$	$D_{00}W_{01}^+$ $D_{01}W_{11}$	$D_{11}W_{11}$	$D_{10}W_{01}^+$ $D_{11}W_{11}$
	+ X	+	+ X	+	+ X	+	
		X		X		X	

(+: adder, x: multiplier)

### 4. 구현 결과

제안된 회로는 Verilog HDL을 이용하여 RTL에서 구현되었고 Xilinx Virtex4 LX60 FPGA를 이용하여 동작을 검증하였다. 8x8 DCT는 MPEG-4 CODEC 회로에 적용하였고 4x4 DCT는 H.264 CODEC 회로에 적용하였다. 130nm 표준 셀 라이브러리를 이용하여 회로는 30,275 게이트로 합성되었고 최대 동작 주파수는 116.8MHz 이다. 8x8 영상 블록을 처리하는데 30 사이클 (데이터 입력과 재 정렬에 4 사이클, 첫 번째 1D DCT에 13 사이클, 두 번째 1D DCT와 양자화에 13 사이클)이 소요된다. 한 개 매크로블럭 (MB)은 6개 8x8 영상 블록을 포함하기 때문에 2D 8x8 DCT (4x4 DCT)와 양자화 하는데 MB 당 필요한 사이클 수는 <표 2>와 같이 180 (104)이다. 구현된 회로는 96 바이트, 4개 4x48-bit 이중 포트 SRAM을 사용한다. 논문 [1]은 MPEG-1/2/4, H.264의 4x4와 8x8 DCT만을 지원하고 논문 [7, 8]은 MPEG-1/2/4, H.264의 4x4, VC-1의 DCT만을 지원한다. 논문 [9]는 MPEG-1/2/4, H.264 4x4의 DCT와 양자화를 지원한다. 본 논문에서 제안된 회로는 MPEG-1/2/4, H.264 4x4와 8x8 그리고 VC-1 등과 같은 DCT와 양자화를 처리하며 <표 2>와 같이 기존 방법들에 비해 회로 크기와 사이클 수가 작은 회로로 구현되었다.

<표 2> 성능 비교

		[1]	[7]	[8]	[9]	제안 방법
사이클/ MB	4x4	n.a.	831	1,063	1,280	104
	8x8	n.a.	768	n.a.	1,280	180
#K 게이트		40	32	50	n.a.	30
최대 동작 주파수 (MHz)		50	136	80	54	117
SRAM (바이트)	Single-port	n.a.	0	512	n.a.	0
	Dual-port	n.a.	128	384	n.a.	96
지원 가능 여부 (DCT/양자화)	MPEG-1/2/4	o/x	o/x	o/x	o/o	o/o
	H.264 4x4	o/x	o/x	o/x	o/o	o/o
	H.264 8x8	o/x	x/x	x/x	x/x	o/o
	VC-1	x/x	o/x	o/x	x/x	o/o

### 5. 결론

본 논문에서는 통합 비디오 코덱을 위한 4x4/8x8 DCT와 양자화 회로의 고성능 구조를 제안한다. 4x4 DCT는 재 정렬과 분할된 4x4 DCT 행렬로 구성된 PE 회로를 통해 동작을 수행한다. 8x8 DCT는 PE 회로를 반복적으로 수행한다. DCT 계수들은 변환의 종류에 따라 외부로부터 공급된다. 변환에 필요한 중간 결과를 저장하고 첫 번째 1D DCT 결과를 전치하기 위해 작은 메모리가 사용되었다. 제안된 회

로의 곱셈기는 DCT와 양자화 동작에서 공유하였다. DCT와 양자화 동작의 클럭 사이클 또한 공유하였다. 회로 크기와 처리 시간 측면에서 DCT 회로에 양자화 회로를 추가하는데 필요한 자원은 무시할 만큼 작다. 제안된 회로는 DCT와 양자화를 기반으로 하는 차세대 비디오 코덱에서 사용될 것으로 기대된다.

### 참고 문헌

- [1] C. Huang, L. Chen, and Y. Lai, "A high-speed 2-D transform architecture with unique kernel for multi-standard video applications," IEEE ISCAS, pp.21-24, May, 2008.
- [2] CCITT Recommendation T.81, Digital Compression and Coding Continuous-Tone Still Images, 1992.
- [3] ISO/IEC 14496-2, Coding of Audio-Visual Objects - part 2: Visual, Nov., 1997.
- [4] Draft IUT-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Mar., 2003.
- [5] SMPTE, Standards for Television: VC-1 Compressed Video Bitstream Format and Decoding Process, SMPTE 421M-2006.
- [6] C.P. Fan, "Fast 2-dimensional 4x4 forward integer transform implementation for H.264/AVC," IEEE Trans. on Circuits and Systems II, Vol.53, pp.174-177, Mar., 2006.
- [7] S. Lee and K. Cho, "Architecture of transform circuit for video decoder supporting multiple standards," IET Electronics Letters, Vol.44, No.4, pp.274-275, Feb., 2008.
- [8] J.H. Park, S.H. Lee, S. Lim, J.H. Kim, and S. Kim, "A flexible transform processor architecture for multi-CODECs (JPEG, MPEG-2, 4 and H.264)," IEEE ISCAS, pp.5347-5350, May, 2006.
- [9] M. Hase, K. Akie, M. Nobori, and K. Matsumoto, "Development of low-power and real-time VC-1/H.264/MPEG-4 video processing hardware," ASPDAC, pp. 637-643, Jan., 2007.



이 선 영

e-mail : drleesy@kети.re.kr

1998년 한국외국어대학교 전자공학과 (학사)

2000년 한국외국어대학교 전자공학과 (공학석사)

2009년 한국외국어대학교 전자공학과 (공학박사)

2001년~2006년 (주)이시티 반도체설계연구소 선임연구원

2009년~현 재 전자부품연구원 융합신호SoC연구센터 선임연구원

관심분야: SoC 설계 등



## 조 경 순

e-mail : kscho@hufs.ac.kr

1982년 서울대학교 전자공학과(학사)

1984년 서울대학교 전자공학과(공학석사)

1988년 미국 Carnegie Mellon University  
대학원 전기 및 컴퓨터 공학과(공학박사)

1988년~1994년 삼성전자(주) 반도체 총괄 선임, 수석 연구원

1994년~현재 한국외국어대학교 전자공학과 조교수, 부교수,  
정교수

관심분야: SoC 설계 등