論文2001-38SP-3-5

시스톨릭 어레이에 기반한 SADCT의 효율적 VLSI 구조설계

(Design of an Efficient VLSI Architecture of SADCT Based on Systolic Array)

姜 泰 俊 *, 鄭 義 潤 *, 權 純 奎 *, 河 永 浩 *

(Tae-Jun Kang, Eui-Yoon Chung, Sun-Kyu Kwon, and Yeong-Ho Ha)

요 약

본 논문에서는 시스톨릭 어레이에 기반한 모양 적응적 이산 여현 변환(SADCT)의 효율적 VLSI 구조를 제안한다. 모양 적응적 이산 여현 변환은 이산 여현 변환과 달리 변환 크기가 각 블록에서의 객체의 모양에 따라 가변적이므로 기존의 시간 순환구조에서는 각 처리소자의 이용도와 처리속도가 모두 저하된다. 본 논문에서는 이러한 단점을 극복하기 위해 메모리를 필요로 하지 않는 시스톨릭 어레이에 기반한 구조를 제안한다. 제안된 구조에서는 1차원 SADCT를 연속적으로 수행함으로 처리속도를 향상시키고 첫 번째 열의 처리소자들을 마지막 열의 처리소자들과 연결하고, 입력 데이터는 각각의 재배열된 블록에서의 최대 데이터 크기에 따라 각 열에 병렬로 입력하여 처리소자의 이용도를 향상시켰다. 제안된 구조는 VHDL로 기술하고 MentorTM를 이용하여 기능검증을 수행하였다. 검증결과, 하드웨어 복잡도가 다소 증가하나, 처리속도는 기존의 방법에 비해 두 배정도 향상되었다.

Abstract

In this paper, an efficient VLSI architecture of Shape Adaptive Discrete Cosine Transform(SADCT) based on systolic array is proposed. Since transform size in SADCT is varied according to the shape of object in each block, it are dropped that both usability of processing elements(PE's) and throughput rate in time-recursive SADCT structure. To overcome these disadvantages, it is proposed that the architecture based on a systolic array structure which doesn't need memory. In the proposed architecture, throughput rate is improved by consecutive processing of one-dimensional SADCT without memory and PE's in the first column are connected to that in the last one for improvement of usability of PE. And input data are put into each column of PE in parallel according to the maximum data number in each rearranged block. The proposed architecture is described by VHDL. Also, its function is evaluated by MentorTM. Even though the hardware complexity is somewhat increased, the throughput rate is improved about twofold.

Ⅰ. 서 론

최근 디지털 영상의 방대한 데이터양를 줄이기 위해 영상압축 표준화가 진행되어 오면서 이산 여현 변환(DCT)[1]은 영상의 공간 데이터 압축의 표준으로 자리 잡아왔다. 특히, 이는 JPEG, MPEG-1,-2와 같은 블록 기반의 부호화 표준에서 각각의 블록의 공간 데이터를 줄이는데 사용되어왔다. 하지만, 최근 표준화가 이루어진 MPEG-4[2]에서는 객체 기반 부호화에 근거한 표준으로 기존의 이산 여현 변환을 통한 영상 객체 부호화는 객체의 경계선에서의 심각한 화질의 열화를 초래하게 된다. 따라서 이러한 문제점을 해결하기 위해 MPEG-4는 객체의 경계선에 해당하는 부분에 대해서

* 正會員, 慶北大學校 電子電氣工學部
(School of Electronic Electrical Eng., Kyungpook Nat'l Univ.)

※ 본 연구는 한국과학재단 특정기초연구(97-0100-0201-3)지원으로 수행되었음.

接受日字:2000年2月14日, 수정완료일:2001年2月22日

는 모양 적응적 이산 여현 변환(Shape Adaptive DCT)[3,4]을 통하여 부호화를 수행한다.

모양 적응적 이산 여현 변환은 객체를 블록으로 나눈 뒤 객체 내부에 해당하는 블록을 제외한 객체의 경계를 포함하는 블록에 대해 적용된다. 이는 경계를 포함하는 블록에서 객체 내부의 화소만을 고려하므로 이산 여현 변환과는 달리 변환 크기(transform size)가 가변적이다. 따라서, 각 변환 크기마다 필요한 여현 값을 저장하기 위해서는 이산 여현 변환에서 보다 더 큰 메모리가 요구된다. 객체는 임의의 모양이므로 행 방향과 열 방향의 일차원 이산 여현 변환(1D-DCT)를 독립적으로 처리해야 하고, 변환 영역에서의 에너지 분포가 일반적인 변환 영역에서의 분포와 같도록 하기 위해 각 일차원 변환 전에 데이터의 재정렬이 필요하다. 또한, 처음에 이루어진 일차원 이산 여현 변환 결과를 저장하기 위한 메모리도 요구된다. 따라서, 모양 적응적 이산 여현 변환을 하드웨어로 구현하기 위해서는 큰 메모리로 인해 하드웨어 복잡도가 커지고, 처리 속도 또한 느려지는 단점이 있다. 하드웨어 복잡도를 줄이기 위해서는 규칙적인 구조를 가져야 하고, 처리 속도를 개선하기 위해서는 데이터의 흐름이 메모리를 요구하지 않는 2차원 구조를 가져야한다.

기존의 시간 순환적 구조에 기반한 방법[5,6]은 구조가 규칙적이고 기능이 모듈별로 구성되어 있어 재사용이 가능하다는 장점이 있지만, 데이터의 흐름이 1차원적이어서 큰 메모리가 필요하여 소요시간(latency time)과 처리속도(throughput rate)가 긴 단점이 있다.

본 논문에서는 이러한 시간 순환 구조에 기반한 모양 적응적 이산 여현 변환 구조의 단점을 개선하고자 시스톨릭 어레이에 기반한 모양 적응적 이산 여현 변환 구조를 제안한다[7~9]. 제안된 구조는 시스톨릭 어레이가 가지는 모듈성(modularity), 규칙성(regularity), 국부적 연결성(local interconnection), 종속 연결성(pipelining), 동기화 된 다중처리(multiprocessing)등의 장점을 가지면서 가변적인 변환 크기를 처리할 수 있다.

제안한 구조에서는 처리 속도의 향상을 위해 양 끝단의 처리소자들을 연결하여 전체적으로 순환적인 구조로 만들고, 전처리 단계에서 1차적으로 정렬된 데이터열의 최대 개수에 따라 입력되는 처리소자 열의 위치를 변화시켰다. 또한, 종속 연결성을 위해 처리된 결과를 세로 방향으로 이동하던 기존 방법에서의 시간 지연을 줄이기 위해 가로 방향으로 데이터를 출력하도록 하였다. 제안한 구조의 검증을 위해 하드웨어 표현 언어인 VHDL(VHSIC Hardware Description Language)로 코딩한 후, MentorTM로 기능 검증을 수행하였다.

II. 모양 적응적 이산 여현 변환(SADCT)

MPEG-4에서의 질감정보 부호화기는 객체를 16×16의 매크로 블록(macroblock)으로 나눈 후각각의 매크로 블록에 대해 객체 영역에 해당하는 화소의 개수를 조사한다. 객체 화소의 개수가 하나라도 존재할 경우에는 다시 8×8의 부블록(subblock)으로 구분하여 부블록 내의 객체 화소의 개수에 따라 변환을 행한다. 이때 객체 영역에 속하는 화소의 개수가 전체 부블록내의 화소의 총 개수(64)와 같을 경우에는 기존의 이산 여현 변환을 사용하여 부호화하고, 객체 영역의 화소의 개수가 하나도 존재하지 않을 경우에는 부호화 하지 않으며, 그 외의 경우에는 모양 적응적 이산 여현 변환을 사용하여

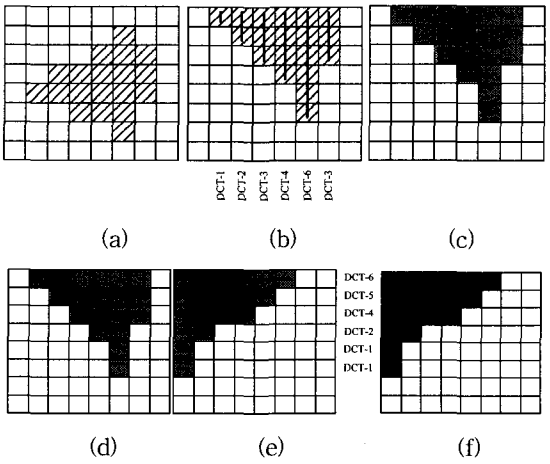


그림 1. 모양 적응적 이산 변환을 수행하는 연속적 단계, (a) 원 세그먼트, (b) 정렬과 수직 방향의 변환, (c) 수직 변환 후의 화소의 위치, (d) 수평 방향 전의 화소의 위치, (e) 정렬과 수평 방향의 변환, (f) 최종의 계수값 위치

Fig. 1. Successive steps involved for performing an 2D-SADCT. (a) original segment, (b) ordering of pels and vertical SADCT used, (c) location of pels after vertical SADCT, (d) location of pels prior to horizontal SADCT, (e) ordering of pels and horizontal SADCT used, (f) location of 2D-SADCT coefficients.

부호화한다. 이 중에서 임의의 모양을 나타내는 영상 세그먼트를 부호화하기 위한 모양 적응적 이산 여현 변환은 MPEG-4에서 인터 부호화 블록(inter-coded block)을 부호화하기 위한 방법으로, 객체의 경계 영역에서의 부호화 손실로 인한 영상화질의 열화를 막아준다.

모양 적응적 이산 여현 변환 알고리즘은 미리 정의된 이산 여현 변환 기저 함수의 직교 집합을 기반으로 하여 그림 1에서와 같이 1차원 이산 여현 변환을 수직과 수평 방향으로 독립적으로 수행한다. 그림 1의 (a)에서는 전경(foreground)과 배경(background)으로 세그먼트된 영상 블록의 한 예를 나타낸다. 그림 1의 (b)에서는 수직 방향의 변환을 수행하기 위해 각 열 $j(0<j<9)$에서 전경 영역의 길이 $N(0<N<9)$을 구한 후에, 전경 영역만을 8×8의 참고(reference) 블록의 상단으로 정렬한 다음, N의 크기에 따라서 1차원 이산 여현 변환를 적용한다. 각 데이터 (x_j)에 대한 계수값 (X_j)은[3]

$$X_k = \sqrt{\frac{2}{N}} E_k \sum_{n=0}^{N-1} x_n \cos \frac{\pi k}{2N}(2n+1), \quad k = 0, 1, \cdots, N-1 \tag{1}$$

와 같이 계산된다. 여기서,

$$E_k = \begin{cases} \sqrt{1/2}, & \text{if } k = 0 \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

이다. 그림 1의 (c)는 위의 수식을 따라 계산된 계수값 중에서 제일 작은 값, 즉 각 열 방향에서의 DC 계수값이 8×8 참고 블록의 상단에 위치함을 보여준다. 그림 1의 (d)는 수평 방향의 변환을 위해 필요한 수직 방향의 변환 결과 값을 나타낸다. 그림 1의 (e)에서는 수직 방향과 마찬가지로, 각 열의 행 방향에 대해 전경 영역의 화소 개수를 구한 후에 8×8 참고 블록의 왼쪽으로 재 정렬시키고, 식 (1)을 사용하여 각 행 방향에 대해 1차원 이산 여현 변환을 수행한다. 그림 1의 (f)는 최종적으로 생성된 2차원 모양 적응적 이산 여현 변환의 계수값들을 나타내는 것으로써, 일반적인 2차원 이산 여현 변환에서와 같이 좌 상단에 저주파 성분을 가지며, 나머지 계수 값들은 세그먼트의 실제 모양에 따라 분포하게 된다.

Ⅲ. 제안한 모양 적응적 이산 여현 변환 구조

1. 시스톨릭 어레이에 기반한 기존의 이산 여현 변환(DCT) 구조[9]

1) 체비셰프 다항식을 이용한 알고리즘

기존의 시스톨릭 어레이에 기반한 이산 여현 변환구조에서는 체비셰프 다항식이 가지는 특징을 기반으로 하고 있다. 체비셰프 다항식에 따르면

$$\cos r\theta = 2\cos\theta\cos(r-1)\theta - \cos(r-2)\theta \tag{3}$$

와 같이 삼각 함수의 현재 인자가 이전의 두 개의 값으로 표현된다. 이를 이산 여현 변환에 적용하기 위해 여현계수와 변환값을 정의하면

$$P(k, n) = \cos\frac{\pi k}{2N}(2n+1),$$

$$A(k, l) = \sum_{n=0}^{l} x_n \cos\frac{\pi k}{2N}(2n+1) = \sum_{n=0}^{l} x_n P(k, n), \tag{4}$$

와 같다. 여기서 $k, l, n = 0, l, \cdots, N-1$ 이고 $P(k, n)$은 여현계수, 그리고 $A(k, l)$는 변환값이다. 식 (1)에서의 스케일링 인자 $(\sqrt{2/N}\cdot E_k)$는 전체 구조에 영향을 주지 않으므로 생략하였다. 식 (3)과 식 (4)를 이용하여 변환값을 순환적으로 계산하는 식은

$$P(k,-1) = P(k,0) = \cos\frac{\pi k}{2N}, \quad A(k,-1) = 0,$$

$$P(k, l+1) = 2\cos\frac{\pi k}{N} P(k.l) - P(k, l-1),$$

$$A(k, l) = A(k, l-1) + x_l P(k, l),$$

$$where\ k, l = 0, l, \cdots, N-1 \tag{5}$$

과 같다. 여기서 $k, l = 0, l, \cdots, N-1$이다. 식 (5)에서 $P(k, l)$은 식 (3)의 체비셰프 다항식을 이용하여 현재의 여현값이 직전의 두 개의 여현값에 의해 순환적으로 구해짐을 볼 수 있다. 그리고 $A(k, l)$은 여현값과 출력값의 누적으로 k번째 변환값을 계산할 수 있다. 따라서 k번째 변환값은

$$X_k = A(k, N-1) \tag{6}$$

와 같다.

2) 체비셰프 다항식을 이용한 이산 여현 변환(DCT) 구조

체비셰프 다항식을 이용한 1차원 이산 여현 변환은 식 (5)에 따라서 그림 2와 같은 구조를 가진다. 그림 2에서 각 PE(processing element)는 변환 계수를 구하는 처리소자를 나타낸다.

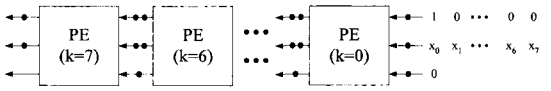


그림 2. 1차원 이산 여현 변환을 위한 1차원 시스톨릭 어레이 구조

Fig. 2. A 1D systolic array for the 1D-DCT.

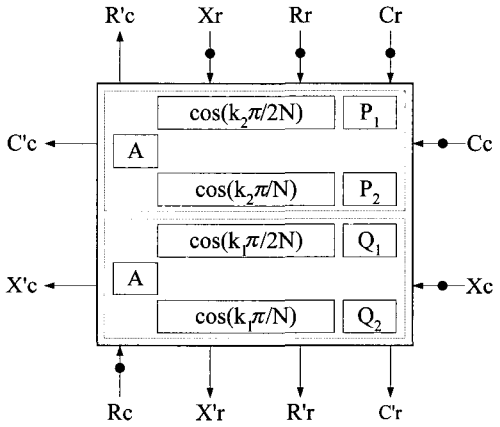


그림 3. 2차원 이산 여현 변환을 위한 처리소자

Fig. 3. The function of PE of 2D-DCT.

2차원 이산 여현 변환을 수행하기 위해서 각 처리소자는 수직과 수평 방향의 변환을 처리하는 동일한 기능의 두 개의 모듈을 가진다. 그림 3은 2차원 이산 여현 변환을 수행하는 처리소자를 나타낸다. 2차원 이산 여현 변환에서는 1차원 이산 여현 변환을 위한 처리소자 두 개를 연결하여 하나의 처리소자를 구성한다. 하나는 수직 방향의 데이터, 즉 입력 데이터를 처리하기 위한 블록이고, 또 하나는 수평 방향의 데이터, 즉 첫 번째 1차원 이산 여현 변환된 결과 값을 가지고 최종의 2차원 이산 여현 변환의 결과를 얻는다.

그림 4에서 $PE(k,l)$ $(k,l=0,1,\cdots,N-1)$ 은 2차원 이산 여현 변환에서의 각 계수값을 계산하는 처리소자를 나타낸다. 입력 데이터는 SIPO(serial input parallel output)을 거쳐 병렬로 입력된다. 1차원 이산 여현 변환에서처럼 입력 데이터가 한 블록의 첫 번째 데이터일 경우에는 변환된 결과를 다음 처리소자에 전달한다. 그러나 1차원 이산 여현 변환과 달리 각 처리소자에서의 1차원 변환결과를 좌측 처리소자로 전달하지 않고 상위 처리소자로 전달한다. 그리고 전달된 1차원 이산 여현 변환 계수를 입력으로 하여 수직 방향으로 1차원 이산 여현 변환을 하고 결과는 아래로 전달하게 된다. 따라서 최종의 변환 결과를 얻기 위해서는 최상위 처리소자에 모든 결과값이 입력될 때까지의 지연시간이 필요하게 된다. 즉, $4N$의 지연 시간이 걸린다. 그러나, 최종으로 얻어지는 결과는 종속연결성을 가지므로 매 N 사이클마다 하나의 변환 결과를 얻을 수 있다.

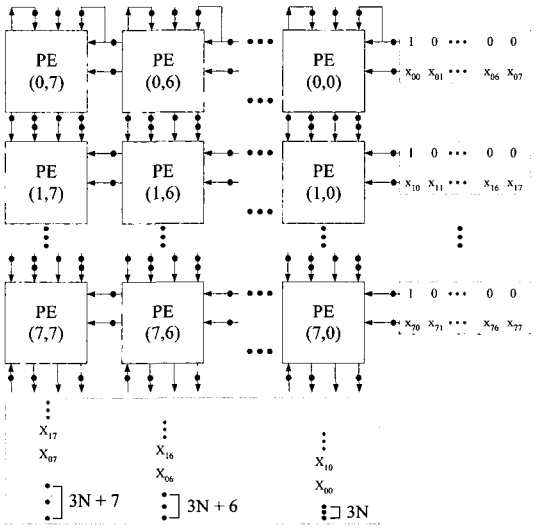


그림 4. 2차원 이산 여현 변환을 위한 시스톨릭 어레이 구조

Fig. 4. An architecture of systolic array for two dimensional DCT.

2. 제안된 모양 적응적 이산 여현 변환(SADCT) 구조

1) 구조 및 특징

제안된 모양 적응적 이산 여현 변환의 하드웨어 구조는 위에서 설명한 시스톨릭 어레이에 기반한 이산 여현 변환구조를 기반하였다. 고정된 변환 크기를 처리하는 구조를 가변적인 변환 크기를 처리하는 구조로 변형할 경우 효율과 속도가 떨어지는 문제점이 발생할 수 있다. 따라서 본 논문에서는 이러한 문제점들을 해결한 효율적인 모양 적응적 이산 여현 변환구조를 제안한다. 제안한 구조에서는 처리속도를 향상시키기 위해 입력데이터 블록의 크기를 가변시키고 이를 연속적으로 처리하기 위해 각 블록의 첫 데이터의 입력위치를 첫 시스톨릭 어레이의 처리소자에 국한 시키지 않

고 블록의 크기에 따라 가변하고 양 끝단의 처리소자들을 연결하여 순환적인 구조로 설계하였다. 그림 5는 제안한 구조를 나타내고 있다.

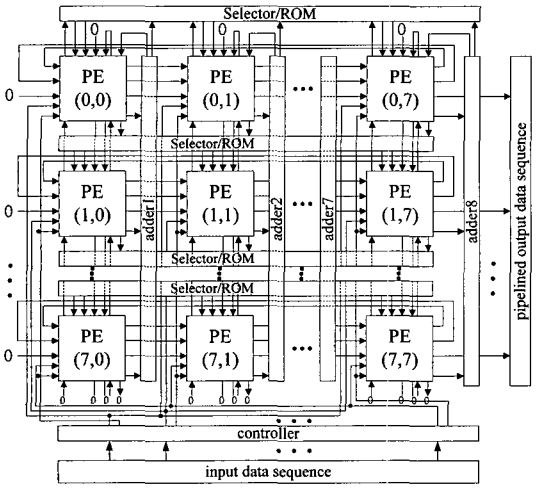


그림 5. 제안한 구조의 전체 블록도
Fig. 5. Block diagram of proposed architecture for SADCT.

입력 데이터 열은 전처리 단계에서 불필요한 데이터를 제거하고 모양 적응적 이산 여현 변환을 위해 사용되는 정보만으로 구성된 새로운 블록의 데이터들이다. 이와 같이 가변적인 블록을 사용하면 블록의 크기에 대한 부가 정보가 필요하다. 제안한 구조에서는 복잡한 제어신호를 쓰지 않고 각 블록에서 각 열의 개수를 추가로 입력한다.

제어부는 각 열 방향에서의 데이터와 그 개수를 입력으로 받아들여 이 정보들을 적절한 처리소자의 열에 병렬로 할당한다.

처리소자는 실제 변환식을 계산하는 모듈로서 가변적인 변환 크기의 처리로 인해 수직과 수평 방향에서 그 기능들이 약간의 차이가 난다. 기존의 체비셰프 다항식에 기반한 처리소자를 기본으로 하여 수직 방향의 변환에서는 입력 데이터가 블록의 첫 번째 데이터인지 아닌지를 입력데이터의 개수와 비교하여 판단하는 부분을 추가하고, 수평 방향의 변환에서는 새로이 데이터의 정렬이 필요하므로 상위 행의 처리소자에서부터 하위방향으로 변환 순서가 이루어지게 하기 위해 새로운 부분이 필요하며, 마찬가지로 첫 번째 1차원 이산 여현 변환의 결과값이 그 블록의 첫 번째 데이터인지 아닌지를 비교하여 판단하는 부분이 요구된다.

선택기/ROM은 각 처리소자에 여현값 ($\cos(k\pi/N)$, $\cos(k\pi/2N)$)을 제공하기 위한 모듈로서 데이터의 개수와 그 데이터의 순서값을 입력으로 받아들여 필요한 여현값을 호출한다.

덧셈기는 각 처리소자에서 플래그(flag) 비트를 입력으로 받아들여 수평 방향의 변환에 필요한 데이터 개수를 계산하는 모듈이다. 블록의 크기가 가변적이므로 수직 방향의 변환된 결과가 수평방향의 변환을 수행하기 위해 상단으로 이동하는 시간이 다르다. 이 경우에, 데이터의 충돌이 일어날 가능성이 있기 때문에 수직 방향의 데이터 이동을 위해서는 기본 변환 크기 ($N = 8$) 만큼의 지연시간을 둔다.

양 끝단 처리소자열의 순환적인 연결은 데이터의 입력 위치가 가변적임으로 해서 생기는 필연적인 관계로써 데이터 개수에 따라 전체 구조의 입력단으로부터 데이터를 받아들이거나 그 이전의 처리소자로부터 데이터를 받아들인다.

제안한 구조는 시간 순환적 구조에서처럼 1차원 이산 여현 변환을 처리하는 블록을 직렬로 연결한 단순한 구조가 아니라 2차원 이산 여현 변환을 1차원 이산 여현 변환으로 분해하여 수행한다. 수평 방향의 변환시에 데이터의 충돌을 막기 위해 고정 변환 크기(N=8) 만큼의 지연 시간을 주기는 했지만, 이것은 결국에는 이산 여현 변환에서의 수직 방향 변환에 걸리는 시간과 같다. 또한, 입력 데이터의 개수 정보가 있어 플래그 비트가 곧바로 만들어진다. 따라서, 수평 방향의 변환을 위한 데이터 개수를 바로 알 수 있으므로 최종적으로 출력되는 결과를 수평방향의 처리소자에서 얻을 수 있어 전체적으로 처리 속도를 향상시킬 수 있다. 그러나, 제안한 구조는 입출력 포트의 수가 많아 전체구조가 다소 복잡하다는 단점이 있다.

2) 가변적인 입력 위치의 지정

제안한 구조의 가장 큰 특징은 입력되는 데이터의 위치를 가변시킴으로써 열 단위로 데이트를 처리하여 처리속도를 향상시켰다는 것이다.

모양 적응적 이산 여현 변환 알고리즘의 첫 단계에서 수직 방향의 1차원 이산 여현 변환을 위한 입력 데이터는 각 블록(8×8)의 상단으로 정렬된다. 모양 적응적 이산 여현 변환에서는 객체 영역의 화소만으로 변환을 수행하므로 데이터의 정렬 후에 나타나는 배경

영역은 불필요하다. 따라서, 정렬 후에 나타나는 배경 영역 중에서 한 행의 배경 크기가 블록의 최대 크기 ($N = 8$)에 해당할 경우, 그 행은 입력 시퀀스에서 제외한 후에 그 다음 블록을 연결한다. 즉, 블록의 크기를 고정시키지 않고 객체의 모양에 따라 가변시켜 데이터를 입력한다.

새로이 정렬된 데이터들의 블록 크기가 가변적이므로 각 블록을 구분하기 위한 제어 신호가 필요하다. 본 구조에서는 별도로 복잡한 제어신호를 가하는 대신에 각 열에서의 객체 화소의 개수를 입력 데이터 블록의 시작과 동기시켜 입력한다. 그림 6은 제어부의 기능 블록을 나타낸 것이다.

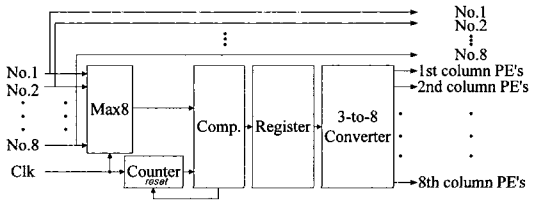


그림 6. 제어부의 기능
Fig. 6. The function of the controller.

표 1. 3-to-8 converter의 입력에 따른 출력 신호의 위치
Table 1. Position of output signal according to input in 3-to-8 converter.

input counter	position of output
000	PE(L,0)
001	PE(L,1)
010	PE(L,2)
011	PE(L,3)
100	PE(L,4)
101	PE(L,5)
110	PE(L,6)
111	PE(L,7)

출력 초기값은 첫 번째 처리소자열 $PE(0,0)$, $PE(1,0), \cdots, PE(7,0)$에 데이터와 화소 개수가 입력되도록 입력가능(enable)신호인 '1'을 할당한다. 이때, 레지스터에는 첫 번째 처리소자열을 나타내는 비트열의 값이 저장된다. 데이터의 입력이 시작될 때, 입력단에서는 한 블록의 각 데이터 열에서의 화소 개수 정보를 입력으로 받아들여 최대값을 구한다. 이 값은 시스템 클럭을 입력으로 받아 자체적으로 수행되는 카운터에서의 수와 비교된다. 카운터에서의 수가 화소 개수보다 작을 경우에는 레지스터에 저장된 값을 그대로 유지하고, 카운터에서의 수가 화소 개수와 같을 때에는 카운터를 초기화시키고 그 값을 레지스터에 있는 값과 더하여 저장한다. 이 때, 기존에 레지스터에 저장된 값과 입력되는 값의 덧셈은 8 mod로 하여 처리소자열의 위치(0~7) 범위를 벗어나지 않도록 한다. 마지막 단은 레지스터의 비트열을 입력으로 하여 올바른 처리소자열을 입력 가능하게 하는 블록으로 표 1에서와 같은 변환이 이루어진다.

3) 제안한 처리소자의 구조 및 특성

처리소자는 실제 변환 계수를 계산하기 위한 모듈로 수직 방향 변환과 수평 방향의 변환을 수행하는 두 개의 모듈로 구분된다. 기본적인 1차원 이산 여현 변환식을 계산하는 부분은 기존의 방법을 그대로 이용하고, 모양 적응적 이산 여현 변환을 위해 필요한 부분을 추가하였다. 1차원 이산 여현 변환의 계산은 식 (5), (6)을 바탕으로 한다. 요약하면, 처리소자가 수행하는 변환식은

$$A(k,l) = A(k,l-1) + x_l P(k,l)$$
$$P(k,l+1) = 2\cos\frac{\pi k}{N} P(k.l) - P(k,l-1), \qquad (7)$$

과 같이 나타낼 수 있다. 여기서 $k, l = 0, l, \cdots, N-1$이며, $P(k,l)$ 과 $A(k,l)$ 은 식 (4)에서와 같다. 식 (7)은 여현값을 체비셰프 다항식에 기반하여 순환적으로 계산하고, 이에 따라 각 변환값도 순환적으로 얻는다. 이 식을 바탕으로 하여 실제 구조를 나타내면 그림 3에서 설명한 바와 같다. 그림 7은 그림 3의 구조를 해부하여 상세하게 나타낸 것이다. DEMUX는 하나의 입력 데이터 열에서 첫 번째 데이터가 입력 될 경우와 나머지 데이터가 입력될 경우에 다른 기능을 하는 블록으로 입력 데이터를 할당한다. 곱셈기와 뺄셈기(덧셈기)는 입력 데이터와 여현값의 곱셈, 또는 다음 데이터가 입력될 때의 새로운 여현값을 계산하기 위한 블록이다. 누적기는 입력 데이터가 그 데이터 열의 첫 번째 데이터가 아닐 때에는 변환된 중간 결과를 저장하고 이전의 처리소자에서 변환된 결과를 그대로 출력하며, 입력 데이터가 그 데이터 열의 첫 번째 데이터일 때에는 최

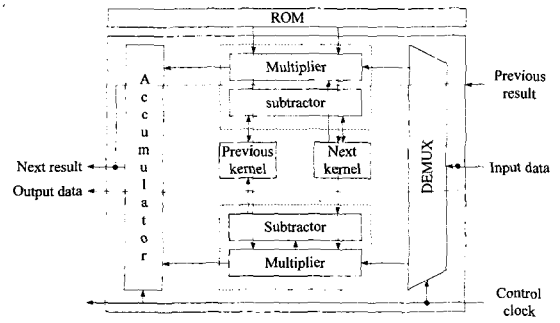


그림 7. 1차원 이산 여현 변환을 수행하는 처리소자의 구조

Fig. 7. Structure of PE for processing 1D-DCT.

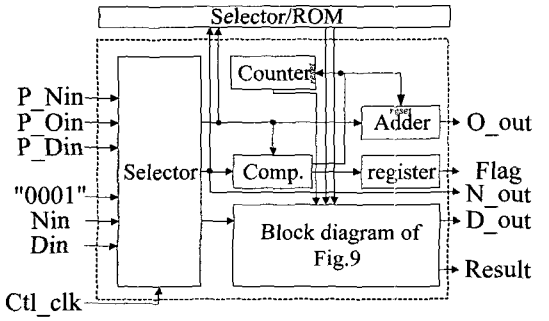


그림 8. 수직 방향의 1차원 모양 적응적 이산 여현 변환을 위한 블록도

Fig. 8. Block diagram for processing vertical 1D-SADCT.

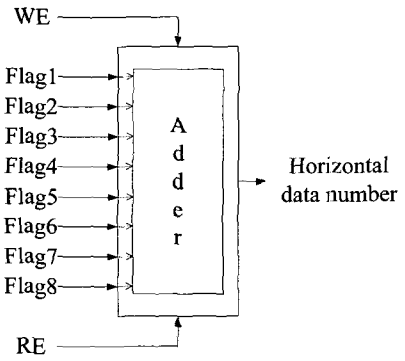


그림 9. 수평 방향의 변환을 위해 변환 크기를 계산하는 덧셈기

Fig. 9. Adder calculating the transform size for horizontal transform.

종 변환 결과를 출력한다.

① 수직 방향의 변환을 수행하는 블록

제안한 처리소자의 구조는 변환 계산을 수행하는 부분은 그림 7의 구조에 기반하여 모양 적응적 이산 여현 변환에 맞게 필요한 부분을 새로이 추가하였다. 먼저, 그림 8은 제안한 처리소자에서 수직 방향의 1차원 모양 적응적 이산 여현 변환을 수행하는 블록을 보여준다.

선택기(Selector)에서는 제어부로부터의 제어 클럭(Ctl_clk)이 '1'일 때에는 입력 데이터(Din)와 입력 화소 개수(Nin)를 받아들이고, 제어 클럭이 '0'일 때에는 이전 처리소자에서의 데이터(P_Din)와 화소 개수(P_Nin)를 입력으로 받아들인다. 또한, 전체 시스템의 입력 데이터가 입력되는 곳은 첫 번째 계수값을 계산하는 부분이므로 첫 번째 위치라는 의미로 "0001"을 부여하고, 그렇지 않을 경우에는 그 이전의 처리소자에서 출력 순서값을 입력으로 받아들인다. 선택기의 출력값 중에서 데이터는 실제의 계산을 위해 그림 7의 입력단에 보내진다. 카운터에서의 출력정보는 연산 결과가 중간 결과인지 최종결과인지를 판단하는데 이용된다. 화소 개수와 처리소자의 위치 정보를 이용하여 ROM에서 알맞는 여현값을 가져온다. 처리소자의 위치 정보는 다음 번 처리소자의 위치를 지정하기 위해 덧셈기로 보내지고, 데이터열의 개수는 곧바로 다음 처리소자에 전해진다. 또한, 처리소자의 위치 정보는 데이터열의 개수와 같을 경우에 그 덧셈기를 초기화시켜 다음 처리소자가 더 이상 변환에 이용되지 않음을 보여준다. 처리소자의 위치와 데이터열의 개수를 비교하는 블록의 또다른 출력정보는 그 처리소자가 수평방향의 변환에 이용될지의 여부를 판단하는 플래그 비트를 전체 시스템의 덧셈기에 보내진다. 덧셈기는 같은 열 방향에 위치한 처리소자 열($PE(0,L), PE(1,L), \cdots, PE(7,L)$)로부터 플래그 비트를 입력으로 하여 수평 방향의 이산 여현 변환을 위한 변환 크기를 계산하여 선택기/ROM에 전달한다. 그림 9는 수직과 수평 방향의 1차원 모양 적응적 이산 여현 변환을 위한 중계 역할을 하는 덧셈기 블록을 나타낸다.

위에서 설명한 것처럼 덧셈기는 RE신호에 따라 입력 플래그 비트를 더하여 4비트의 수평 방향의 객체 화소의 개수를 계산한 후 WE에 따라 최상위 처리소자에 전달한다. RE, WE는 각 처리소자열의 순서에 따라 8클럭 간격으로 입력된다.

② 수평 방향의 변환을 위한 블록

수평 방향의 1차원 모양 적응적 이산 여현 변환은 수직 방향의 연산 결과와 객체 화소의 개수정보와 처리소자의 위치정보를 입력으로 받아 최종 계수값을 계산한다. 그림 10은 수평 방향의 변환을 위한 블록도를

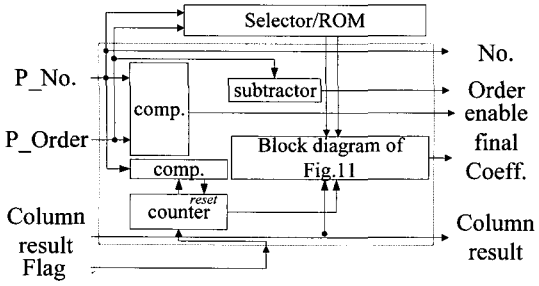


그림 10. 수평 방향의 1차원 모양 적응적 이산 여현 변환을 위한 블록도

Fig. 10. Block diagram for horizontal 1D-SADCT.

나타낸다.

수평 방향의 1차원 모양 적응적 이산 여현 변환은 수직 방향의 변환에서 얻은 결과를 이용한다. 변환이 시작되면 같은 열 방향에 있는 처리소자들은 변환 결과를 한 행 위의 처리소자에 전달한다. 이 중에서 수직 방향의 변환에 이용되지 않은 처리소자는 아무런 결과도 없을 것이므로 이들 값은 변환에 이용되어서는 안 된다. 따라서, 수직 방향 변환의 출력인 플래그 비트를 입력으로 하여 블록의 수행 여부를 판가름한다. 즉, 플래그 비트가 '1'이면 변환을 수행하고, 플래그 비트가 '0'이면 변환을 수행하지 않고 그 값들을 유지하게 된다.

4) 처리소자의 순환적 연결 구조

제안한 구조에서는 전처리 단계에서 객체의 경계 블록에서 비객체 영역에 해당하는 부분을 제거하여 크기가 가변적인 데이터 블록을 만들었다. 또한, 이러한 데이터를 입력할 처리소자의 위치를 그 블록의 열 방향 크기에 따라 가변시켰기 때문에 한 블록이 변환되는 시간적 차이가 생기게 된다. 따라서, 기존의 일반적인 시스톨릭 어레이 구조에서와 같이 첫 번째 처리소자 ($PE(l,0)$)와 여덟 번째 처리소자($PE(l,7)$ ($l = 0,1, \cdots, N-1$))이 서로 연결되지 않은 비순환적인 구조이면 가변적인 데이터 블록을 처리할 수 없다. 따라서 이러한 문제점을 해결하기 위해 첫 번째 처리소자와 여덟 번째 처리소자를 서로 연결하고, 수평과 수직 방향의 연산은 최대 변환 크기(N)에 해당하는 클럭 만큼의 지연시간을 두어 데이터 충돌이 없이 가변적이 데이터 블록을 처리하도록 하였다. 그림 11은 처리소자의 순환적 연결 구조를 간략히 나타낸 것이다.

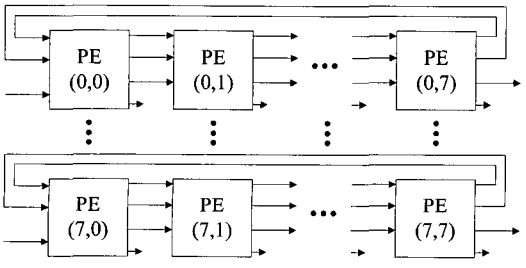
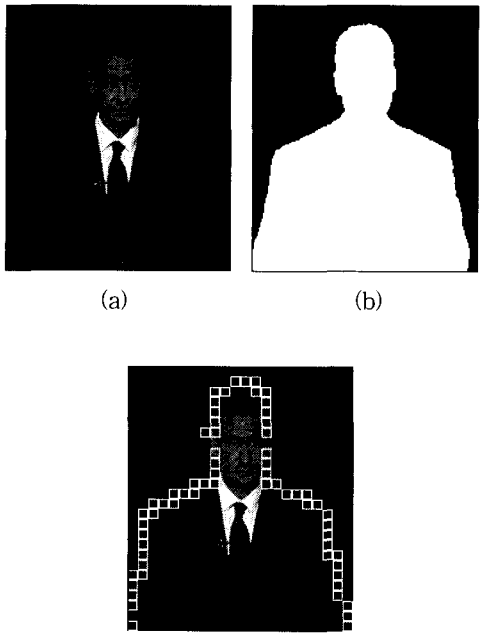


그림 11. 처리소자의 순환적 연결 구조

Fig. 11. Recursive structure of PE's in the same row.

Ⅳ. 실험 및 고찰

본 논문에서는 제안한 구조를 검증하기 위해 하드웨어 기술 언어인 VHDL을 이용하여 설계하였다. 또한, 캐드 툴(CAD Tool)인 MentorTM를 이용한 기능검증을 수행하였다. 인가한 시스템 클럭은 50Mhz로 하여, Sun Sparc UltraII워크스테이션에서 실험하였다. 실험 영상은 CIF포맷의 'News' 영상을 사용하였다. 그림 12는 'News' 영상에서 세그먼테이션 마스크로 추출된 남자앵커 객체를 추출한 그림이다. 그림 12에서 (a)는 원



(a) (b)

(c)

그림 12. 실험 영상(176x208), (a) 원 VOP, (b) 세그먼트 마스크, (c) SADCT 블록

Fig. 12. Test image. (a) original VOP, (b) segment mask, (c) blocks for SADCT.

VOP(original video object plane)를 나타내고, (b)는 세그먼트 마스크를 보여준다. (c)에서의 사각형 영역은 객체의 질감정보 부호화에서 모양 적응적 이산 여현 변환이 적용되는 영역이다.

표 2. 시간 순환적 구조에 기반한 방법과의 비교

Table 2. Comparison of Time-recursive method and proposed method.

	Time-recursive architecure[6]	Proposed architecture
Memory	YES	NO
Number of multiplier per PE	2	4
Throughput rate	4N	2N

표 2는 제안한 구조를 기존의 시간 순환적 구조에 기반한 방법과 비교한 결과를 보여준다. 시간 순환적 구조를 사용한 기존의 방법에서는 하나의 계수가 얻어지는 처리 속도(throughput rate)가 1차원 DCT를 처리하는 두 개의 블록과 1D-DCT 결과를 저장하는 메모리로 인해 최대 변환 크기의 4배, 즉 4N이었다. 그러나, 본 구조에서는 메모리를 필요로 하지 않고, 또한 종속 연결한 데이터 출력으로 인해 최대 변환 크기의 2배인 2N의 처리속도를 얻었다. 즉, 2배의 빠른 처리속도를 보여준다.

V. 결 론

본 논문은 시스톨릭 어레이에 기반한 모양 적응적 이산 여현 변환의 효율적인 VLSI 구조를 제안한다. 제안한 구조에서는 메모리를 필요로 하지 않으면서 각 처리소자에 간단한 모듈을 추가함으로써 데이터의 자동 정렬이 이루어지게 하였다. 또한, 입력 데이터 열의 최대 데이터 개수에 따라 입력되는 처리소자열을 가변시키고, 가변 데이터 입력 위치로 인해 출력 결과가 서로 충돌하는 것을 방지하기 위해 양 끝단의 처리소자를 서로 연결하여 처리 속도를 향상시켰다.

MentorTM로 검증한 결과, 시간 순환적 구조에 기반한 방법보다 2배의 처리 속도가 향상됨을 보았다. 데이터가 병렬로 입출력되므로 시스템의 하드웨어 복잡도가 증가하지만 메모리가 사용되지 않고 제어 블록이 간단하여 시간 순환적 구조에 비해 향상된 성능을 가진다. 또한, 제안한 구조는 각 처리소자가 규칙적이고 모듈성을 지니므로 재사용이 가능한 장점을 가지고 있다.

참 고 문 헌

[1] Anil K. Jain, Fundamental of digital image processing, Prentice Hall, pp. 150-154, 1989.

[2] "JTC1/SC29/WG11 MPEG-4 video verification model Version 11.0," Technical Report N2172, International Organization for Standardization ISO/IEC (1998).

[3] Thomas Sikora and Bela Makai, "Shape-Adaptive DCT for Generic Coding of Video", IEEE Trans. on Circuits and Systems for Video Technology, vol. 5, no. 1, pp. 59-62, February 1995.

[4] Peter Kauff and Klaas Schuur, "Shape-Adaptive DCT with Block-Based DC Separation and ΔDC Correction", IEEE Trans. on Circuits and Systems for Video Technology, vol. 8, no. 3, pp. 237-242, June 1998.

[5] Vishnu Srinivasan and K. J. Ray Liu, "VLSI Design of High-Speed Time-Recursive 2-D DCT/IDCT Processor for Video Applications", IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, no. 1, pp. 87-96, February 1996.

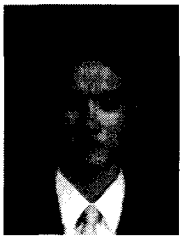
[6] Thuyen Le, Martin Wendt and Manfred Glesner, "VLSI Architecture of a Time-Recursive 2-D Shape-Adaptive DCT Processor for Generic Coding of Video", International Conference on Signal Processing Applications & Technology, pp. 1238-1242, September 1997.

[7] Yu-Tai Chang and Chin-Liang, "New Systolic Array Implementation of the 2-D Discrete Cosine Transform and Its Inverse", IEEE Trans. on Circuits and Systems for Video Technology, vol. 5, no. 2, pp. 150-157, April 1995.

[8] K. J. R. Liu and C. T. Chiu, "Optimal Unified Architectures for the Real-Time Computation of Time-Recursive Discrete Sinusoidal Transforms", IEEE Trans. on Circuits and Systems for Video Technology, vol. 4, no. 2, pp. 168~180, April 1994.

[9] Chin-Liang Wang and Chang-Yu Chen, "High-Throughput VLSI Architectures for the 1-D and 2-D Discrete Cosine Transforms", IEEE Trans. on Circuits and Systems for Video Technology, vol. 5, no. 1, pp. 31-40, February 1995.

저 자 소 개



姜 泰 俊(正會員)

1998년 2월 : 경북대학교 공과대학 전자공학과 졸업(공학사). 2000년 2월 : 경북대학교 대학원 전자공학과 졸업(공학석사). 2000년 3월~현재 : LG전자 시스템 IC 사업담당, SIC R&D센터 연구원. ※ 주관심분야 : 영상부호화, VLSI설계, 신호처리, 컴퓨터 비젼 등

鄭 義 潤(正會員)

1992년 2월 : 경북대학교 공과대학 전자공학과 졸업(공학사). 1997년 2월 : 경북대학교 대학원 전자공학과 졸업(공학석사).

權 純 奎(正會員)

1997년 2월 : 경북대학교 공과대학 전자공학과 졸업(공학사). 1999년 2월 : 경북대학교 대학원 전자공학과 졸업(공학석사). 2001년 2월 : 경북대학교 대학원 전자공학과 박사과정 수료. 2001년 2월~현재 : LG전자 디스플레이 연구소 주임연구원. ※ 주관심분야 : 스테레오, VLSI설계, 신호처리, 컴퓨터 비젼 등

河 永 浩(正會員) 第 38卷 S編 第 1號 參照