

논문 2012-07-22

리프팅 기반의 고속 정수 웨이블릿 변환의 효율적인 구현 구조

(An Efficient Implementation Architecture for Lifting Based
High Speed Integer Wavelet Transform)

김 석 준, 장 영 조*

(Suc June Kim, Young Jo Jang)

Abstract : In this paper, we propose an efficient architecture for 2D IWT using an existing 1D IWT. Lifting based IWT is the architecture of which a multiplier is replaced by adders and shift registers. The structure is relatively simple and modular. The proposed architecture to process an image size with 256x256 pixels consists of 16 adders, 8 shift registers, and some memories. By processing two rows at the same time, 2D sub-band coefficients can be calculated immediately after 1D sub-band coefficients have been processed. The architecture is designed so that each image can be inputted consecutively. The number of adders and shift registers is increased by twice comparing the existing architecture, but the memory size and the execution time are decreased by half. The proposed architecture is implemented using Verilog-HDL and simulated using iSim. It is synthesized and demonstrated at ISE for xc5v1x330 in RPS3K board.

Keywords : Lifting based IWT, Wavelet transform, Hardware architecture

1. 서론

최근 이동전화 단말기를 중심으로 휴대용 디지털 기기의 큰 발전이 이루어졌다. 이러한 발전에 더불어 휴대용 기기를 사용하는 사용자 역시 동영상이나 비디오 서비스와 같은 대용량 정보와 이를 적절하게 제공할 수 있는 질적으로 향상된 서비스를 요구하고 있다. 이와 같은 사용자의 요구에 맞춰 대용량 데이터를 효율적으로 처리하기 위해 영상 압축에 관련된 여러 가지 표준이 제정되어 있다. 이러한 표준으로는 광범위하게 사용되고 있는 JPEG (Joint Photographic Experts Group), MPEG (Moving Picture Experts Group) 및 H.26x가 있다.

* 교신저자(Corresponding Author)

논문접수 : 2012.01.27., 수정일 : 2012.02.25.,

채택확정 : 2012.04.09.

김석준, 장영조 : 한국기술교육대학교

JPEG과 MPEG은 DCT (Discrete Cosine Transform)를 기반으로 하고 있다. 그러나 DCT는 블록 효과라는 극복할 수 없는 단점을 갖고 있다. 이를 보완하기 위한 많은 연구가 활발히 진행되고 있다.

이중 가장 주목받고 있는 기술은 웨이블릿 (wavelet) 기반의 영상처리 기술로써 DWT (Discrete Wavelet Transform)는 블록 효과를 제거할 수 있을 뿐 아니라 전체 영상을 대상으로 인간의 시각에 따른 처리가 가능하다. 이러한 특성 때문에 DWT는 DCT를 대신하여 JPEG2000의 주파수 변환 기술로 지정되었다 [1].

DWT에 대한 전통적인 구현 방식은 FIR (Finite Impulse Response) 필터 뱅크(filter bank)를 사용하거나 컨볼루션 기반의 방법이 사용된다. 이러한 구현 방식은 영상처리에 요구되는 고속 처리나 저전력의 하드웨어 구현을 위한 면적 면에서 적합하지 않은 방식이다.

Knowles를 시작으로 Daubechies와 Sweldens 등이 삼각 행렬과 밴드 행렬 곱셈을 사용하여 푸리

에 변환(Fourier Transform)에 의존하지 않는 리프팅(lifting) 기반의 웨이블릿 변환을 제안하였으며 [2-4], 이를 바탕으로 VLSI 하드웨어 구현을 위한 연구가 활발하게 진행되고 있다 [5-8]. 이러한 리프팅 기반의 구현은 FIR 필터 뱅크나 컨볼루션을 기반으로 하는 구현 방식보다 우수한 성능 향상을 보였으며, 실시간 처리를 해야 하는 모바일 분야에도 적용 가능하다. 그러나 하드웨어 구현을 위하여서는 비교적 큰 메모리와 큰 처리 시간을 요구한다.

본 논문에서는 기존에 제안된 1차원 IWT(Integer Wavelet Transform) [9]를 사용하여 2차원 변환기를 설계한다. 이 1차원 IWT는 곱셈기를 시프트 레지스터로 대체하여 (5, 3) 필터를 설계할 수 있으며 정수 연산을 사용하기 때문에 구조가 비교적 간단하고 규칙적이며 무손실 변환의 특성을 가진다.

1차원 변환기를 2차원 변환에 적용하기 위해서는 행 방향으로 1차원 변환을 적용하고, 그 결과 값에 대해 열 방향으로 두 번째 1차원 변환을 적용한다. 그러나 1차원 변환기를 구조 변경없이 2차원 변환에 적용하면, NxM 크기의 영상은 세로 방향의 변환을 위해 NxM 크기의 메모리가 필요하고 실행 시간은 1.5xNxM개 정도의 클럭이 필요하다. 기존에 제안된 2차원 변환기는 샘플을 순차적으로 입력받아 1차원 변환이 완료된 값을 메모리에 저장하기 때문에 영상의 크기가 증가하면 이에 필요한 메모리의 크기도 증가한다 [10].

본 논문에서 제안하는 변환기는 기존의 1차원 변환기의 구조는 유지하면서 N/2 크기의 메모리를 사용하여 행과 열 방향 연산을 동시에 수행할 수 있게 한다. 그 결과 2xN 크기의 메모리를 사용하고 실행시간은 NxM/4개 정도로 감소된 클럭 수가 필요하게 된다. 제안한 변환기는 입력 영상의 크기와 상관없이 입력, 덧셈기 그리고 시프트 레지스터의 개수가 2배 증가하지만, 영상의 열 개수만큼의 메모리를 사용하고 1차원 변환 값을 저장하지 않고 연산에 필요한 계수만을 저장하기 때문에 영상의 전체 크기에 영향을 받지 않는다. 이를 통해 실행 클럭 수와 메모리 사용량은 기존 변환기에 비해 약 2배 정도 감소한다.

제안하는 변환기는 Verilog HDL로 설계하고 Xilinx사의 xc5v1x330 FPGA에 구현하여 그 동작을 확인하였다. 실험에 사용된 영상은 8 비트의 15x15 크기인 정지 영상과 48 프레임(frame)으로 구성된 CIF(Common Intermediate Format) 동영상이다.

2장에서는 제안하는 변환기의 구조에 대한 설계와 구현 및 검증에 대해 설명한다. 3장에서는 결론과 향후 연구 과제를 제시한다.

II. 본 론

1. 제안하는 2차원 IWT 변환기의 구조

그림 1은 제안하는 순방향 2차원 IWT 변환기의 블록 다이어그램이다. 제안하는 변환기는 두 개의 행 방향 처리기(Row Processor, RP)와 두 개의 열 방향 처리기(Column Processor, CP)로 구성된다.

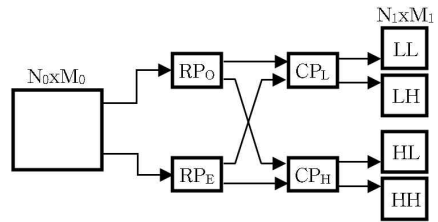


그림 1. 제안하는 구조
Fig. 1. Proposed architecture

원본 영상을 $N_0 \times M_0$ (열x행), 2차원 변환이 완료된 영상을 $N_1 \times M_1$ 이라 할 때, 2차원 영상을 처리하는 방법은 다음과 같다. 두 개의 행 방향 처리기(RP₀와 RP_E)는 네 개의 값(L_{odd}, H_{odd}와 L_{even}, H_{even})을 출력한다. CP_L은 L_{odd}와 L_{even}을 사용하여 LL과 LH를 출력한다. CP_H는 H_{odd}와 H_{even}을 사용하여 HL과 HH를 출력한다.

이러한 구조를 사용하면 RP에서 처리한 값이 출력됨과 동시에 CP의 입력으로 사용될 수 있기 때문에 기존 변환기를 사용하여 2차원 변환을 할 때에 비해 변환 시간을 약 2배 단축할 수 있게 된다.

1.1 행 방향 처리기

영상에 행 방향으로 1차원 변환을 적용하기 위해 각 H(high) 및 L(low) 패스 웨이블릿 계수 $d_1(i, j)$ 와 $a_1(i, j)$ 는 식 (1)을 사용하여 구한다.

$$d_1(i, j) = a_0^{odd}(i, j) - \frac{1}{2}(a_0^{even}(i-1, j) + a_0^{even}(i, j))$$

$$a_1(i, j) = a_0^{even}(i-1, j) + \frac{1}{4}(d_1(i, j) + d_1(i-1, j)) \quad (1)$$

i, j 는 열과 행을 의미한다. $i = 0, 1, 2, \dots, N/2-1$ 이고, $j = 0, 1, 2, \dots, M-1$ 이다. a_0^{odd} 와 a_0^{even} 은 입력을

홀수와 짝수 샘플로 나누는 것을 의미한다.

식 (1)을 덧셈기와 시프트 레지스터를 조합하여 그림 2와 같이 구성할 수 있다. 이 구조는 Al-Sulaifanie 등 [9]이 제안한 변환기와 동일하지만 제어 신호를 제거하고 L과 H 처리기를 각각 구성하고, 두 개의 행 방향 처리기를 사용한다.

reg는 D-F/F를 사용하여 이전 입력을 저장하는 역할을 하며, 입력된 신호가 지연 없이 출력 신호로 반영되는 래치(latch)를 사용하여 행 방향 변환을 한 클록에 수행할 수 있도록 한다. in1과 in2는 M 번째 행의 홀수와 짝수 번째 입력 샘플이다. out1과 out2는 행 방향 변환이 완료된 계수 값으로 각각 $L(a_1)$ 과 $H(d_1)$ 이다.

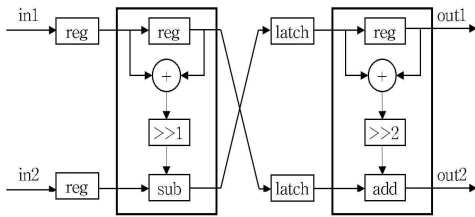


그림 2. 행 방향 처리기
Fig. 2. Row processor

1.2 열 방향 처리기

열 방향 변환은 행 방향 변환과 유사하게 이전에 변환된 계수가 필요하다. 이것은 식 (2)와 같이 표현할 수 있다.

$$d_2(i, j) = a_1^{odd}(i, j) - \frac{1}{2}(a_1^{even}(i, j-1) + a_1^{even}(i, j))$$

$$a_2(i, j) = a_1^{even}(i, j-1) + \frac{1}{4}(d_2(i, j) + d_2(i, j-1)) \quad (2)$$

식 (2)의 i, j 는 각각 열과 행을 의미한다. $i=0, 1, 2, \dots, N/2-1$ 이고, $j=0, 1, 2, \dots, M/2-1$ 이다. 그리고 a_1^{odd}, a_1^{even} 은 행 방향 처리기에서 출력된 계수를 홀수와 짝수로 나누는 것을 의미한다.

열 방향 처리기의 구조는 행 방향 처리기와 유사하다. 그러나 제안하는 변환기의 입력은 행 방향으로 입력되고, 행 방향 변환이 완료된 직후 연속으로 열 방향 변환이 진행된다. 이로 인해 열 방향 변환기는 행 방향 변환기에서 출력된 계수를 저장하는 메모리가 필요하다. 이러한 계수를 저장하기 위해서는 $N_0/2$ 크기의 메모리 두 개가 필요하다.

그림 3은 열 방향 처리기의 구조를 보여준다.

MEMO는 행 방향 변환이 완료된 짝수 행의 L 또는 H를 저장한다. MEM1은 열 방향 변환이 완료된 LH 또는 HH를 저장한다. 이렇게 저장된 값들은 다음 행 변환에 사용된다. in1과 in2는 행 방향 변환이 완료된 L 또는 H이다. out1과 out2는 L 변환일 경우 LL과 LH이고, H 변환일 경우 HL과 HH이다.

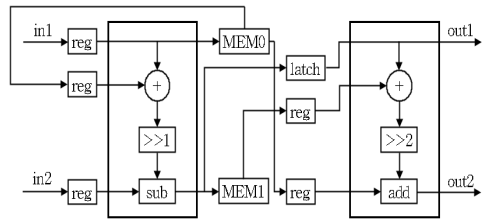


그림 3. 열 방향 처리기
Fig. 3. Column processor

1.3 제어기

그림 4는 2차원 IWT 변환기의 제어기의 블록도이다. 열 방향 처리기는 특별한 제어 신호 없이 동작 가능하다. 그러나 열 방향 처리기는 이전 값을 저장하는 메모리를 포함하므로 제어 신호가 필요하다.

각 신호의 역할은 다음과 같다. start 신호는 2차원 변환의 시작을 알리는 신호이다. done 신호는 2차원 변환기와 기타 모듈을 제어하는 외부 제어 모듈에서 출력되는 신호를 입력받는다. 이 신호는 양방향 2차원 변환이 완료되었음을 의미하며, 이 신호로 인해 출력 신호인 rd_en, rd_addr, wr_en 및 wr_addr 신호를 초기화한다. rd_en과 rd_addr은 메모리 읽기 신호와 주소로 열 방향 처리기의 연산에 적절한 값을 제공하기 위해 사용된다. wr_en과 wr_addr은 그림 3의 중간 출력 값을 메모리에 저장하기 위한 신호로 사용한다.

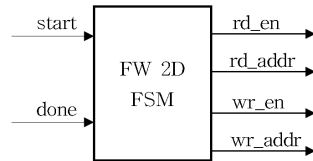


그림 4. 제어기
Fig. 4. Controller

2. 구현 및 동작실험

제안한 2차원 변환기의 구조를 Verilog HDL로 설계하고 FPGA에 구현하여 8 비트의 15x15의 픽

셀 크기로 된 정지영상과 CIF 동영상을 사용하여 동작을 검증한다. Xilinx의 xc5vlx330 FPGA가 탑재된 Huins 사의 RPS3K 장비에서 CIF 영상 48 프레임(약 1.5초)을 실시간으로 변환하고 동작하는 실험 결과를 설명한다.

2.1 15x15 영상

그림 5는 제안하는 구조를 사용하여 변환할 3장의 영상을 보여준다. 이 영상의 RGB 색상 중 8 비트의 R 성분을 사용한다. 또한, 3장의 영상들을 사용하여 연속으로 변환할 수 있음을 시뮬레이션을 통하여 확인한다.

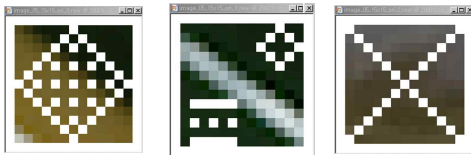
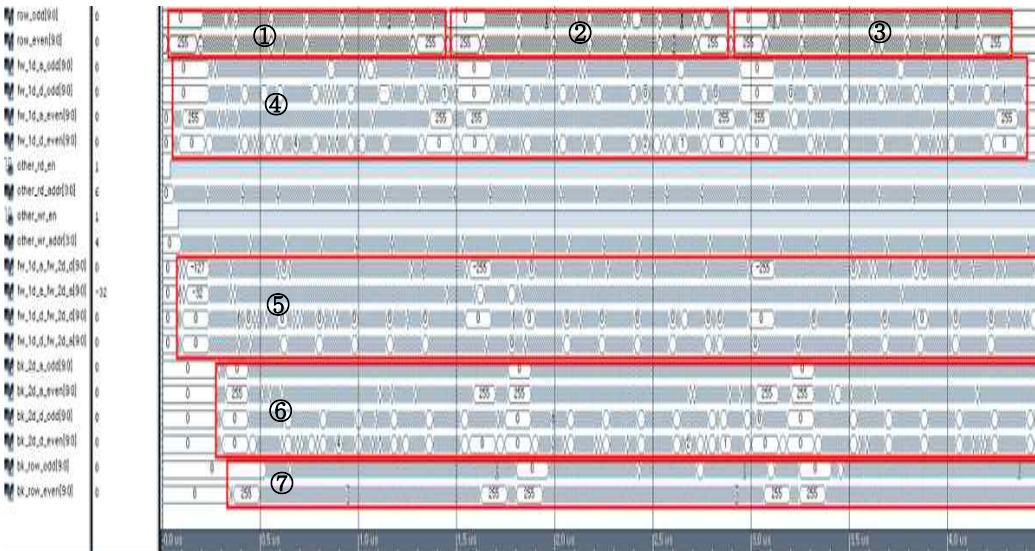


그림 5. 원본 입력 영상
Fig. 5. Original input images

그림 6은 iSim을 사용한 시뮬레이션 파형이다. 파형에서 2차원 변환기는 2개의 클록 지연이 있고, 한 개의 행을 변환하려면 10개의 클록이 필요하다. 두 개의 행을 동시에 변환하므로 15x15 영상은 총



- ① ~ ③ : 3장의 연속된 입력 영상
- ④ : 순방향 1차원(행) 변환
- ⑤ : 순방향 2차원(열) 변환
- ⑥ : 역방향 2차원(열) 변환
- ⑦ : 역방향 1차원(행) 변환

그림 6. 시뮬레이션 파형
Fig. 6. Simulation Wave

8번의 행 변환이 필요하다. 따라서 한 장의 영상을 변환하는 데 필요한 클록 수는 82개이고 3장의 영상을 연속으로 변환하면 총 242개의 클록이 필요하다. 이때 메모리는 40x10 비트가 필요하다.

그림 7은 RPS3K를 사용하여 검증한 결과이다. 그림 7의 윗 줄 영상들은 원본 입력 영상들이고, 중간 줄의 영상들은 양 방향 변환이 완료된 영상들이다. 아래 줄 영상들은 각 원본 영상과 변환이 완료된 영상의 차이를 보여주고 있는데 원본 영상과 변환된 영상의 차이가 없음을 알 수 있다.

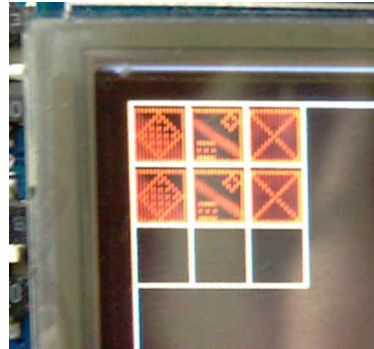


그림 7. 15x15 3개 영상의 실험 결과
Fig. 7. Experimental result for three 15x15 images

2.2 352x240 (CIF) 영상

4:4:4로 샘플링 된 YCbCr의 각 성분이 8 비트인 CIF 영상 48 프레임에 대한 IWT 변환을 위해 RPS3K 장비의 외부 SDRAM에 영상을 저장한다. 그림 8은 CIF 영상을 변환하기 위한 전체 블록도이다. 외부 SDRAM에 저장된 YCbCr 영상을 한 프레임씩 FPGA의 내부 메모리에 저장하고, 저장된 영상을 제안하는 2차원 변환기의 입력으로 사용한다. 순방향 변환기의 출력은 역방향 변환기의 입력으로 사용하고, 복원된 영상은 메모리와 RGB 변환기를 거쳐 TFT-LCD로 출력한다.

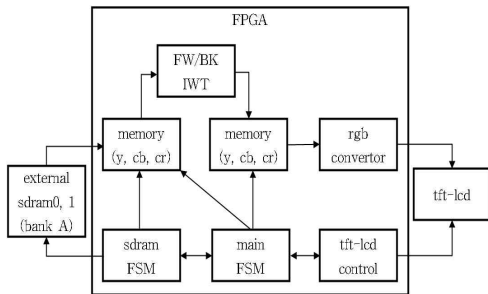


그림 8. CIF 영상을 처리하기 위한 블록도
Fig. 8. Block diagram to process CIF images

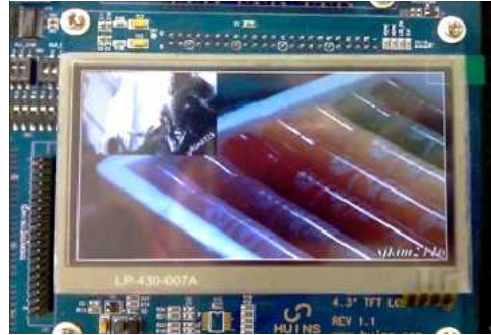
그림 9 (a)는 CIF 동영상의 순방향 2차원 변환이 완료된 동영상상을 보여주며, 그림 9 (b)는 역방향 2차원 변환이 완료된 동영상상을 보여준다. 1 프레임의 CIF 영상을 변환하기 위해서는 21,540개의 클럭이 필요하고, 712x10 비트의 메모리가 필요하다. 실험의 동작 확인을 통하여 제안하는 변환기는 실시간 응용에 사용할 수 있고, 입력 영상의 손실 없이 복원되고 있음을 알 수 있다.

3. 성능분석 및 비교

표 1은 8비트 256x256 크기의 영상에 대해, 기존의 2차원 변환기를 사용한 경우(①) [10], Al-Sulaifanie 등이 제안한 1차원 변환기를 사용하여 2차원 변환기를 사용한 경우(②) [9], 그리고 본 논문에서 제안하는 변환기를 사용한 경우(③)의 하드웨어 사용과 실행 클럭 수에 대한 성능을 비교한 것이다.

제안하는 구조는 기존의 변환기(①)보다 약 두 배의 덧셈기와 시프트 레지스터를 사용한다. 그러나 메모리 사용량은 반으로 감소하여 하드웨어의 구현에 필요한 면적은 유사한 비율을 유지함에 비하여

변환에 필요한 실행 클럭수가 반으로 감소하여 동작속도의 성능이 향상됨을 알 수 있다.



(a) 순방향 2차원 변환
(a) Forward 2D transform



(b) 역방향 2차원 변환
(b) Backward 2D transform
그림 9. CIF 영상의 실험 결과

Fig. 9. Experimental result for CIF images

표 1. 성능 비교

Table 1. Performance Comparisons

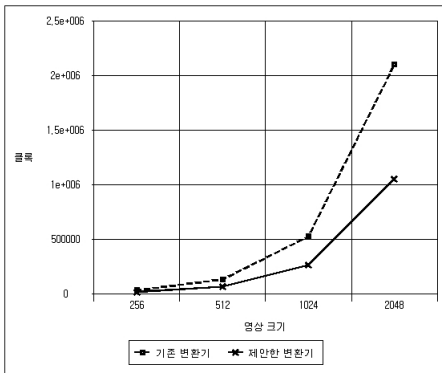
	①	②	③
덧셈기	8	12	16
시프트 레지스터	4	6	8
레지스터	18	18	22
메모리	1,024	66,560	520
실행 클럭 수	33,290	50,184	16,772

표 2는 제안하는 변환기를 사용하여 다양한 크기의 동영상에 실시간 변환을 적용할 수 있음을 확인한 것으로 각 영상의 크기에 따른 실행 클럭과 메모리 사용량을 보여준다. 4:4:4로 샘플링된 YCbCr이 각 8비트인 CIF 동영상 30 프레임을 변환할 때 사용된 메인 클럭의 동작 주파수가 100MHz이라면 약 6.5ms 정도의 변환 시간이 소요

된다. HD급 1,920x1,080의 영상 크기인 동영상에 대하여 30 프레임을 변환하면 약 156ms 정도가 필요하고, 메모리 사용량은 영상 열 크기(1,080)의 두 배 정도를 사용한다.

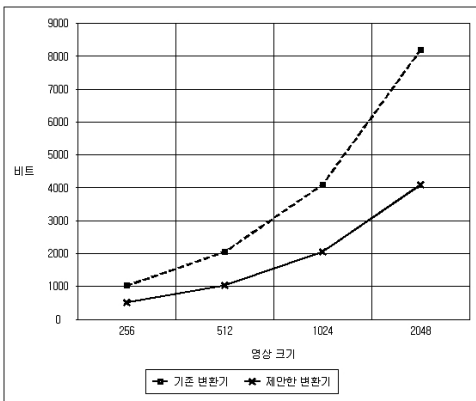
표 2. 동영상 크기에 대한 성능
Table 2. Performance for image size

영상크기		클럭 (30장)	시간(ms) @100MHz	메모리 (10bit)
열	행			
1,920	1,080	15,613,320	156.1	3,848
1,280	720	6,952,920	69.5	2,568
720	576	3,138,600	31.4	1,448
704	576	3,069,240	30.7	1,416
640	480	2,328,120	23.3	1,288
352	240	646,200	6.5	712
176	120	164,760	1.6	360



(a) 실행 클럭 수

(a) Number of execution clock



(b) 메모리 사용량

(b) Memory size

그림 10. NxN 영상에 대한 비교

Fig. 10. Comparisons for NxN image

그림 10은 행과 열의 크기가 같은 NxN 영상에 대해 기존 변환기(①)와 제안하는 변환기의 실행 클럭과 메모리 사용량에 대한 비교를 보여준다. 제안하는 변환기의 실행 클럭과 메모리 사용량은 영상 크기와 상관없이 기존 변환기보다 두 배의 성능이 향상됨을 예측할 수 있으며, 이를 통해 제안하는 구조를 사용하여 다양한 크기의 영상을 실시간 2차원 변환이 가능함을 알 수 있다.

III. 결론

본 논문에서는 덧셈기와 시프트 레지스터로 구성된 2차원 IWT 변환기를 설계하였다. 설계된 2차원 변환기는 실시간 영상 압축과 복원에 사용될 수 있다. 또한 주파수 변환을 사용하는 위터마킹 기법이나 추가적인 모듈과 함께 JPEG2000의 구현에 적용할 수 있다. 실시간 처리와 대용량 데이터를 요구하는 감시 및 보안과 관련된 CCTV와 모바일 동영상 처리 분야에서도 사용될 수 있다.

제안하는 구조를 구현하기 위해 Verilog HDL로 설계하고 Xilinx의 iSim을 사용하여 동작을 시뮬레이션하였다. Xilinx의 xc5v1x330을 탑재하고 있는 Huins사의 RPS3K 장비를 사용하여 실시간 동작을 검증하였다. 2개의 행을 동시에 처리하여 1차원 변환 값이 출력된 직후 2차원 변환이 이루어지고 각 영상은 연속으로 입력될 수 있도록 설계하였다.

FPGA의 동작 주파수를 100MHz로 하고 YCbCr이 4:4:4로 샘플링된 CIF 30 프레임을 변환할 때 순방향 2차원 변환에는 총 646,200개의 클럭이 필요하며, 이를 시간으로 환산하면 약 6.5ms 정도이다. 연산에 필요한 값을 저장하기 위한 메모리는 712x10 비트가 필요하다. 기존에 제안된 2차원 변환기와 비교하면 영상의 크기와 상관없이 입력, 덧셈기 그리고 시프트 레지스터의 개수는 두 배 증가하지만, 실행 클럭과 메모리 사용량은 반으로 감소한다. 제안하는 구조의 IWT를 적용하면 모바일 동영상과 같은 실시간 영상 처리가 충분히 가능할 것으로 예측된다.

본 논문에서 제안하는 구조는 고정 크기의 영상만을 변환할 수 있는 구조이다. 향후 다양한 크기의 영상에 적용할 수 있는 변환기를 설계하여, 여러 크기의 영상을 추가적인 구조의 변경없이 사용할 수 있는 변환기의 연구를 진행하고자 한다. 또한, 제안하는 구조는 유지하면서 (5, 3) 필터 이외의 다양한 필터도 지원하는 연구를 진행할 것이다.

참 고 문 헌

[1] M. Boliek, "JPEG 2000 Part I Final Draft International Standard," ISO/IEC JTC1/SC29 WG1, 2000.

[2] G. Knowles, "VLSI Architecture for the Discrete Wavelet Transform," IEEE Electronic Letters, Vol. 26, No. 15, pp.1184-1185, 1990.

[3] I. Daubechies, W. Sweldens, "Factoring wavelet transforms into lifting steps," Preprint, Bell Lab. 1996.

[4] A.R. Calderbank, I. Daubechies, W. Sweldens, B. Yeo, "Wavelet transforms that map integers to integers," Technical Reports, Depart. of Mathematics, Princeton University, 1996.

[5] C. Hunag, P. Tseng, L. Chen, "Efficient VLSI architectures of lifting-based discrete wavelet transform by systematic design method," Proceedings on IEEE International Symposium Circuits and Systems, Vol. 5, pp.565 - 568. 2002.

[6] B. Wu, C. Lin, "A high-performance and memory efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 12, pp.1615 - 628, 2005.

[7] T. Acharya, "A Survey on Lifting-based Discrete Wavelet Transform Architectures," Journal of VLSI Signal Process, Vol. 42, No. 3, pp.312-339, 2006.

[8] B. Harish, "FPGA Implementation of Multiplierless 5/3 Legal Discrete Wavelet Transform Using Lifting Approach," Proceedings on the International Conference & Workshop on Emerging Trends in Technology, pp.1066-1071, 2011.

[9] A.K. Al-Sulaifanie, A. Ahmadi, M. Zwolinski, "Very large scale integration architecture for integer wavelet transform," IET Computers and Digital Techniques, Vol. 4, No. 6, pp.471-483, 2009.

[10] K. Andra, "A VLSI Architecture for

Lifting-Based Forward and Inverse Wavelet Transform," IEEE Transactions on Signal Processing, Vol. 50, No. 4, pp.966-977, 2002.

저 자 소 개

김 석 중



2008년 한국기술교육대학교 정보기술공학부 전자공학 학사.

2012년 한국기술교육대학교 전기전자통신공학과 전자공학 전공 석사.

관심 분야 : 영상 압축
Email : sjkim2116@gmail.com

장 영 조



1979년 경북대학교 전자공학과 학사.

1982년 동 대학원 석사.
1992년 한국과학기술원 전기및전자공학과 박사.

1981년~1993년 전자통신연구원.

1993년~현재 한국기술교육대학교 전기전자통신공학부 교수.

관심분야 : 영상처리 SoC 설계, 임베디드 SoC 설계
Email: yjjang@kut.ac.kr