論 壇

HDTV의 DCT 시스톨릭 어레이

李門浩* 朴鍾五** 李光宰*** 朴周用*
(* 전북대학교 정보통신공학과, ** 태산정밀연구원, *** 전주KBS 기술국)

■ 차 례 ■

1 서 론
2 HDTV의 DCT
3 시스톨릭 어레이의 개념
4 DCT의 시스톨릭 어레이
4.1 DCT
4.2 Chen의 알고리즘
4.3 Lee의 알고리즘
4.4 제안한 알고리즘과의 비교
4.4.1 제안한 1차원 시스톨릭 어레이
4.4.2 제안한 2차원 시스톨릭 어레이
5 성능분석및 컴퓨터 시뮬레이션
6 결 론

1 머리말

정보 교환의 방법으로써 음성, 문자, 영상중 정보전달 효과가 가장 큰 것은 영상이다. 그런데 디지탈 영상 데이타(digital image data)를 전송할 때 큰 대역폭이 요구되므로 데이타 압축(data compression)이 필요하다.[13][14] 영상 데이타의 많은 중복성(redundancy)을 줄이기 위한 부호화 방법으로는 인접 화소로 부터 현재 값을 추정하고 실제 값과의 차이를 부호화하여 전송하는 예측 부호화(Predictive coding)방법과, 에너지가 어느 특정 지역에 밀집하는 효과를 이용한 변환 부호화(Transform coding)방법, 그리고 예측 부호화와 변환 부호화를 혼합한 혼합 부호화(Hybrid coding)등이 있다.

예측 부호화 방법에는 DPCM(Differential Pulse Code Modulation), ADPCM(Adaptive DPCM), Frame differential coding, DM(Delta Modulation)등이 있는데 이 방식은 전송 에러가 누적되는 단점이 있다.[15][16]

또한 변환 부호화는 영상이 모든 화소(picture element=pel)에 분산된 에너지를 변환후 에너지가 집중(compaction)되어 있는 낮은 변환 계수(transform coefficient)에 대해서 많은 전송 비트를 할당하고 낮은 에너지를 갖는 계수들을 버림으로써 영상 데이타를 압축할 수 있다. 여기서 에너지란 각 변환 계수의 분산(variance)을 의미한다. 이 방식의 특성은 1bpp(bit per pel) 이하에서도 좋은 성능을 가지며, 영상 데이타를 분할하여 독립적으로 처리하기 때문에 전송 오차의 영향을 분할된 영역으로 국한시킬 수 있고, 영상의 통계적 변화 및 전송 잡음(channel noise)에 의한 성능 저하가 적은 반면 복잡한 많은 계산량으로 인해 하드웨어 구현이 매우 어려운 점이 있다.[15][16]

변환 기법으로는 KLT(Karhunen-Loeve Transform), DCT(Discrete Cosine Transform), Hadamard, Haar, Slant, Sine 변환등이 있으

며, 이들 중 KLT는 변환 계수들의 에너지 집중(energy compaction)현상이 가장 우수하기 때문에 압축 효율면에서 가장 이상적인 최적(optimum) 변환이지만 실제 하드웨어 구현이 거의 불가능하다.[21] 반면 DCT는 성능면에서 KLT에 가장 근접하기 때문에 영상 데이타의 압축에 널리 이용되는 변환 방식이다.[1]

그런데 대부분의 변환 기법과 마찬가지로 DCT 는 계산량이 많아 실시간 계산이 어렵기 때문에 이와 같은 문제점을 해결하기 위하여 Chen, Lee등의 여러 고속 DCT(FDCT=fast DCT) 알고리즘들[2 3]이 제안되었으나, 영상 데이타와 같은 정보량이 매우 많은 데이타들은 이러한 고속 알고리즘으로도 실시간 처리가 매우 어렵다. 따라서 일반적인 Von-Neumann형태의 프로세서를 이용하여 순차적인 처리(sequential processing)를 하는 방법과는 근본적으로 다른 병렬 처리(parallel processing)를 이용하여 DCT를 실시간으로 계산하는 방법에 대한 연구가 필요하다. 병렬 처리란 알고리즘 내에서 서로 독립적으로 계산될 수 있는 부분은 서로 다른 프로세서에서 계산될 수 있도록 하는 것이다. 최근에는 단일 프로세서가 아닌 다수 개의 처리요소(PE=Processing Element)를 병렬로 사용하여 고속연산이 가능하고 급속히 발달하고 있는 VLSI(Very Large Scale Integration) 구현이 용이한 구조 즉, 시스톨릭 어레이(Systolic array) 와 Wavefront array 등이 H.T. Kung과 S.Y. Kung등에 의해 제안되었다. [4-12]

시스톨릭 어레이는 병렬 처리할 알고리즘을 독립적으로 계산되는 간단한 부분들로 분해하고 각 부분의 계산을 수행하는 프로세서들을 구성하여 이들을 규칙적으로 하나의 VLSI칩에 집적시키는 것이다. 이 구조의 특성을 요약하면 모듈화(modularity), 규칙성(regularity), 국부 연결성(local connectivity), 그리고 파이프라인(pipeline) 처리이다. 즉, 알고리즘을 서로 독립적인 부분들로 나누어 각 부분들이 수행되는 프로세서들을 하나의 칩에 배열할 때 이들이 가능하면 같은 구조를 갖도록 하고, 또한 각 프로세서들이 규칙적인 연산과 데이타 교환을 하도록 하여 설계시의 복잡성과 비용을 낮출 수 있다.

여러 종류의 신호 처리 알고리즘들이 시스톨릭 어레이로 구현되고 있으나, Chen, Lee등의 고속 DCT 알고리즘등은 시스톨릭 구현이 매우 어렵다는 것을 확인했다. 그 이유는 데이타의 신호 흐름도를 보면 시스톨릭 구조의 조건인 규칙성이나 국부 연결성이 부족하기 때문이다. 그런데 N점 DCT는 2N점 DFT(Discrete Fourier Transform) 로 계산이 가능하므로 2N점 FFT(Fast Fourier Transform)를 이용하여 계산할 수 있으나, FFT도 FDCT와 같은 이유로 시스톨릭 구현이 어렵다. 반면에 DFT 자체는 벡터와 벡터의 곱을 계산하는 시스톨릭 어레이를 변형하여 수행이 가능하고, 여러 형태의 DFT형 시스톨릭 어레이들이 제안되었다.[5 7 11]

따라서 본 논문은 DFT의 시스톨릭 어레이를 이용하여 N점 DCT를 병렬 연산으로 처리할 때 2N점 DFT의 시스톨릭 어레이를 이용하는 것보다 복잡성이 낮은 시스톨릭 어레이를 구현하기 위하여, FDCT나 FFT보다 순차적 계산으로는 계산 속도가 낮더라도 시스톨릭 어레이에 적합한 알고리즘과 구조를 제안한다. 그 한가지 방법으로서 Kung의 DFT 연산을 위한 시스톨릭 어레이를 이용하기 위하여 DFT와 유사한 변형 DFT(MDFT=modified DFT)를 정의하고 고르첼(Goertzel) 알고리즘을 적용하여 N점 DCT를 계산하는데 N개의 처리요소(PE)와 한개의 곱셈기만이 필요한 1차원 구조를 제안하고, 또 다른 방법은 하드웨어 부담은 1차원 어레이보다 크지만 처리 성능을 향상시킬 수 있는 2차원 시스톨릭 어레이를 제안한다.

본 논문은 2장에서 HDTV와 DCT 관계를, 3장에서 시스톨릭 어레이의 개념과 특성을 소개하였고, 4장에서는 DCT와 그의 고속 알고리즘들을 제안한 변형 DFT 알고리즘과 연산 횟수와 신호 흐름도면에서 비교 분석하였으며, N점 DCT 연산을 수행하는 1차원, 2차원 시스톨릭 어레이와 어레이를 구성하는 처리요소와 곱셈기의 구조를 제안하였다. 마지막으로 6장에서 결론

을 맺었다.

|2| HDTV의 DCT

DCT는 HDTV의 디지탈 비디오 코덱 그림 2.1, 2.2, VCR · G 4 Fax 영상부분 압축 Codec, 비디오 폰의 Px64 KBPS의 CCITT H.261 그림 2.3과 JPEG(Joint Photographic Experts Group) 의 표준으로 널리 쓰이고 있다.

그림 2.2는 HDTV의 신호를 대역 분할 필터 뱅크를 거쳐 DCT로 양자화 압축하고 있다.

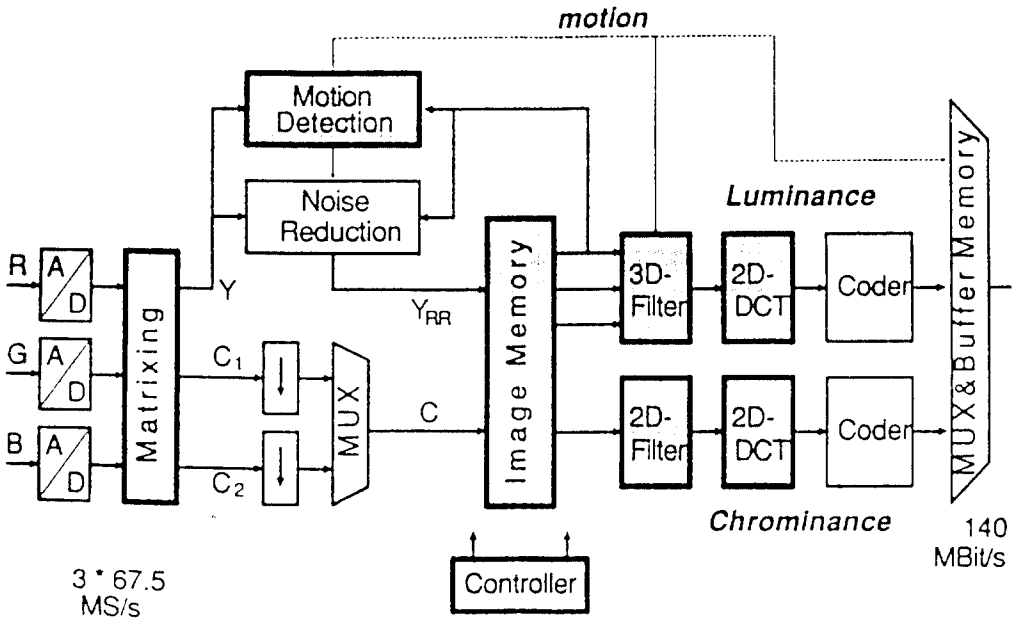


그림 2.1 HDTV의 영상 압축 시스템

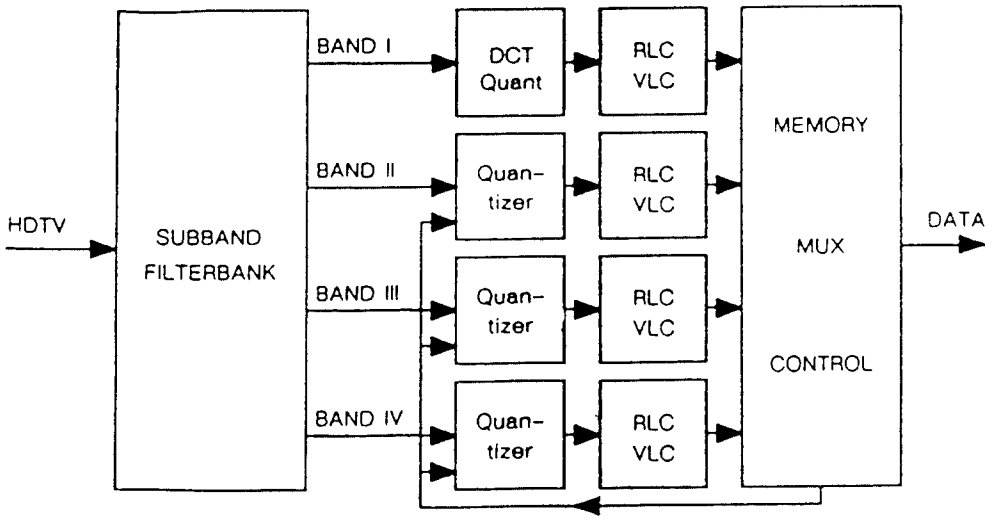


그림 2.2 HDTV의 대역 분할 coder

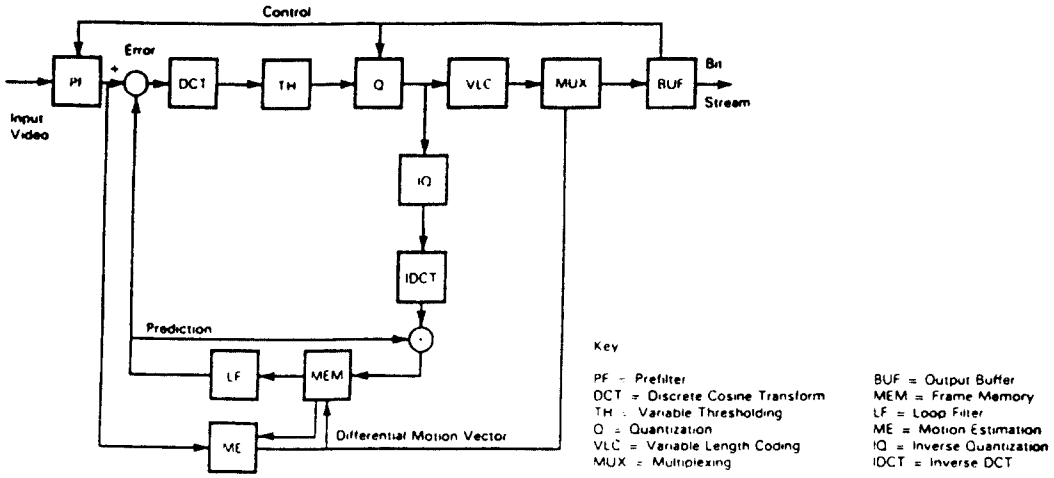


그림 2.3 H.261의 예측 변환 루프

표 2.1 국내 DCT 연구 동향[41X42]

제 안 자 및 발표년도	알고리즘의 특징	발 표 지
이 병 기 1984 1989	N/2 DCT를 이용 곱셈수가 적다 일반적으로 많이 이용	1. IEEE. Trans on ASSP vol. 32, no.12, 1984. 2. IEEE. Trans on ASSP vol. 37, no. 2, 1989.
이 상 욱 1990	N point DFT와 Prime Factor이용 시스톨릭 어레이로 VLSI 용이	1. IEEE. Trans on ASSP vol. 38, no. 1, 1990.
이 문 호 1989 1990	1차원 N point MDFT 2차원 N point IDFT (:Index mapping) 시스톨릭 어레이로 VLSI 용이	1. Electronics Letters vol. 25, no. 25, Dec. 1989. 2. IEEE. Trans on CAS vol. 37, no. 10, 1990. 3. Multidim, Systems and Signal processing vol.1, no.4, 1990.

DCT를 1 Chip으로 구현하기 위해서는 고속 알고리즘을 병렬 처리로 해서 만드는 방법과 HT(Hadamand Transform)나 DFT(Discrete Fourier Transform)를 이용하여 시스톨릭 어레이를 구현하는데 8×8 블록, 10~20 MHz이상이다.

국내에서 DCT 연구가 한국과학원, 서울대, 중앙대, 경북대, 전북대 등에서 활발하며 고속 알고리즘과 하드웨어 구현을 위한 연구 상황은 표2.1과 같다.

[3] 시스톨릭 어레이의 개념

대규모 집적회로 시스템 설계시 VLSI 기술이 제공하는 장점들을 최대한으로 활용하기 위해서는 다음과 같은 기본적인 설계원칙(design philosophy) 들을 고려하여 알고리즘 및 하드웨어 아키텍춰를 결정해야 한다.[9]

첫째, 칩의 집적도가 증가함에 따라 단위 소자에 대한 제조비용과 설계 비용중 설계비용이 상대적으로 증가하게 되었다. 이는 VLSI 자체가 가지고 있는 고유의 복잡성에 기인한 것으로,

Mead와 Conway[32]에 의하면, 구조적(structured) 이며, 규칙적(modular)인 설계기법을 도입함으로써 해결할 수 있다. 즉, 전체 시스템은 단순하고, 반복적인 구조를 가지며, 상호연결이 단순한 여러개의 처리요소들로 분해함으로써 설계비용, 설계시간은 물론 설계오류도 현저하게 줄일 수 있다.[33]

둘째, 여러개의 처리요소들을 동시에 이용하는 병렬처리 하드웨어 구조에서는 처리요소들간의 communication 구조가 그 시스템의 성능을 좌우하게 된다. 처리요소들간의 communication cost는 면적, 시간요소에 의해 좌우되므로 단순하고, 규칙적인 communication 구조를 갖도록 즉, VLSI에서는 각 요소의 연결에 많은 칩 면적이 필요하므로 VLSI내에 배열된 각 처리요소간의 데이타 교환은 인접한 처리 요소들 간에만 이루어지도록 함으로써 칩의 집적도를 높일 뿐만 아니라 연산속도를 빠르게 할 수 있다.[34][35]

셋째, VLSI 시스템과 메모리간의 communication bottleneck 때문에 VLSI 시스템의 최대 throughput은 입출력대역폭 (I/O bandwidth)의 제약을 받게 된다.[4] 따라서, 한번의 I/O access를 통해 얻어진 데이타로 최종 결과가 얻어질 때까지 연속하여 연산을 할 수 있는 구조가 바람직하다.

이상과 같은 VLSI 설계 원칙들을 최대한으로 만족시켜 주는 하드웨어 아키텍춰로써 시스톨릭 어레이가 크게 주목받고 있으며, 이에 대한 많은 연구 결과들이 발표되고 있다.[4-12]

시스톨릭 어레이는 1978년 H.T. Kung과 C.E. Leiserson에 의해 처음으로 그 개념이 소개된 이래, VLSI 기술의 발달과 더불어 그 중요성이 부각되기 시작한 새로운 하드웨어 아키텍춰이다.[10]

시스톨릭 어레이는 SIMD(Single-Instruction Multiple-Data stream), MIMD(Multiple-Instruction Multiple-Data stream) 및 Wavefront Array등과 함께 병렬 처리 구조(Parallel Processing Architecture)에 속하며, 이들 병렬처리 구조는 Timing과 Data I/O에 따라 그림

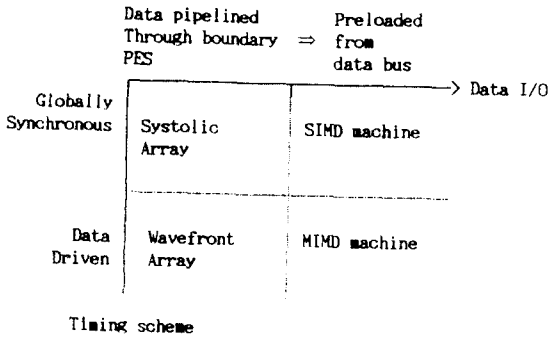


그림 3-1 병렬 처리 구조의 분류

3-1과 같이 분류된다. 시스템의 Timing 측면에서 보면, 시스톨릭 어레이는 SIMD 구조와 같이 동기(Global Synchronous)를 갖는 반면, Wavefront Array나 MIMD구조는 Data-driven, 즉 비동기(Asynchronous)를 갖는다. 한편 Data I/O 측면에서 보면, 시스톨릭 어레이는 Wavefront Array와 같이 어레이의 외곽에 위치한 처리요소들을 통해 pipelining이 되며, SIMD 구조와 MIMD 구조는 데이타 버스를 통하여 preloading되는 점이 다르다.

간략히 말하면, 시스톨릭 어레이는 동기(global synchronous clock)에 의해 동작되며, 파이프라인 data I/O를 갖는 병렬처리 구조의 일종이라 할 수 있다.

시스톨릭 시스템은 기존의 어레이 프로세서가 갖는 고속 연산 throughput 장점외에 처리요소 사이의 규칙적인 최소 인접 연결(nearest neighbor interconnection), 국부적인 데이타 교환, 단순하고 규칙적인 데이타 및 제어 흐름등의 장점을 갖는다.

또한, 시스템 설계 측면에서도 시스톨릭 어레이는 다음과 같은 여러가지 장점을 갖는다. 즉, 반복적 연결, 구조적인 확장성등에 의하여 전체 시스템의 설계가 단순한 셀의 설계로 분해될 수 있으므로 설계, 레이아웃, 검증등에 소요되는 시간과 경비는 물론, 설계 오류를 줄일 수 있다. 또한 시스톨릭 어레이의 국부적인 데이타 교환 특성은 칩의 전체 배선면적을 크게 줄여

집적도를 높일 수 있을 뿐만 아니라, 배선에 의한 기생 정전 용량의 감소로 전체적인 성능이 향상된다.

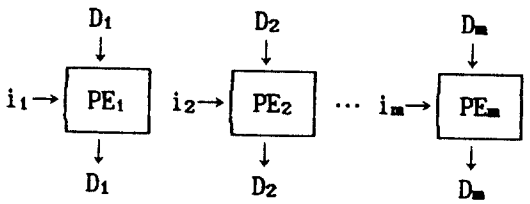
이상에서 언급한 시스톨릭 어레이의 특징을 요약하면 다음과 같다.

- 동기에 의한 동작(synchrony)
- 단순한 처리요소(processing element)로 구성
- 단순하고 규칙적인 데이타의 흐름(regularity)
- 최소인접배선을 통한 국부적인 데이타 교환(local communication)
- 규칙적인 확장성(modularity)
- 고도의 병렬성(concurrency)과 파이프라인(pipeline) 처리

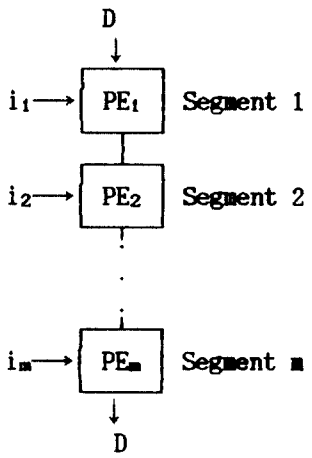
한편, 시스톨릭 어레이의 탁월한 연산능력은 병렬연산과 파이프라인 처리에 기인한다. 병렬처리는 m개의 처리요소에서 m개의 데이타가 동시에 처리되는 것을 말하며, 파이프라인 처리는 하나의 process가 m개의 subprocess로 분해되어 사슬(chain) 형태로 연결된 m개의 처리요소들을 순차적으로 통과하면서 처리되는 것을 말한다. 데이타의 파이프라인 처리는 데이타가 일단 어레이에 입력된 후, 최종결과가 나올 때까지 반복적으로 사용됨으로써 메모리 입출력 대역폭(Memory I/O bandwidth)의 제약을 제거할 수 있다는 특징을 갖는다. 그림 3-2는 시스톨릭 어레이가 갖는 병렬 처리와 파이프라인 처리를 개념적으로 보여주고 있다.

시스톨릭 어레이의 형태는 실현시키고자하는 알고리즘에 따라 1차원 또는 2차원등의 여러가지 배열 형태를 가지며, 그 대표적인 예는 그림 3-3과 같다.

시스톨릭 어레이는 그 응용 분야가 광범위하며, 특히 대량의 데이타에 대한 단순한 연산이 반복적으로 이루어지는 연산 집약적(computaion intensive)인 특성과 연속되는 연산사이에 국부적인 데이타 교환(local data communication) 특성을 갖는 알고리즘들은 시스톨릭 어레이 구현에 적합하다. 최근까지 발표된 시스톨릭 어레이의

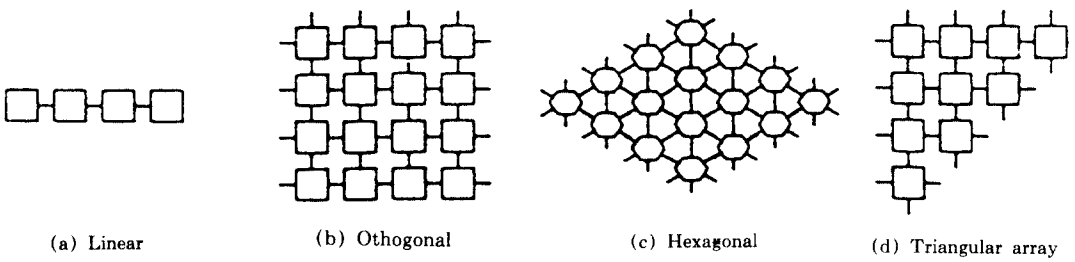


(a) Parallel Processing



(b) Pipelined processing

그림 3-2 병렬 처리와 파이프라인 처리.



(a) Linear (b) Othogonal (c) Hexagonal (d) Triangular array

그림 3-3 시스톨릭 어레이의 형태

응용분야는 크게, 신호 및 영상 처리 분야, matrix 연산분야 그리고 Non-Numeric 응용분야 등으로 나눌 수 있다.

[4] DCT의 시스톨릭 어레이

본 장에서는 영상 데이타의 감축을 위한 변환 부호화로써 성능면에서 최적(optimum)인 KLT와 가장 유사한 성능을 갖는 DCT 및 Chen, Lee등에 의해 제안된 고속 DCT 알고리즘들과 제안한 변형 DFT 알고리즘과 연산 횟수와 신호 흐름도면에서 비교 분석하였으며, DCT연산을 수행하는 1차원 및 2차원 시스톨릭 알고리즘과 그 구조를 제안하였다.

4.1 DCT

실제 영상 데이타 값과 이미 부호화된 데이타로부터 예측된 값과의 차이를 양자화(quantization)하여 전송하는 예측 부호화와는 달리 변환 부호화에서는 영상 전체를 몇 개의 작은 구역으로 분할한 후, 각 구역을 새로운 영역으로 변환시킨다. 이를 변환 영역(transform domain)이라고 하며, 일반적인 영상의 경우 구역 전체에 분산된 에너지가 변환 영역에서 몇 개의 변환 계수(transform coefficient)에 집중된다.[19][20] 변환 부호화 방식의 송수신 블럭도는 그림 4-1과 같다.

변환 기법으로는 KLT(Karhunen-leove transform), DCT, Hadamard, Haar, Slant, Sine 변환등이 있으며, 이들 중 KLT는 변환 계수들의 에너지 집중(energy compaction) 현상이 가장 우수하기 때문에 압축 효율면에서 가장 이상적이지만, 영상의 공분산 함수(covariance function)에 근거한 변환이기 때문에 TV 영상과 같이 각 구역마다 공분산 행렬이 다른 경우 매번 공분산 행렬을 구해야 하므로 하드웨어의 실현이 곤란하다.[22][23] 영상 처리의 응용을 위해 단일 변환(unitary transform)으로써 역변환이 용이한 준 최적(sub-optimal) 변환중 영상 데이타가 stationary하고 지수형 상관 계수(exponential correlation coefficient)를 갖는 1차 Marcovprocess 라고 가정할 경우, DCT는 최적 변환인 KLT와 가장 유사한 성능을 가지므로, 변환을 이용한 데이타의 감축에 널리 사용된다.

입력 데이타 시퀀스(Input data sequence) f(n), n=0,1,…,N-1에 대한 순방향 DCT(FDCT= forward DCT) 및 역방향 DCT(IDCT=inverse DCT)는 각각 식(4-1)과 식(4-2)와 같이 정의된다.[1]

$$F(k)=\frac{2}{N}\,C(k)\sum_{n=0}^{N-1} f(n)\cos[(2n+1)k\pi/2N] \tag{4-1}$$

$$f(n)=\sum_{n=0}^{N-1} C(k)\,F(k)\cos[(2N+1)k\pi/2N] \tag{4-2}$$

$$C(k)=\begin{cases}1/\sqrt{2}, & \text{for } k=0\\ 1, & \text{for } k\neq 0\end{cases}$$

$$n,k=0,1,\cdots,N-1$$

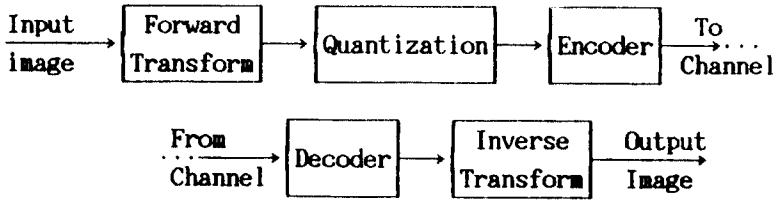


그림 4-1 변환 부호화 방식의 송수신 블럭도

여기서 F는 변환 영역의 변환 계수를 나타내고, cosine항을 kernel이라고 하며 순방향과 역방향의 kernel이 같음을 알수 있다. 식(4-1)에서 N개 입력에 대한 DCT는 기존의 2N점 FFT[24]를 사용하여 DFT[25]를 구함으로써 얻을 수가 있다.

4.2 Chen의 고속 알고리즘

Chen의 고속 알고리즘은 행렬 분해(matrix decomposition)에 기본을 두고 있다. N×1 데이타 벡터 [f]에 대한 DCT인 식(4-1)은 식(4-3)과 같이 행렬의 형태로 나타낼 수 있다. 이때 [F]는 변환 계수 벡터이며 N×N kernel 행렬 $[A_N]$은

$$[F]=\frac{2}{N}\ [A_N][f] \tag{4-3}$$

$$[A_N]=[C(k)\ \cos\frac{(2n+1)k\pi}{2N}\ ] \tag{4-4}$$

$$n,k=0,1,\cdots.N-1$$

로써 영이 아닌 요소(non-zero element)로 구성된다. 다른 종류의 단일 변환과 같이 2×2 DCT kernel 행렬은

$$[A_2]=\frac{1}{2}\ \begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \tag{4-5}$$

이며, Chen은 식(4-5)로부터 $[A_N]$을 순환형(recursive form)으로 구하는 방법을 제시했으며 그 방법은 다음 식과 같다.

$$[A_N]=[P_N]\begin{bmatrix}A_{N/2} & 0\\ 0 & R_{N/2}\end{bmatrix} \tag{4-6}$$

$$[R_{N/2}]=[C(k)\ \cos\frac{(2n+1)(2k+1)\pi}{2N}\ ] \tag{4-7}$$

$$n,k=0,1,\cdots,N-1$$

$$[P_N]=\begin{bmatrix}1&0&0&\cdots&&0&0\\0&0&0&\cdots&&0&1\\0&1&0&\cdots&&0&0\\0&0&0&\cdots&&1&0\\\cdot&\cdot&\cdot&&\cdot&\cdot&\\0&0&0&\cdots&1\ 0\ \cdot&0&0\\1&0&0&\cdots&1\ 0\ \cdot&0&0\end{bmatrix} \tag{4-8}$$

$$[B_N]=\begin{bmatrix}I_{N/2} & \overline{I}_{N/2}\\ \overline{I}_{N/2} & -I_{N/2}\end{bmatrix} \tag{4-9}$$

단일 행렬 $[I_{N/2}]$로 구성된 $[B_N]$과 변환 계수 벡터 [F]를 BRO(bit reversed order)에서 NO(natural order)로 바뀌지는 순환 행렬(permutation matrix) $[P_N]$은 기지(known)의 행렬이고 또 $[R_{N/2}]$은 (2logN−3)개의 행렬로 $R_{N/2}$를 분해하여 0인 요소(element)를 많이 갖는 sparse matrix의 곱으로 만드는 일반적인 방법을 얻어냄으로써 구할 수가 있다. 따라서 식(4-5)에 의해 $[A_2]$를 알고 있으므로 $[A_N]$은 순환적으로 얻을 수 있다. 그림 4-2는 이 방법을 사용하여 N=8에 대한 순방향 신호 흐름도이다.

4-3. Lee의 고속 알고리즘

Lee의 고속 알고리즘은 FFT와 유사한 전개 방식을 사용하였으며 따라서 FFT 신호 흐름도와 유사하다. N개의 변환된 데이타의 IDCT 식(4-2)를 Lee 알고리즘의 계산 방법에 의해 서술하면 다음과 같다.

$$\hat{F}(k)=C(k)F(k) \tag{4-10}$$

$$G(k)=\hat{F}(2k) \tag{4-11a}$$

$$H(k)=\hat{F}(2k+1)+\hat{F}(2k-1) \tag{4-11b}$$

$$k=0,1,\cdots,\ N-1$$

$$g(n)=\sum_{n=0}^{N/2-1} G(k)\ C_{2(N/2)}^{(2n+1)k} \tag{4-12a}$$

$$h(n)=\sum_{n=0}^{N/2-1} H(k)\ C_{2(N/2)}^{(2n+1)k} \tag{4-12b}$$

$$k=0,\ 1\cdots,\ N-1$$

여기서,

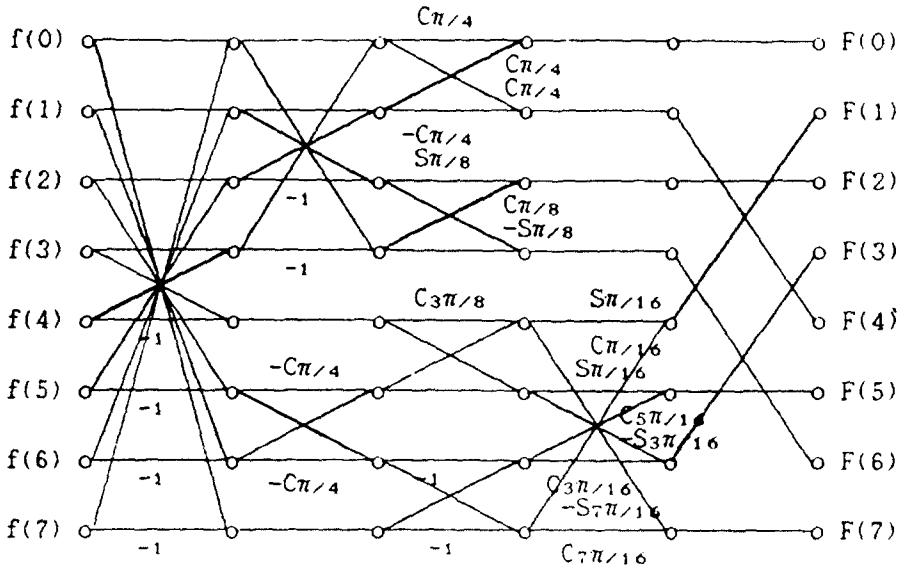


그림 4-2 Chen 알고리즘의 신호 흐름도

$$C_{2(N/2)}^{(2n+1)k}=\cos[\pi(2n+1)k/2N] \qquad (4\text{-}13)$$

이며, 따라서 식(4-2)의 IDCT는 아래와 같이 나타낼 수 있다.

$$f(n)=g(n)+(1/2C_{2N}^{(2n+1)})h(n) \qquad (4\text{-}14a)$$

$$f(N-1-n)=g(n)-(1/2C_{2N}^{(2n+1)})h(n) \qquad (4\text{-}14b)$$

위의 식은 식(4-2)의 N개의 변환 계수를 식(4-12)와 같이 N/2개 IDCT합으로 표시했으며 이 과정을 식(4-14)까지 반복함으로써 IDCT 값을 얻는 것을 보여준다. 그림 4-3은 위와 같은 방법을 사용하여 N=8에 대한 순방향 신호 흐름도를 나타낸다.

식(4-1)과 식(4-2)에서 보는 바와 같이 순방향 DCT의 kernel은 역방향 DCT의 그것과 같기 때문에 역방향 DCT의 신호 흐름도는 순방향 DCT의 그것을 뒤집어 놓은 형태를 갖게 된다.

4.4 제안한 알고리즘과의 비교

N개의 입력에 대한 DCT의 계수를 직접 계산, 2N점 FFT, Chen과 Lee등의 고속알고리즘, 그리고 2절에서 제안한 변형 DFT 방법에 따른 연산 횟수와 신호 흐름도를 비교해 보기로 하자.

DCT를 직접 계산할 경우 N×N 행렬과 N×1 벡터와의 곱셈이 되므로 N(N-1)번의 가산과 N^2번의 승산이 필요하고, 2N점 FFT를 사용하는 경우 복소 승산을 행하여야 하므로 ($6\log_2 2N+2N$)번의 실가산과 $4N(\log_2 2N+1)$번의 실승산이 필요하다. Chen의 고속 알고리즘을 사용 할 경우 가산횟수는 $3N/2(\log_2 N-1)+2$이고, 승산 횟수는 $N\log_2 N-3N/2+4$이며, Lee의 고속 알고리즘을 사용할 경우 가산횟수는 $3N/2\log_2 -N+1$이고, 승산횟수는 $N/2\log_2 N$이다. 그리고 본 논문에서 제안한 식(4-16)의 변형 DFT 알고리즘을 이용하는 경우 변형 DFT는 DFT와 유사하므로 $N(4N-2)+1$의 가산과 $4N^2+2$의 승산이 필요하다.

일반적으로 N이 증가할 수록 더욱 많은 화소

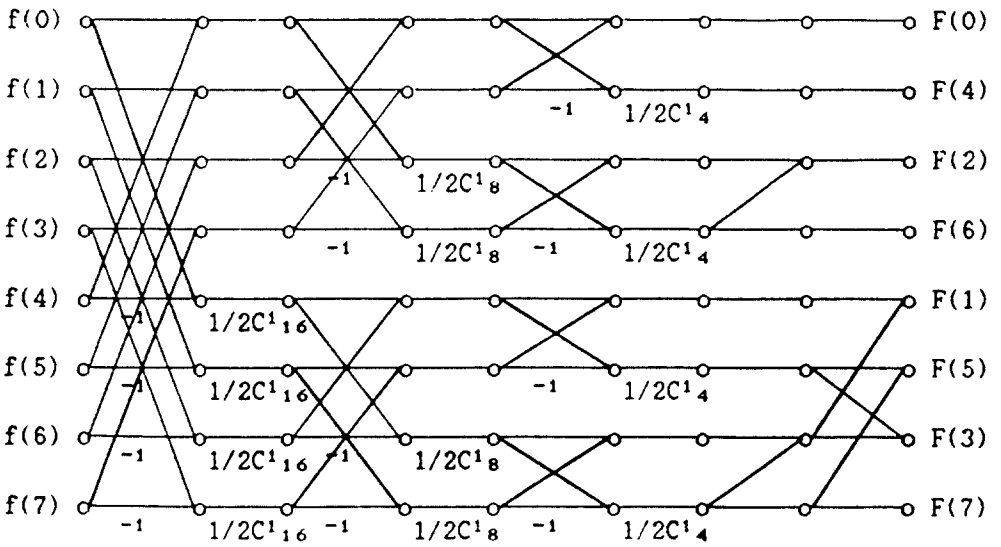


그림 4-3 Lee 알고리즘의 신호 흐름도

들의 상관성(correlation)이 고려되므로 변환의 성능은 좋아지지만, 실제의 경우 20개 정도의 서로 이웃한 화소들이 많은 상관성을 가지므로 N=16이상의 1차원 변환에 의한 성능 증가는 적다.[27]

그림 4-3과 그림 4-4의 고속 DCT 알고리즘의 신호 흐름도는 DFT의 신호 흐름도[24]와는 다르게 시스톨릭 어레이의 조건인 일정한 규칙성이 어긋남으로써 시스톨릭 어레이로 구현하기가 어려움을 알수 있다. 그러나 본 논문에서 제안한 변형 DFT는 연산 횟수와 에러면에서 고속 알고리즘에 비해 불리하지만 DFT와 유사하기 때문에 신호 흐름도 상에서 DFT와 같이 일정한 규칙성을 유지하므로 시스톨릭 어레이로 구현하기가 용이함을 알수 있다.

4.4.1 제안한 1차원 시스톨릭 어레이

가. 1차원 시스톨릭 알고리즘

입력 시퀀스 f(0), f(1),⋯,f(N−1)에 대한 N점 DFT는 다음 식과 같이 주어진다.[24]

$$F(k)=\sum_{n=0}^{N-1} f(n)\ W_N^{nk},\ k=0,1,\cdots,N-1 \quad (4\text{-}15)$$

여기서 $W_N=\exp\{-j2\pi/N\}$.

또한 윗 식과 같이 DCT를 DFT와 같은 형태로 표시하기 위하여 Chirp-z 변환[37]과 유사한 변형 DFT(MDFT : Modified Discrete Fourier Transform)를 다음 식과 같이 정의하자.

$$Z(k)\triangleq\sum_{n=0}^{N-1} f(n)\ U_N^{nk},\ k=0,1,\cdots,N-1 \quad (4\text{-}16)$$

여기서 $U_N=\exp\{j\pi/N\}$.

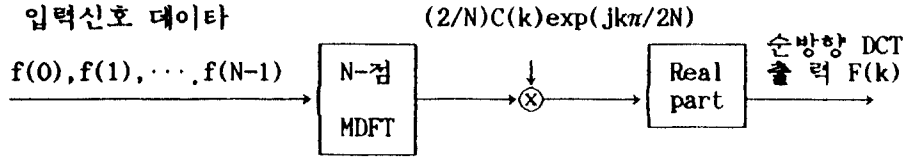
따라서 순방향 DCT 식(4-1)은 아래 식과 같이 MDFT를 이용하여 표현할 수 있다.

$$F(k)=(2/N)C(k)\ \mathrm{Re}\{\exp(jk\pi/2N)\cdot Z(k)\} \quad (4\text{-}17)$$

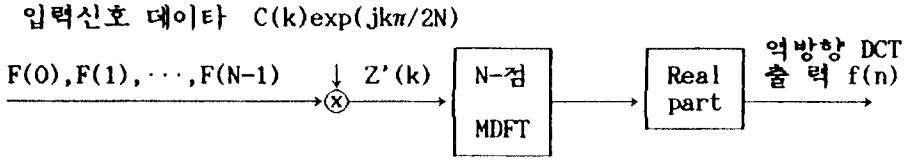
또한 역방향 DCT 역시 MDFT를 이용함으로써 다음 식과 같이 표현할 수 있다.

$$f(k)=\mathrm{Re}\{Z'(n)\} \quad (4\text{-}18)$$

단,



(a) Forward DCT computation



(b) Inverse DCT computation

그림 4-4 DCT 연산을 위한 블럭 다이아그램.

$$Z'(n) \triangleq \sum_{n=0}^{N-1} F'(k)\ U_N^{nk},\ n=0,1,\cdots,N-1 \qquad (4\text{-}19)$$

여기서 $F'(k)=C(k)\exp(jk\pi/2N)F(k)$이다. 식(4-7)과 식(4-18)로 부터 N점 DCT는 N점 MDFT로 실현할 수 있고, $C(k)\exp(jk\pi/2N)$에 의한 곱셈과 실수 부분을 취하는 연산이 필요하다. 그림 4-4는 DCT연산을 수행하는 전체적인 블럭 다이아그램을 보여 주고 있다.

식(4-16)의 MDFT는 Horner의 룰을 이용하여 아래 식과 같이 다항식으로 표시할 수 있다[7].

$$Z(k)=((\cdots(f(N-1)U_N^k+f(N-2)U_N^k+\cdots \cdots+f(2))U_N^k+f(1))U_N^k+f(0) \qquad (4\text{-}20)$$

연속적으로 데이타 시퀀스를 처리하기 위하여, 윗 다항식은 아래 식처럼 f (0)을 첫 항에, f(N-1)항을 마지막에 오도록 재배치 함으로써 정상적인 순서로 데이타를 처리할 수 있다.

$$A(k)=((\cdots(f(0)U_N^k+f(1))U_N^k+\cdots \cdots+f(N-2)U_N^k+f(N-1) \qquad (4\text{-}21)$$

또한, A(k)를 MDFT 출력 Z(k)로 표시하기 위하여 아래 식처럼 닫혀진 형태로 쓸수 있다.

$$A(k)=\sum_{n=0}^{N-1} f(n)\ U_N^{(N-n)k},\ k=0,1,\cdots,N-1 \qquad (4\text{-}22)$$

$U_N^{2N}=1$로 회전 인자(twiddle factor)의 주기성을 고려하여 윗 식을 간략히하면

$$A(k)=(-1)^k U_N^{\ k} \sum_{n=0}^{N-1} f(n)\ U_N^{\ nk} \qquad (4\text{-}23)$$

따라서, MDFT의 대칭성을 이용하여 정리하면

$$A(k)=(-1)^k\ U_N^{\ k}\ Z(-k) = \begin{cases} +U_N^{\ k}\ Z(-k), & k \text{ is even} \\ -U_N^{\ k}\ Z(-k), & k \text{ is odd} \end{cases} \qquad (4\text{-}24)$$

변수 Z와 A를 바꾸고 재배치하면, 입력 시퀀스 f(n)에 대한 MDFT의 출력은 아래 식과 같다.

$$Z(k)=(-1)^k\ U_N^{\ k}\ A(-k) \qquad (4\text{-}25)$$

위의 식을 고르첼 알고리즘(Goertzel algorithm) 이라 부르며, 식(4-25)를 N개의 처리요소로 구성되는 시스톨릭 어레이로 직접 실현하려

면 정확한 연산을 위해서 식(4-22)와 식(4-25)에서 보는 바와 같이 데이타 시퀀스 사이에 null 샘플이 필요하다.[36] 식(4-25)를 펼쳐보면

$$Z(k)=(-1)^k U_N^{-k}[(\cdots(f(0)U_N^{-k}+f(1))U_N^{-k}+\cdots+f(N-2))U_N^{-k}+f(N-1)] \quad (4\text{-}26)$$

한 번에 다항식의 한항을 곱하기 위하여 격리된 U_N^{-k}항을 괄호안으로 이동시키면

$$Z(k)=(-1)^k[(\cdots(f'(0)U_N^{-k}+f'(1))U_N^{-k}+\cdots\cdots+f'(N-2))U_N^{-k}+f'(N-1)] \quad (4\text{-}27)$$

여기서 $f'(n)=f(n)_N^{-k}$ 윗 식은 식(4-20)과 비슷하게 보이지만 데이타 샘플의 순서가 바뀌져 있고, 여기서는 회전 인자(twiddle factor)가 $U_N^{-k}=\exp\{-jk\pi/N\}$로 주어진다.

식(4-27)을 시스톨릭 어레이로 매핑하기 위하여, 선형 1차 순환식으로 표시하면

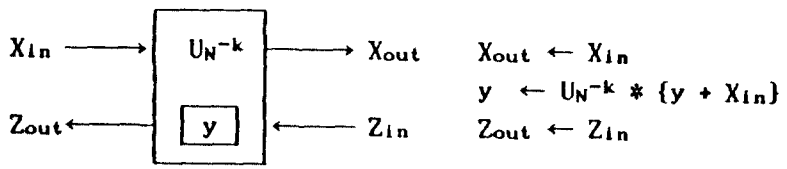
$$y(n,k)=U_N^{-k}\{f(n)+y(n-1,k)\} \quad (4\text{-}28)$$

여기서 y(n,k)는 n번째 순환 스텝과 k번째 MDFT 샘플을 지시하고, $0\leq k\leq N-1$, $0\leq n\leq N-1$ 이고 $y(-1,k)=0$. MDFT 샘플은 다음 식과 같이 N번의 순환 연산을 거친 후 얻을 수 있다.

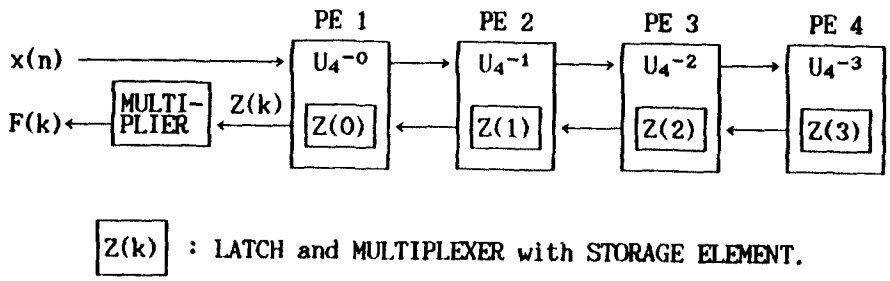
$$Z(k)=(-1)^k\, y(N-1,k),\ k=0,1,\cdots N-1 \quad (4\text{-}29)$$

한편, 고르첼 알고리즘에 의해 매핑(mapping)으로 얻어진 어레이는 모듈성, 국부 연결성, 단순한 처리요소등 VLSI설계 양립적(compatible)인 모든 성질을 포함하고 있다.[4]

그림 4-5는 N점 DFT를 수행하는 시스톨릭 어레이와 비슷한 형태이나, 곱셈기가 추가된 N점 DCT 연산을 수행하는 1차원 시스톨릭 어레이를 보여주고 있다. 그림 4-2의 (a)는 처리요소의 기본함수를 나타내고, (b)는 N=4일 때의 시스톨릭 어레이를 보여주고 있다. 각각의 처리요소는 한 개의 고정적인 회전인자 U_N^{-k}를 포함하고 있고, 중간 결과를 저장하기 위하여 저장요소(storage element)를 포함하고 있으며, 최종 결과를 출력하기 위하여 저장 능력이 있는 멀티프랙서(mulitiplexer)를 사용하고



(a) Basic function of PE



(b) Systolic array for N=4

그림 4-5 4점 DCT 연산을 위한 시스톨릭 어레이

표 4-2 4점 DCT의 시스톨릭 연산 과정

CLK.	MULT.	PE 1	PE 2	PE 3	PE4
1		$Z(0)=U_4^0\,a(0)$			
2		$Z(0)=U_4^0Z(0)+a(1)$;	$Z(1)=U_4^1a(0)$		
3		$Z(0)=U_4^0Z(0)+a(2)$;	$Z(1)=U_4^1Z(1)+a(1)$;	$Z(2)=U_4^2a(0)$	
4		$Z(0)=U_4^0Z(0)+a(3)$;	$Z(1)=U_4^1Z(1)+a(2)$;	$Z(2)=U_4^2Z(2)+a(1)$;	$Z(3)=U_4^3a(0)$
5		$Z(0)=U_4^0b(0)$	$Z(1)=U_4^1Z(1)+a(3)$;	$Z(2)=U_4^2Z(2)+a(2)$;	$Z(3)=U_4^3Z(3)+a(1)$;
6		$Z(0)=U_4^0Z(0)+b(1)$;	$Z(1)=U_4^1b(0)$	$Z(2)=U_4^2Z(2)+a(3)$;	$Z(3)=U_4^3Z(3)+a(2)$;
7		$Z(0)=U_4^0Z(0)+b(2)$;	$Z(1)=U_4^1Z(1)+b(1)$;	$Z(2)=U_4^2b(0)$	$Z(3)=U_4^3Z(3)+a(3)$;
8	A(0)	$Z(0)=U_4^0Z(0)+b(3)$;	$Z(1)=U_4^1Z(1)+b(2)$;	$Z(2)=U_4^2Z(2)+b(1)$;	$Z(3)=U_4^3b(0)$
9	A(1)	$Z(0)=U_4^0\,c(0)$	$Z(1)=U_4^1Z(1)+b(3)$;	$Z(2)=U_4^2Z(2)+b(2)$;	$Z(3)=U_4^3Z(3)+b(1)$;
10	A(2)	$Z(0)=U_4^0Z(0)+c(1)$;	$Z(1)=U_4^1c(0)$	$Z(2)=U_4^2Z(2)+b(3)$;	$Z(3)=U_4^3Z(3)+b(2)$;
11	A(3)	$Z(0)=U_4^0Z(0)+c(2)$;	$Z(1)=U_4^1Z(1)+c(1)$;	$Z(2)=U_4^2c(0)$	$Z(3)=U_4^3Z(3)+b(3)$;
12	B(0)	$Z(0)=U_4^0Z(0)+c(3)$;	$Z(1)=U_4^1Z(1)+c(2)$;	$Z(2)=U_4^2Z(2)+c(1)$;	$Z(3)=U_4^3c(0)$
13	B(1)	$Z(0)=U_4^0d(0)$	$Z(1)=U_4^1Z(1)+c(3)$;	$Z(2)=U_4^2Z(2)+c(2)$;	$Z(3)=U_4^3Z(3)+c(1)$;
14	B(2)	$Z(0)=U_4^0Z(0)+d(1)$;	$Z(1)=U_4^1d(0)$	$Z(2)=U_4^2Z(2)+c(3)$;	$Z(3)=U_4^3Z(3)+c(2)$;
15	B(3)	$Z(0)=U_4^0Z(0)+d(2)$;	$Z(1)=U_4^1Z(1)+d(1)$;	$Z(2)=U_4^2d(0)$	$Z(3)=U_4^3Z(3)+c(3)$;
16	C(0)	$Z(0)=U_4^0Z(0)+d(3)$;	$Z(1)=U_4^1Z(1)+d(2)$;	$Z(2)=U_4^2Z(2)+d(1)$;	$Z(3)=U_4^3d(0)$

여기서,

CLK. : Number of clock cycles

MULT. : Multiplier

MDFT samples : Z(i) denotes i-th MDFT sample

있다. 앞의 식(4-17)과 그림 4-4에서 보는 바와 같이, MDFT 출력은 왼쪽의 곱셈기에 의하여 $(2/N)C(k)\exp(jk\pi/2N)$가 곱해진다.

제안한 알고리즘을 적용하여 4점 MDFT를 시스톨릭 어레이로 계산하는 과정을 그림 4-6에 기술하였다. 각각의 처리요소는 한 개의 고정적인 회전인자, U_N^k를 포함하고 있으며 따라서 ROM을 사용할 수 있다. 입력 데이타 샘플들은 연속적인 형태로 어레이를 통과하고, 각각의 데이타 샘플은 한 개의 MDFT 샘플의 연산을 위해 처리요소에서 사용되며, 데이타 샘플들은 인접한 처리요소로 이동한다. 따라서 새로운 데이타 샘플은 한 개의 MDFT 샘플 연산을 위해 각각의 처리요소에서 이용이 가능하다. 그림 4-6에서 입력 데이타 시퀀스를 {a(0),a(1),a(2),a(3)}, {b(0),b(1),b(2),b(3)}, {c(0),c(1),c(2),c(3)}, {d(0),d(1),d(2),d(3)}라 하고 출력 MDFT 시퀀스를 {A(0),A(1),A(2),A(3)}, {B(0),B(1),B(2),B(3)}, {C(0),C(1),C(2),C

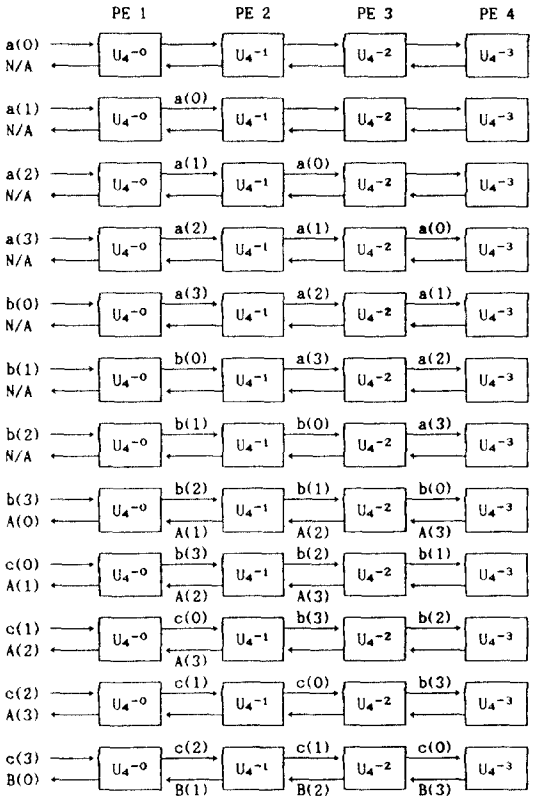


그림 4-6 4점 MDFT의 시스톨릭 연산 과정

(3)}라 가정하였다. 그림에서 2N−1 클럭 싸이클 후에 N점 DCT 샘플이 출력되고, 새로운 입력 데이타 시퀀스는 매 N 클럭 싸이클 동안에 연산되며 처리요소안의 플립플롭(flip-flop)은 매 N점 입력 샘플을 받은후 리셑된다. 모든 처리요소는 식(4-28)에서 기술된 것과 같이 한 개의 MDFT샘플의 중간 결과를 누산하면서 똑같은 동작을 수행한다. 표 4-2는 그림 4-6으로부터 4점 DCT를 연산하는 과정을 보여준다.

제안한 시스톨릭 어레이 매핑 알고리즘은 데이타를 운용하기가 쉽고, 처리요소안의 회전 인자만을 바꿔줌으로써 다른 직교 변환(orthogonal transform)을 쉽게 구현할 수 있다.

4.4.2 제안한 2차원 시스톨릭 어레이

가. 2차원 시스톨릭 알고리즘

간단한 입력 데이타의 재배열을 통해 N점 DFT로부터 N점 DCT를 연산하기 위하여 DCT의 정의식(4-1)에서 N이 짝수라 가정하고, 새로운 N점 시퀀스 x(n)을 아래와 같이 정의한다.[38]

$$\left.\begin{aligned} x(n)&=f(2n)\\ x(N-1-n)&=f(2n+1)\end{aligned}\right\}\ n=0,1,\cdots,N/2-1 \qquad (4\text{-}30)$$

위의 관계를 식(4-1)에 적용하면 정리하면

$$F(k)=\frac{2c(k)}{N}\left\{\sum_{n=0}^{(N/2)-1} x(n)\cos[(4n+1)k\pi/2N]\right.$$

$$\left.+\sum_{n=0}^{(N/2)-1} x(N-1-n)\cos[(4n+3)k\pi/2N]\right\} \qquad (4\text{-}31)$$

윗 식의 두번째 항에서 n'=N−1−n라 놓고, 두 항을 결합하고, 간략히 하면

$$F(k)=\frac{2c(k)}{N}\sum_{n=0}^{(N/2)-1} x(n)\cos[(4n+1)k\pi/2N] \qquad (4\text{-}32)$$

따라서 F(k)는 아래 식과 표현할 수 있다.

$$F(k)=(2/N)\,C(k)\,\mathrm{Re}\{\exp[jk\pi/2N]\cdot Z(k)\} \qquad (4\text{-}33)$$

여기서,

$$Z(k)=\sum_{n=0}^{N-1} x(n)\,W_N^{nk} \qquad (4\text{-}34)$$

Z(k)는 입력 데이타 시퀀스 x(n), n=0,1,⋯,N−1에 대한 역 DFT(IDFT)를 나타낸다.

또한 N은 두 정수의 곱으로 표시될 수 있다(즉 N=p·q). 1차원 데이타를 2차원 시스톨릭 어레이로 처리하기 위하여 STA2S(Systematic Transformation of Algorithm for 2-dimensional Systolic array)를 이용하여 아래와 같은 인덱스 매핑(Index mapping)을 정의한다.[39]

$$\left.\begin{aligned} rt&=pr+t\\ ij&=qi+j\end{aligned}\right\}\ rt,\ ij=0,1,\cdots,N-1 \qquad (4\text{-}35)$$

여기서 i,t=0,1,⋯,p-1이고, j,r=0,1,⋯,q-1이다. 위의 인덱스 매팅을 이용하면 식(4-33)과 식(4-34)는 각각 아래 식과 같이 쓸 수 있다.

$$F_{rt}=(2/N)\,C_{rt}\,\mathrm{Re}\{\exp[j(pr+t)/2N]\cdot Z_{rt}\} \qquad (4\text{-}36)$$

여기서,

$$Z_{rt}=\sum_{i=0}^{p-1}\sum_{j=0}^{q-1} x_{ij}W_N^{(pr+t)(qi+j)} \qquad (4\text{-}37)$$

$$rt=0,1,\cdots,N-1$$

여기서, i=0,1,⋯,q-1이고 각각의 r에 대해 t=0,1,⋯,p-1이다. 식(4-37)을 전개하면 정리하면

$$Z_{rt}=\sum_{i=0}^{p-1}\sum_{j=0}^{q-1} x_{ij}W_N^{pqri}\,W_N^{qti}\,W_N^{(pr+t)j}$$

$$=\sum_{i=0}^{p-1}\sum_{j=0}^{q-1} x_{ij}W_N^{qti}\,W_N^{(pr+t)j}\quad (W_N^{pq}=W_N^{N}=1)$$

$$=\sum_{j=0}^{q-1}\sum_{i=0}^{p-1} x_{ij}W_N^{-qti}\ W_N^{-(pr+tj)}$$

$$=\sum_{j=0}^{q-1} W_N^{-(pr+tj)}\sum_{i=0}^{p-1} x_{ij}W_N^{-qti}$$

$$=\sum_{j=0}^{q-1} W_N^{-prj}(W_N^{-tj}\ (\sum_{i=0}^{p-1} x_{ij}W_N^{-qti})) \qquad (4\text{-}38)$$

결과적으로 Z_{rt}를 구하기 위해서는 아래와 같이 세 개의 과정으로 분해 할 수 있다.

(1) $V_{tj}=\sum_{i=0}^{p-1} x_{ij}\ W_N^{-qti}$, $0\leqq t\leqq q-1$;p점 역 DFT
(2) $W_{tj}=W_N^{-tj}\ V_{tj}$;곱셈
(3) $Z_{rt}=\sum_{j=0}^{q-1} W_{tj}\ W_N^{prj}$, $0\leqq r\leqq p-1$;q점 역 DFT
(4-39)

식(4-39)의 V_{tj}와 Z_{rt}에 Horner의 룰을 적용하면 각각 다음 식과 같이 다항식으로 표시할 수 있다.

$$V_{tj}=((\cdots(x_{(p-1)j}W_N^{-qt}+x_{(p-2)j})W_N^{-qt}+\cdots$$
$$\cdots+x_{2j})W_N^{-qt}+x_{1j})W_N^{-qt}+x_{0j} \qquad (4\text{-}40)$$

$$Z_{rt}=((\cdots(W_{t(q-1)}W_N^{-pr}+W_{t(q-2)})W_N^{-pr}+\cdots$$
$$\cdots+W_{t2})W_N^{-pr}+W_{t1})W_N^{-pr}+W_{t0} \qquad (4\text{-}41)$$

2차원 시스톨릭 어레이로 매핑하기 위하여, 1차원에서의 방법과 똑같이 윗 식에 Goertzel 알고리즘을 적용하여 V_{tj}와 Z_{rt}를 다음 식과 같이 선형 1차 순환식으로 쓸 수 있다.

$$y_1(i,t)=W_N^{qt}\{x_{ij}+y_1(i-1,t)\} \qquad (4\text{-}42)$$
$$y_2(j,r)=W_N^{pr}\{x_{tj}+y_2(j-1,r)\} \qquad (4\text{-}43)$$

여기서, $y_1(i,t)$는 i번째 순환 스텝과 t번째 V_{tj} 샘플을, $y_2(j,r)$는 j번째 순환 스텝과 r번째 Z_{rt} 샘플을 나타내고, $0\leqq i,\ r\leqq p-1$ $0\leqq j,t\leqq q-1$ 이고, $y_1(-1,t)=0$, $y_2(-1,r)=0$이다. 그리고 V_{tj}와 Z_{rt} 샘플은 다음 식과 같이 각각 p번과 q번의 순환 연산후에 얻을 수 있다.

$$V_{tj}=y_1(p-1,t),\ 0\leqq t\leqq p-1 \qquad (4\text{-}44)$$

$$Z_{rt}=y_2(q-1,r),\ 0\leqq r\leqq q-1 \qquad (4\text{-}45)$$

결국, DCT F_{rt}를 계산하기 위해서는 아래와 같이 네 개의 과정으로 요약 할 수 있다.

(1) $V_{tj}=W_N^{qt}\{x_{tj}+y_1(p-1,t)\}$, $0\leqq t\leqq p-1$
(2) $W_{tj}=W_N^{tj}\ V_{tj}$
(3) $Z_{rt}=W_N^{pr}\ \{W_{tj}+y_2(q-1,r)\}$, $0\leqq r\leqq q-1$
(4) $F_{rt}=(2/N)C(k)\ Re\{\exp[j(pr+t)\pi/2N]\cdot Z_{rt}\}$ (4-46)

가. 2차원 시스톨릭 어레이

그림 4-7은 2차원 시스톨릭 어레이의 처리요소의 기본 기능을 보여주고 있으며, 2차원 어레이의 처리요소는 1차원 어레이의 처리요소 두 개의 합한 형태이므로 기본 구조도 1차원 어레이의 처리요소와 같다. 각각의 처리요소는 두개의 고성적인 회전인자 W_N^{qt}, W_N^{pr}를 포함하고 있고, 또한 어레이는 중간 결과를 저장하기 위하여 저장요소를 포함하고 있으므로, ROM을 사용할 수 있으며, 최종 결과를 출력하기 위하여 저장인자를 갖는 멀티플랙서를 사용하고 있다.

그림 4-8은 4점 DCT를 실현하기 위한 2차원 시스톨릭 어레이를 보여주고 있다. 처음에 2점 역 DFT가 처리요소의 윗 부분에서 병렬로 계산되고, 그 결과로 V_{tj}를 만들고, V_{tj}는 멀티플랙서를 통하여 어레이의 왼쪽으로 쉬프트하면서 출력된다. 왼쪽 윗 부분의 곱셈기에서는 입력 V_{tj}로부터 W_{tj}를 만들고, 다시 오른쪽으로 출력한다. 입력 W_{tj}에 대한 2점 역 DFT의 계산이 처리요소의 아랫 부분에서 병렬로 계산되고, 그 결과로 Z_{rt}를 어레이에 남겨 놓는다. 마지막으로 식(4-46)에서 언급한 바와 같이 왼쪽 아랫 부분의 곱셈기에서 $(2/N)C_{rt}\ \exp[j(2r+t)\pi/8]\cdot Z_{rt}$의 곱에 실수 부분만을 취함으로써 DCT를 얻게 된다.

다. 제안한 처리요소(PE)와 곱셈기의 구조

그림 4-9는 제안한 1차원 및 2차원 시스톨릭 어레이를 구성하는 처리요소(PE)의 구조를 보여

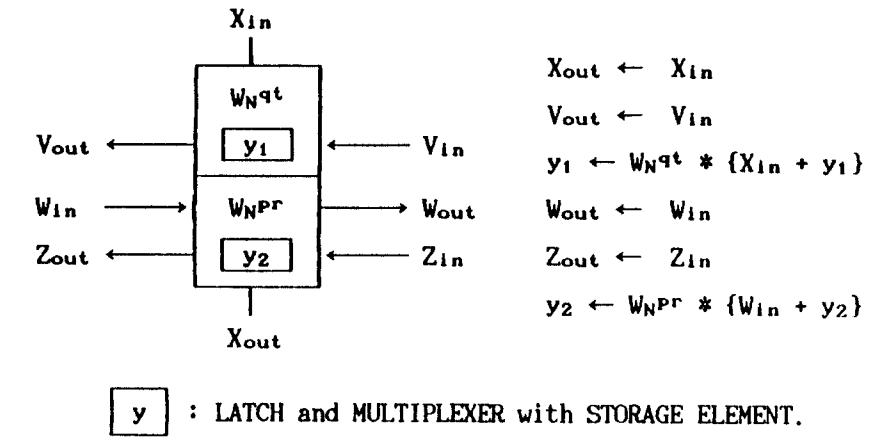


그림 4-7 2차원 시스톨릭 어레이의 처리요소의 기본 기능

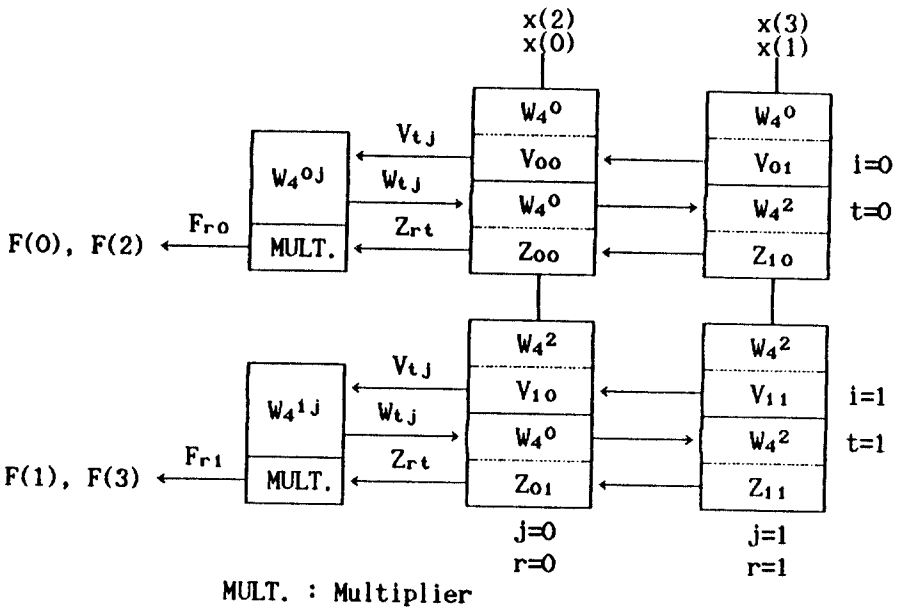


그림 4-8 4점 DCT의 2차원 시스톨릭 어레이

준다. Latch #1은 입력 데이타를 저장하고, latch #2와 latch #3은 똑같이 중간 MDFT 샘플 값을 저장하는데, 그중 latch #3은 최종 MDFT 샘플 값을 멀티플랙서를 통해 출력하기 위한 것이다. 모든 저장요소(latch #1, #2, #3)는 D형 플립 플롭(flip-flop)을 사용하고 그 중 latch #2는 클리어(clear)를 가진다.

그림에서 CLK1은 시스톨릭 어레이 전체를 동기시키는 신호이고, CLK2는 간단한 주파수 분해(÷N)로써 CLK1로부터 만들어지며 latch #2를 클리어 시키고 latch #3의 결과를 로드하는데 쓰인다. 즉, CLK2는 시스톨릭 어레이의 파이프라인 처리로 이용된다. MPX신호는 모든 처리요소에 보내지고, 저장요소를 갖는 멀티플랙

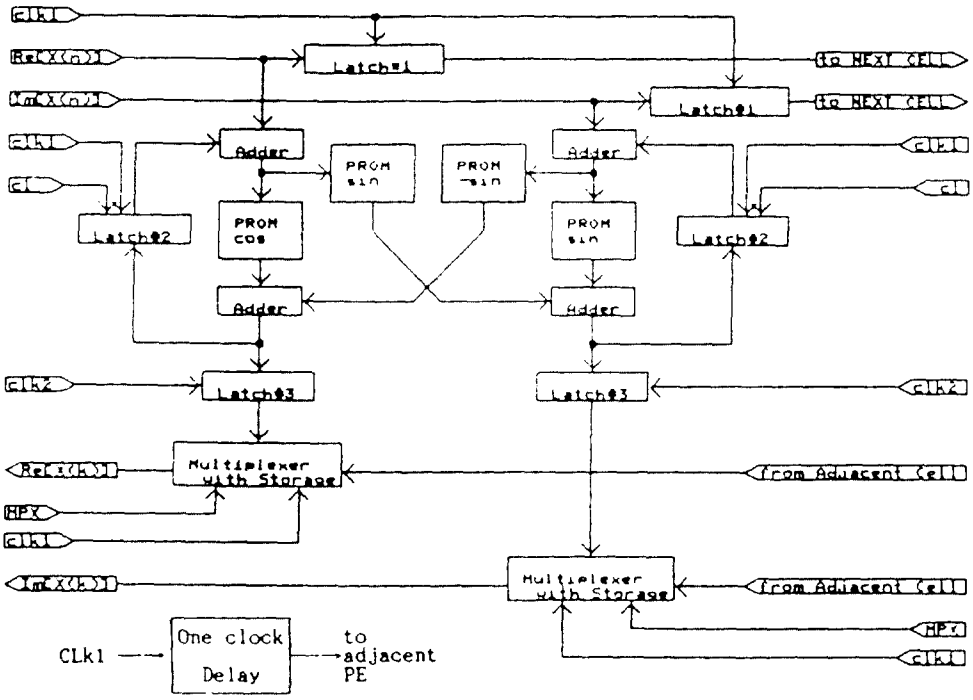


그림 4.9 처리요소의 구조

서를 통한 데이타의 방향을 조정한다.

각각의 처리요소 PE는 곱셈기대신에 연산의 오차를 줄이고 하드웨어의 복잡도를 감소시키기 위하여 ROM방식을 채택하였기 때문에, 곱해질 수 있는 모든 가능한 조합의 수를 미리 ROM에 입력시켜 놓고, 들어오는 입력값이 ROM의 주소를 지정하여 곱셈을 수행하였다.

일반적으로 DCT의 여난은 N=8 또는 N=16에서 그 성능이 가장 좋은 것으로 알려져 있고, 8비트 입력 영상 데이타에 대하여 N=16일때 최대 변환 계수(maximum transform coefficient) DC는 11비트가 되므로 데이타 패스는 적어도 12비트 이상이어야 한다. 따라서 처리요소 내부의 내부 정밀도(internal accuracy) 는 12비트로 제한하였다.

곱셈기 구조 역시 처리요소 구조와 동일하게 ROM방식을 가지고 단지 중간 결과를 저장하는 latch #2만이 제거된 형태이다.

5. 성능분석 및 컴퓨터 시뮬레이션 결과

본 장에서는 제안한 DCT 시스톨릭 어레이의 처리요소 구조의 throughput rate를 기존의 소자(components)를 이용하여 계산해 보았고, SNR을 해석해 보았으며, 비교적 구현하기 쉬운 정수형 연산으로 컴퓨터 시뮬레이션을 통해 제안한 시스톨릭 어레이의 성능을 NMSE와 주관적 화질에 의해 평가하였다.

5.1 성능 분석 및 SNR 해석

제안한 1차원 시스토릭 어레이 구조에서 MDFT 출력 Z(k)는 매 클럭마다 새로운 값이

어레이에 남아있게 된다.

제안한 1차원 시스톨릭 어레이 구조에서는 2N-1 클럭 싸이클 만큼의 초기지연이 필요하고, 매 N 클럭 싸이클동안 N점 DCT 샘플이 연산되며, 2차원 어레이의 처리요소는 1차원 처리요소를 두 개 합한 형태이므로, 하드웨어의 부담은 늘지만 〈계산상의 편의상 $p=q=\sqrt{N}$라 가정〉, 제안한 2차 원 시스톨릭 어레이의 경우에는 약 $2(\sqrt{N}+1)$ 클럭의 초기 지연이 필요하고 매 $\sqrt{N}$ 클럭 싸이클 동안 N점 DCT에 대한 연산이 완료됨으로써 연산 속도의 향상이 기대된다. 각 샘플에 대한 throughput비는 그림 3-9로 부터 플립 플롭(T_{FF}), 가산기(T_A), 그리고 ROM(T_{ROM})등에 의해 좌우된다. 따라서 소요되는 총 지연을 T_D라 하면

$$T_D=T_{FF}+2T_A+T_{ROM} \qquad (5\text{-}1)$$

여기서,

T_{FF}=Flip-flop access time
T_A=Adder arithmetic time
T_{ROM}=ROM access time

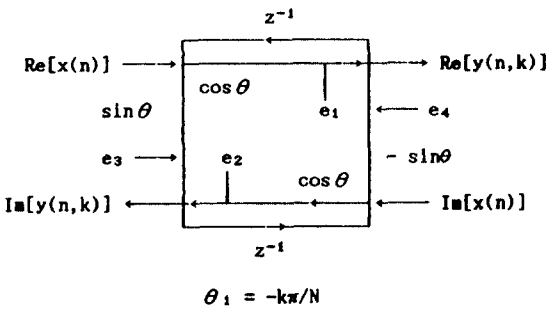
표 5-1에서 보는 바와 같이 처리요소를 TTL 소자를 사용하여 구성하였을 때, 비데오 신호를 처리할 수 있을 정도의 rate에 동작함을 알 수 있다.

신호대 잡음비 SNR(signal-to-noise ratio)은 아래 식과 같이 주어진다.[26]

$$SNR=\frac{\sigma_y^2}{\sigma_f^2} \qquad (5\text{-}2)$$

위에서 σ_y^2은 출력 신호의 분산을, σ_f^2은 출력 에러 신호의 분산을 나타낸다. 그림 5-1은 그림 3-9로 부터 제안한 처리요소(PE)와 곱셈기의확률 모델(statistical model)을 보여주고 있다.

단위 임펄스 응답(unit impulse response)은 다음과 같이 구할 수 있다. 그림 5-1(a)로부터, 잡음원 e_1-e_4에서 출력 Re[y(n,k)]와 Im[y(n,k)]로의 임펄스 응답[$h_e(n)$'s]을, 그리고 입력 신호 Re[x(n)]와 Im[x(n)]에서 출력 신호로의 응답[$g_x(n)$'s]을 구할 수 있다. 위의 임펄스 응답은 Mason의 룰을 적용함으로써 얻어진다.



(a) Processing element(PE)

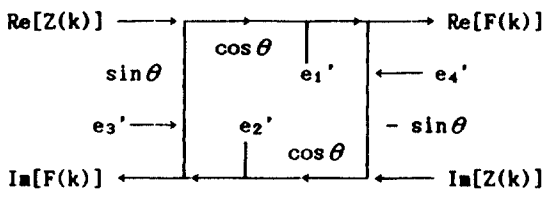
표 5-1 PE의 클럭 싸이클

(a) using VSC100 components · [40]

T_{FF}=4.2 ns
T_A=15.9 ns
T_{ROM}=96ns
Total T_D=96 ns
Clock cycles=10.4 MHz

(b) using TTL components

T_{FF}=5 ns
T_A=20 ns
T_{ROM}=25 ns
Total T_D=70 ns
Clock cycle=14.3 MHz



(b) Multiplier

그림 5-1 처리요소와 곱셈기의 확률 모델

$$Re[h_{e1}(n)]=\frac{Re[y(n,k)]}{e_1} \tag{5-3}$$

$$Re[H_{e1}(z)]=\frac{1-(z^{-1}\cos\theta_1)}{1-(z^{-1}\cos\theta_1+z^{-1}\cos\theta_1+z^{-2}\sin^2\theta_1)+z^{-2}\cos^2\theta_1}=\frac{1-z^{-1}\cos\theta_1}{1-2z^{-1}\cos\theta_1+z^{-2}} \tag{5-4}$$

윗 식에 역 z-변환을 취하면

$$Re[h_{e1}(n)]=Re[h_{e4}(n)]=\cos(n\theta_1)$$
$$Re[h_{e2}(n)]=Re[h_{e3}(n)]=-\sin(n\theta_1)$$
$$Im[h_{e1}(n)]=Im[h_{e4}(n)]=\sin(n\theta_1)$$
$$Im[h_{e2}(n)]=Im[h_{e3}(n)]=\cos(n\theta_1)$$
$$Re[g_{xr}(n)]=Im[g_{xi}(n)]=\frac{[\cos(\theta_1)\sin(n+1)\theta_1)-\sin(n\theta_1)]}{\sin(\theta_1)}$$
$$Re[g_{xi}(n)]=Im[g_{xr}(n)]=-\sin[(n+1)\theta_1] \tag{5-5}$$

여기서, $\theta_1=-k\pi/N$이고 $0\leq n,k\leq N-1$,

똑 같은 방법을 곱셈기에 적용시키면, 그림 5-1(b)로 부터 단위 임펄스 응답[he'(n)'s, gx(n)'s]을 구할 수 있다.

$$Re[g_{xi}'(n)]=-Im[g_{xr}'(n)]=-\sin(\theta_2) \tag{5-6}$$

여기서, $\theta_2=k\pi/2N$ 이고, $0\leq n,k\leq N-1$.
따라서 출력 에러 신호의 분산은

$$\sigma_f^2=\sigma_e^2\sum_{n=0}^{N-1}\{Re[h_{e1}(n)]^2=Re[h_{e2}(n)]^2+Re[h_{e3}(n)]^2+Re[h_{e4}(n)]^2\}+\sigma_e^2\{Re[h_{e1}'(n)]^2+Re[h_{e2}'(n)]^2+Re[h_{e3}'(n)]^2+Re[h_{e4}'(n)]^2\} \tag{5-7}$$

여기서, 내부 정밀도(internal accuracy)를 B bits(sign bit 포함)로 정한 경우 $\sigma_e^2=2^{-2B}/12$ 이므로,

$$\sigma_f^2=2^{-2B}(N+1)/6 \tag{5-8}$$

위의 결과는 실수부와 허수부 양쪽에 적용할 수 있다. 입력 신호에 가해지는 스케링 인자(scaling factor)는 식(5-9)과 같이 표현할 수 있다.

$$x_{max}<\frac{1}{\sum_{n=0}^{N-1}\{|Re[g_{xr}(n)]|+|Re[g_{xi}(n)]|\}+1\{|Re[g_{xr}(n)]|+|Re[g_{xi}(n)]|\}} \tag{5-9}$$

식(5-9)에 식(5-5)와 식(5-6)를 대입하면, 스케링 인자는 $1/(N+1)$이 된다. 이 값을 사용한 출력 신호의 분산은

$$\sigma_y^2=\sigma_x^2\sum_{n=0}^{N-1}\{Re[g_{xr}(i)]^2+Re[g_{xi}(i)]^2\}+\sigma_x^2\{Re[g_{xr}'(i)]^2+Re[g_{xi}'(i)]^2\} \tag{5-10}$$

식(5-10)에 식(5-5)와 식(5-6)를 대입하여 정리하면,

$$\sigma_y^2=\frac{1}{3(N+1)} \tag{5-11}$$

따라서 SNR은 식(5-12)과 같이 얻어진다.

$$SNR=2^{(2B+1)}/(N+1)^2 \qquad (5\text{-}12)$$

식(5-12)로 부터 제안한 어레이의 SNR은 DFT용 시스톨릭 어레이의 그것보다 약간 떨어짐을 알 수 있다.[36]

2차원의 시스톨릭 어레이에도 위와 동일한 방법을 적용함으로써 SNR을 해석하여 볼수 있다. 계산상의 편의상 $p=q=\sqrt{N}$ 인 직교형 어레이(Othogonal array)라 가정하면, SNR은 아래 식과 같이 표시된다.

$$SNR=2^{2B}/(2N+3\sqrt{N}+1) \qquad (5\text{-}13)$$

2차원의 어레이의 SNR은 승산 횟수의 감소로 인하여 1차원의 경우보다 약간 우세함을 알 수 있다.

5.2 컴퓨터 시뮬레이션 결과 및 검토

본 컴퓨터 시뮬레이션은 IBM PC/AT를 이용하였으며 실험 영상은 256×256개의 화소로 구성되고, 8비트의 해상도를 갖는 즉, 그레이 레벨(gray level)이 256인 Lenna영상과 Cronkite 영상을 이용하였다. 성능 평가의 기준으로는 NMSE(Normalized Mean Square Error)를 구했으며, 주관적 평가를 병행하였다. NMSE는 아래 식과같이 정의되며 단위는 dB이다.

$$NMSE=10\log_{10}=\frac{E[\{f(i,j)-f'(i,j)\}^2]}{E[f(i,j)^2]} \qquad (5\text{-}14)$$

이 때 f(i,j)는 각각 원 영상 및 재생 영상에서 i번째 주사선(rasterscan line)의 j번째 화소의 밝기(intensity)를 나타낸다. 그리고 주관적 평가란 원 영상과 재생된 영상을 실제로 보고 판단하는 방법을 말한다.

시뮬레이션은 C언어를 사용하였으며 각각의 처리 요소 PE의 ROM에 해당하는 부분을 미리 배열을 선언하여 저장하고 들어오는 입력 값을 배열의 좌표값으로 설정하여 수행하였다. 그러나 고정 소수점 연산시 발생하는 유한 길이 연산에 의한 성능 감소를 줄이고, cosine항과 sine항의 버림 오차(truncation error)를 감소시키기 위하여 round up 처리하였다.

그림 5-2는 N=8일때 1차원 시스톨릭 어레이를 이용한 원영상과 재생 영상을 보여주고 있

(a) Original 'Lenna' image

(b) Reconstructed 'Lenna' image

(c) Original 'Cronkite' image

(d) Reconstructed 'Cronkite' image

그림 5.2 원영상과 복원 영상의 비교

고, 각각의 경우에서의 NMSE는 Lenna 영상의 경우에 -36.2[dB]였고, Cronkite 영상의 경우는 -38.17[dB]정도였다. 실제 주관적으로 원 영상과 재생 영상의 차이를 구별하기는 어려웠다.

[6] 결 론

시스톨릭 어레이는 기존의 어레이 프로세서가 갖는 고속 연산의 장점외에 처리요소 사이의 규칙적이고 국부적인 데이타 교환, 그리고 단순한 데이타 및 제어 흐름등의 장점을 가지며 급속히 발달하고 있는 VLSI 기술지향적인 어레이 프로세서이다.

본 논문에서는 영상 신호의 압축에 널리 쓰이는 DCT를 실시간 처리하는 1차원, 2차원 시스톨릭 어레이를 제안하였다. 제안한 어레이는 변형 DFT와 역 DFT를 기초로 하고 있고, Kung 의 접근 방법에 고르첼 알고리즘을 적용하여 얻어졌다.

제안한 어레이의 처리요소는 내부에 고정적인 회전 인자를 갖고, 입력 데이타 시퀀스를 연속적으로 처리할 수 있으며, 계수 양자화에 따른 에러를 감소시키고 가격을 제한하기 위하여 곱셈기 대신에 ROM을 사용하였다. 1차원 어레이는 N점 DCT를 N 클럭 싸이클에 수행하고, 2차원의 경우는 하드웨어의 부담은 늘어나지만 1차원보다 연산 시간을 줄일 수 있고 SNR도 증가함을 알 수 있다. 제안한 구조는 처리요소의 회전 인자만을 바꿈으로써 다른 직교변환을 수행할 수 있는 융통성을 가지고 있다.

앞으로 본 논문에서 제안한 DCT 연산을 위한 시스톨릭 어레이의 처리요소 구조의 VLSI칩 설계에 대한 연구가 요구된다.

참 고 문 헌

1. N.Ahmed, T.Natarajan, K.R. Rao, "Discrete Cosine Transform," IEEE Trans. on Computers, vol.C-24, pp.90-93, Jan. 1974.
2. W.H. Chen, C.H. Smith and S.C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Trans. Commun. vol. COM-25,

No.9, pp.1004-1009, Sep. 1977.
3. B.G. Lee, "A New Algorithm to Computer the Discrete Cosine Transform," IEEE Trans. on ASSP., vol. ASSP-32, No.6, pp.1243-1245, Dec. 1984.
4. H.T. Kung, "Why systolic arrays," Computer, pp. 37-46, Jan. 1982.
5. H.T. Kung, "Special-Purose Device for Signal and Image Processing : An opportunity in VLSI," SPIE, Real Time Signal Processing III, vol. 241, pp.74-84, 1980.
6. D.S. Broomhead, et al, "A Practical Comparison of the Systolic and Wavefront Array Processing Architectures," Proceesdings, IEEE Workshop on VLSI Signal Processing, Los Angeles, Nov. 1984.
7. S.Y. Kung, "On Supercomputing with Systolic / Wavefront Array Processor," Proceedings of IEEE vo.72, pp.867-884, July 1984.
8. Special issue on "Systolic Arrays," Computer, pp.12-103, July 1987.
9. M.J. Foster, H.T. Kung, "The Design of Special Purpose VLSI Chips," Computer, pp.26-40, Jan. 1980.
10. H.T. Kung, and C.E. Leiserson, "Systolic Arrarys (for VLSI)," Sparse Matrix Proc. 1978, 1979, Academic Press.
11. L.W. Chang and M.Y. Chen, "A New Systolic Array for Discrete Fourier Transform," IEEE Trans. on ASSP, vol.36, No.10, Oct. 1988.
12. J.S. Ward, B.J. Stanier, "Fast Discrete Cosine Transform Algoritm for Systolic Arrays," Electronics Letters, vol.19, No.2, Jan. 1983.
13. A.K. Jain, "Image Data compression: A Review," Proc. IEEE, vol.69, No.3, pp.349-389, March 1981
14. A.Habibi, "Hybrid Coding of Pictorial Data," IEEE Trans. Communi., vol.COM-22, No.5, pp.614-624, May 1974.
15. J.B. O'Neal, "Predictive quantizing System for the Transmission of Television Signals," BSTJ, vol.45, pp.689-721, May-June 1966.
16. B.Girod, "Adaptive Prediction for DPCM Coding of TV Signals," IEEE Trans. on ASSP, vol.29, No.6, pp.1142-1147, Dec. 1981.
17. P.A. Wintz, "Transform Picture Coding" Proc. IEEE, vol.60, No.7, pp.809-820, July 1972.
18. R.J.Clark, Transform Coding of Images, Academic Press, 1985.
19. H.Harmuth, Transmission of Information by Orthogonal Functions, 2nd ed. New York, Heideberg, Berlin: Springer, 1972, Chap.1.
20. N.Ahmed, K.R.Rao, Orthogonal Transforms for Digital Signal Processing. New York, Heideberg, Berlin: Springer, 1975.
21. A.K. Jain, "A Fast Karhunen-Loeve Transform for A Class of Random Process," IEEE Trans. Commun., vol.COM-24, pp.1023-1029, 1976.
22. J.J.Y.Huang and P.M.Schultheiss, "Block Quantization of Correlated Gaussian Random Variables," IEEE Trans. Commun., Syst., vol.CS-11, pp.289-296, Sep. 1963.
23. H.J.Landau and D.Slepian, "Some Computer Experiments in Picture Processing for Bandwidth Reduction," BSTJ, vol.50, pp.1525-1540, May-June 1971.
24. W.T.Cochran, et al., "What is Fast Fourier Transform," Proc.IEEE 55, pp.1664-1674, 1967.
25. R.C.Gonzales, P.Wintz, Digital Image Processing, Addison-Wesley Publishing Company, Inc., 1977.
26. A.V.Oppenheim and R.W.Shafer, Digital Signal Processing, Englewood Cliffs, NJ:Prentice-Hall 1975.
27. A.Habibi and P.A.Wintz, "Image Coding by Linear Transformation and Block Quantization," IEEE Trans. Commun. Tech., vol.COM-19, pp.50-63, Feb. 1971.
28. S.Y.Kung, H.J.Whitehouse, T.Kailath, "VLSI and Modern Signal Processing," Prentice-Hall Inc. 1985.
29. S.Y.Kung, "VLSI Array Processors," IEEE ASSP magazine, pp.4-22, July 1985.
30. L.S. Haynes, R.L. Lau, D.P.Siewiorek, D.W.Mizell, "A survey of highly parallel computing," IEEE Computer, pp.9-24, Jan. 1982.
31. H.M.Ahmed, J.M.Delosme, M.Morf, "Highly Concurrent Computing Structures for Matrix and Signal Processing", IEEE Computer, pp.65-82, Jan. 1982
32. C.A.Mead, L.A.Conway, "Introduction to VLSI System," Addison-Wesley, Reading, Mass., 1980.
33. Jos'e A.B.Fortes, Benjamin W.Wah, "Systolic Arrays-From Concept to Implementation," Computer, pp.12-17, July 1987.

34. H.T.Kung, "Notes on VLSI computaion", Department of Computer Science, Carnegie-Mellon Univ. Sep. 1980.
35. C.E.Leiseron, "Area-Efficient VLSI Computation," The MIT Press. 1983.
36. J.A. Belaldin, T.Aboulnasr, and W.Steenaart, "Efficient One dimensional Systolic Array Realization of the Discrete Fourier Transform," IEEE Trans. on CAS, vol.36, No.1, p.95 100, Jan. 1989.
37. Leo I. Bluestein, "A Linear filtering Approch to the Computation of Discrete Fourier Transform," IEEE Trans. on Audio and Electroaccoustics, vol .AU-18, No.4, pp.451-455, Dec. 1970.
38. M.J.Narasimha and A.M. Peterson, "On the Computation of the Discrete Cosine Transform," IEEE Trans. on Comm., vol.COM-26, pp.934-936, 1978.
39. N.L.Gand and M.A.Bayoumi, "Algorithms for High Speed Multi-Dimensional Arithmetic and DSP Systolic Arrays," Proc. of the 1988 International Conference on Parallel Processing, Aug. 15-19, 1988.
40. VSC100 Potable Library, VLSI Technology, Inc.
41. 이문호, THE WEIGHTED HADAMARD / DISCRETE COSINE TRANSFORM FOR IMAGE CODING AND THEIR SYSTOLIC ARRAY PROCESSING 논서출판 해외 마케팅, 1990
42. K.R.Rao, P.Yip, Discrete Cosine Transform, Algorithms, Advantages, Applications Academic press, 1990.

李 門 浩

저자약력

- 일본동경대 전자과(공박)
- 미국 미네소타 주립대 전기과 포스트닥터
- 독일 하노버대학 전자과 초청교수
- 전기통신 기술사(1982) 및 무선설비기사 1급
- 1970～1980 : 남양 MBC 송신소장
- 1980～현재 : 전북대 정보통신학과 교수

朴 鍾 五

저자약력

- 전북대 전자공학과(학사)
- 전북대 대학원 전자공학과(석사)
- (주)태산성밀 연구위(현재)

朴 周 用

저자약력

- 1982 : 전북대 전자공학과(학사)
- 1986 : 전북대 전자공학과(석사)
- 1988～1990 : 전북대 전자공학과 박사과정중
- 1988～현재 : 전북대 정보통신공학과 조교

李 光 宰

저자약력

- 1986 : 전북대 전자공학과(학사)
- 1990 : 전북대 대학원 전자과(석사)
- 1987～현재 : 한국방송공사 전주총국 기술국