

FPGA를 리용한 H.264용근수변환 및 량자화변환모듈 실행방법

류현성, 남철만

컴퓨터망이나 텔레비존방영에서는 QCIF(177×144pixel), CIF(352×288pixel), SD(720×480 pixel)와 같은 비디오봉사를 진행하는데 H.264표준[3]을 리용하고있다. 그러나 H.264표준[1, 2]을 장치적으로 실현하는것은 매우 어려운 문제이다.

론문에서는 FPGA를 리용하여 H.264부호화구조에서 용근수변환과 량자화, 역용근수 변환과 역량자화구조로 이루어진 변환부호화모듈을 장치적으로 실현하는 방법을 제기 하였다.

1. 변환알고리즘과 장치구성방식

1) 변환알고리즘

(1) 용근수변환알고리즘

H.264에서 밝기에측우차신호 Y 는 4×4용근수코시누스변환(ICT)에 의해 주파수영역 z 로 전환된다. 이때 4×4입력행렬 X 에 대해 변환기의 출력행렬을 다음과 같이 쓸수 있다.

$$Y = (C_f \cdot X \cdot C_f^T) \otimes E_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & -2 \\ 1 & -2 & 2 & -1 \end{bmatrix} \cdot X \cdot \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \quad (1)$$

여기서 $a = \frac{1}{2}$, $b = \sqrt{1/2} \cos(\pi/8)$ 이다.

변환행렬 E_f 를 량자화처리하면 용근수변환부분이 얻어진다. $C_f X C_f$ 는 4개의 결수들인 1, -1, 2, -2를 리용하는 2차원용근수변환의 핵심부분으로서 더하기와 밀기연산만을 포함한다.

H.264에서는 또한 16×16내부에측방식에서 밝기 Y 의 4×4 DC성분들과 색도 Cr, Cb의 DC성분들에 대해 아다마르변환(HT)을 리용하여 다시 변환을 진행한다.

$$Y_{LumaDC} = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \cdot X_{LumaDC} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \right) \cdot 2^{-1} \quad (2)$$

$$Y_{LumaDC} = \left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot X_{chraDC} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right) \quad (3)$$

(2) 량자화알고리즘

H.264에서 밝기신호에 대한 량자화공식은 다음과 같다.

$$Z_{ij} = \text{round}\left(Y_{ij} \frac{PF}{Qstep}\right) \quad (4)$$

여기서 Y_{ij} 는 옹근수변환후의 출력신호이며 $Qstep$ 는 량자화파라미터 QP 에 따라 결정되는 량자화결음으로서 52개의 값을 가진다.

옹근수변환에서 후처리행렬 E_f 를 량자화단계에 통합시키면 식 (4)를 다음과 같이 쓸 수 있다.

$$Z_{ij} = \text{round}\left(W_{ij} \times \frac{POF}{Qstep}\right) = \text{round}\left(W_{ij} \times \frac{LevelScale}{2^q}\right) \quad (5)$$

$$q = 15 + \text{floor}\left(\frac{QP}{6}\right) \quad (6)$$

량자화에서 곱하기인수 $LevelScale$ 의 값은 표 1과 같다.

표 1. 량자화에서 곱하기인수 $LevelScale$ 의 값			
QP	위치 1		기타 위치
	(0, 0)(2, 0)(0, 2)(2, 2)	(1, 1)(1, 3)(3, 1)(3, 3)	
0	13 107	5 243	8 066
1	11 916	4 660	7 490
2	10 082	4 194	6 554
3	9 362	3 647	5 825
4	8 192	3 355	5 243
5	7 282	2 893	4 559

이 값을 고려하여 식 (5)를 간단히 하면

$$Z_{ij} = \left|W_{ij}\right| \times LevelScale(QP \% 6, (i, j)) + f \gg q \text{ (bit)} \quad (7)$$

$$\text{sign}(Z_{ij}) = \text{sign}(W_{ij})$$

이다.

DC성분들에 대해 량자화과정은 다음과 같다.

$$\left|Z_{DC_{ij}}\right| = \left|W_{DC_{ij}}\right| \times LevelScale(QP \% 6, (0, 0)) + 2f \gg (q+1) \quad (8)$$

$$\text{sign}(Z_{DC_{ij}}) = \text{sign}(W_{DC_{ij}})$$

① 역량자화알고리즘

역량자화식을 쓰면 다음과 같다.

$$Y'_{ij} = Z_{ij} \times Qstep \times PRF \times 64 \quad (9)$$

여기서 $Qstep \times PRF \times 64$ 를 $InvLevelScale$ 로 넘기면 식 (9)를 다음과 같이 쓸 수 있다.

$$Y'_{ij} = Z_{ij} \times InvLevelScale(QP \% 6, (i, j)) \ll Qp / 6 \quad (10)$$

표 2에 역량자화에서 곱하기인수 $InvLevelScale$ 의 값을 주었다.

4×4 DC블록에 대해 량자화식은 다음과 같다.

$$Y'_{DC_{ij}} = (Z_{DC_{ij}} \times InvLevelScale(QP\%6, (0, 0))) << \left(\frac{QP}{6} - 2 \right), QP \geq 12 \quad (11)$$

표 2. 역량자화에서 곱하기인수 $InvLevelScale$ 의 값

QP	위치 1	위치 2	기타 위치
	(0, 0)(2, 0)(2, 2)(0, 2)	(1, 1)(1, 3)(3, 1)(3, 3)	
0	10	16	13
1	11	18	14
2	13	20	16
3	14	23	18
4	16	25	20
5	18	29	23

$$Y'_{DC_{ij}} = (Z_{DC_{ij}} \times InvLevelScale(QP\%6, (0, 0))) + 2^{1-(QP/6)} >> \left(2 - \frac{QP}{6} \right), QP \leq 12 \quad (12)$$

2×2 DC블록에 대해 량자화식은 다음과 같다.

$$Y'_{DC_{ij}} = (Z_{DC_{ij}} \times InvLevelScale(QP\%6, (0, 0))) << \left(\frac{QP}{6} - 1 \right), QP \geq 6 \quad (13)$$

$$Y'_{DC_{ij}} = (Z_{DC_{ij}} \times InvLevelScale(QP\%6, (0, 0))) >> 1, QP \leq 6 \quad (14)$$

② 역용근수변환알고리즘

밝기신호에 대한 역용근수변환의 출력식은 다음과 같다.

$$X' = (C_i^T (Y \otimes E_i) C_i) = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \times \left(Y \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix} \quad (15)$$

그리고 식 (15)의 DC성분들에 대한 역변환식은 꼭 반대이다.

$$X'_{LumaDC} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times Y'_{LumaDC} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (16)$$

$$X'_{chroma} = \left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right) \times Y'_{chromaDC} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (17)$$

2) 장치구성방식

(1) 용근수변환 및 역용근수변환을 위한 장치구성

그림 1에 용근수변환기의 장치구성방식을 보여주었다.

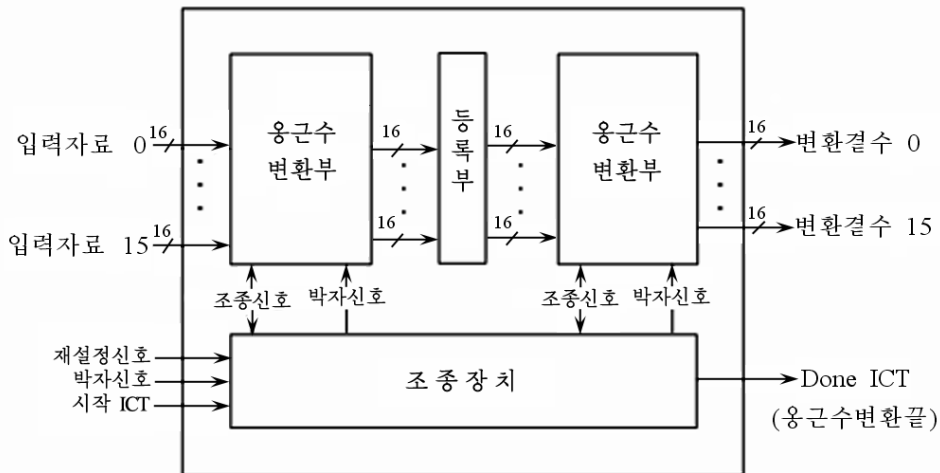


그림 1. 용근수변환기의 장치구성방식

입력자료는 화상에측값(0~255)과 원화상값(0~255)의 차로 주어지는 4×4블록의 잔류오차들로서 한주기동안에 2개의 1차원 행, 열변환기들에서 동시에 처리되어 16개의 변환결수들이 출력된다.

(2) 량자화 및 역량자화를 위한 장치구성

그림 2에 4×4 AC결수들과 DC결수들에 대한 량자화장치구성방식을 보여주었다.

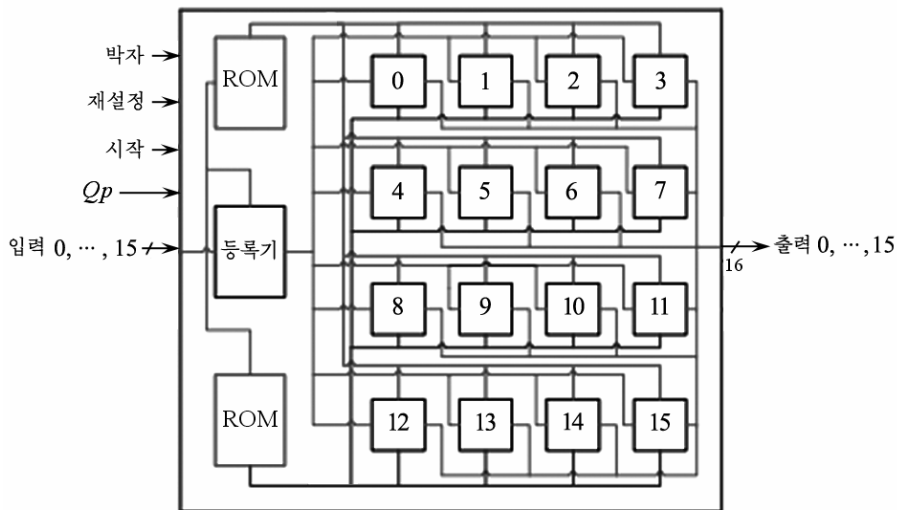


그림 2. 4×4 AC결수들과 DC결수들에 대한 량자화장치구성방식

H.264용근수변환기에서 출력된 신호는 다음 량자화모듈에 입력되어 처리된다.

량자화모듈은 16개의 처리단위(PE)들과 16개의 자료를 입력하는 입력모선, 입력값을 저장하기 위한 등록기렬, 2개의 읽기전용기억기(ROM)들로 구성된다. 량자화모듈에서는 16개의 입력자료를 받아 QP 인수에 따라 4개 박자주기동안에 량자화를 처리한다.

처리단위(PE)는 ROM-MF, 곱하기, 더하기, 밀기등록기, 조종단으로 구성되었다. 옹근수 곱하기장치는 AC와 DC변환결수들과 *LevelScale*인수의 곱하기를 실현하며 그 자료는 ROM-MF 기억기에 보관되었다가 $QP/6$ 의 나머지에 따라 선택된다. 가산기는 곱한 결과와 ROM-F기억기에서 주어진 f 파라미터를 더한다. 밀기등록기는 가산기의 결과를 $qbit$ 만큼 밀기한다. 곱하기, 더하기, 밀기장치들과 ROM-MF기억기모듈들은 각각 1개의 박자주기를 차지한다.

역량자화모듈은 량자화모듈과 류사하다. 차이를 보면 PE모듈에서 가산기가 없을뿐이다. 그러나 DC성분에 대한 역량자화는 꼭같고 밀기등록기가 $QP < 12$ 일 때 진행되는것이다. 역량자화모듈은 3~4주기마다 16개의 결수들을 출력시킨다.

2. 실험 결과

QuatosII 16 standards환경에서 Cyclone 5계렬 FPGA(5csx6c6d6f31c6)로써 장치적구성을 실현하였다. 제안방법을 실현하는데 필요한 FPGA장치자원을 표 3에 보여주었다.

표 3. FPGA장치자원

항 목	지 표
Family	Cyclone V
Device	5csx6c6d6f31c6
Timing Model	Final
Logic Utilization(in ALMs)	3 777/41 910(9%)
Register	146
Total DSP Blocks	34/112(30%)

변환모듈장치는 6박자주기동안에 1개의 마크로블록에 대한 변환을 실현하며 이것은 장치소편의 박자주파수가 156MHz일 때 HD(1 920×1 080)비디오를 실시간적으로 부호화하는데 적합하다는것을 보여준다. 또한 장치자원소비도 아주 적기때문에 운동예측을 비롯하여 H.264부호화를 실현하는데 영향을 미치지 않는다.

맺 는 말

제안된 방법은 QuatosII 16 standards환경에서 Cyclone 5계렬 FPGA소편구조로써 실현되었으며 실험결과는 HD(1 920×1 080pixel)비디오를 압축하는데 효과적이라는것을 보여주었다.

참 고 문 헌

- [1] A. B. Atitallah et al.; International J. of Computer Science, Engineering and Applications, 1, 5, 119, 2011.
- [2] S. Hamdy et al.; Turkish Journal of Electrical Engineering & Computer Sciences, 24, 1967, 2016.
- [3] S. L. Youn-Long et al.; VLSI Design for Video Coding, Springer, 52~71, 2010.

A Method for Implementation of Integer and Quantization Transform Modules in H.264 by Using FPGA

Ryu Hyon Song, Nam Chol Man

In this paper we proposed a method for implementation of integer and quantization transform modules which became as core in H.264 encoder by using FPGA

Key words: integer transform, quantization transform