特輯

HDTV용 VLSI 시스템 구현

李 商 郁, 金 鱗 澈

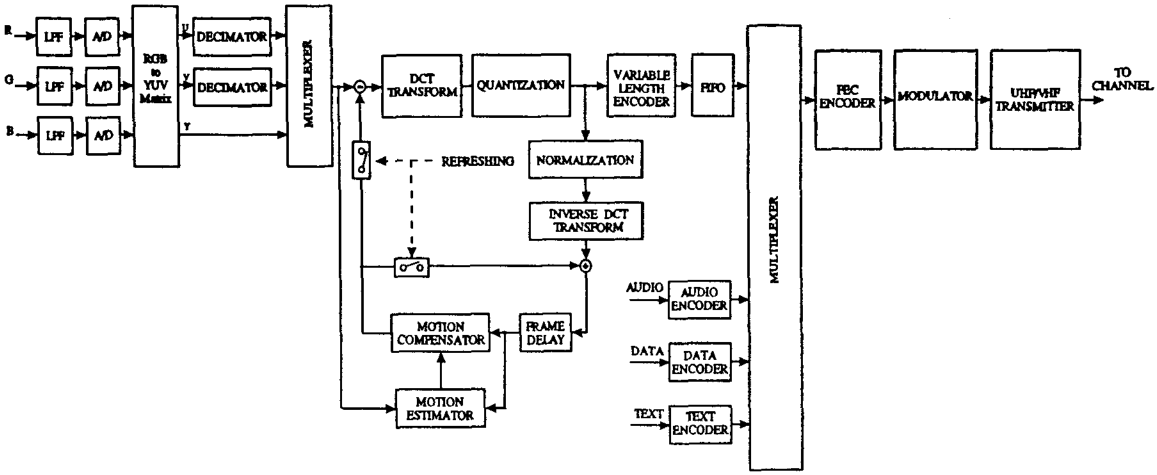
서울大學校 制御計測工學科, 大宇電子 映像硏究所

I. 서 론

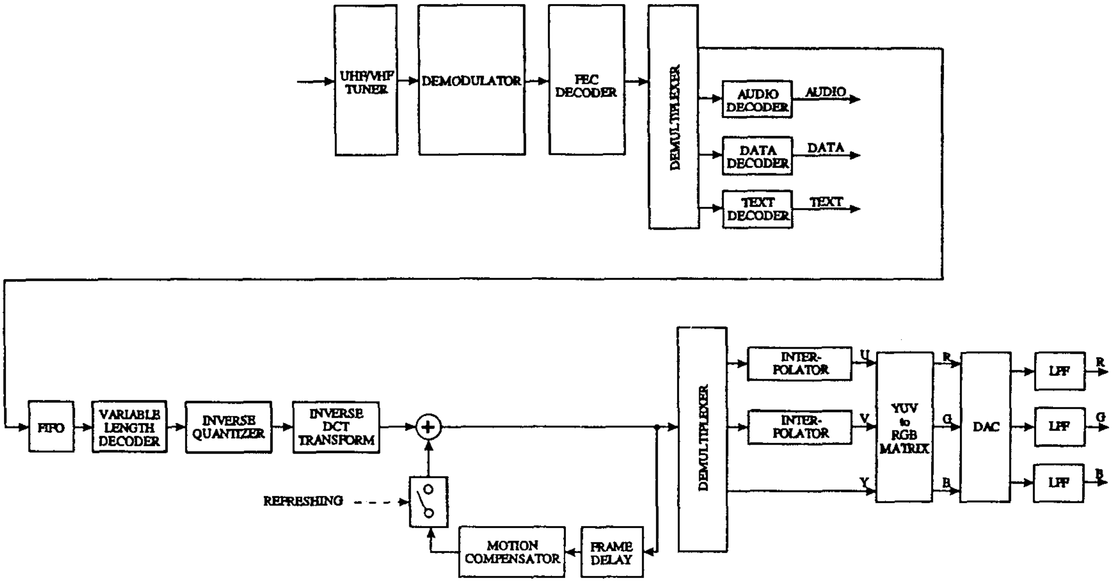
TV는 현대 사회에서 정보의 다양성, 정보의 품질 및 이해도, 정보 파급 효과 등의 측면에서 매우 우수한 대중 정보 전달 수단이므로 그 중요성이 높아지고 있다. 그러나, 현대 사회가 급격하게 발전함에 따라 보다 다양한 정보, 보다 고품질의 정보에 대한 욕구가 크게 증가하고 있으나, 기존의 TV로는 이러한 수요를 충분히 충족시킬 수 없다는 단점이 있다. 따라서, 기존의 TV의 성능을 크게 향상시킬 수 있는 고화질 TV, 혹은 HDTV(high-definition TV)[1-5]에 대한 연구가 활발히 전개되어, 일본에서는 현재 시험 방송을 하고 있고, 향후 수년내로 미국, 유럽 뿐만 아니라 우리나라에서도 HDTV 방송을 할 계획으로 있다. 현재 개발 완료 혹은 개발중에 있는 HDTV는 크게 아날로그 방식과 디지탈 방식으로 나눌 수 있다. 그 중에서 디지탈 방식의 HDTV[3-5]는 영상 정보를 전적으로 디지탈로 처리함으로써, 정보 처리가 간편할 뿐만 아니라 암호화에 유리하고 전송 범위(coverage area) 측면에서도 우수한 특성을 보인다. 따라서 디지탈 방식의 HDTV는 정상 정보의 신뢰도를 향상시킴과 동시에 데이타 압축 능력을 크게 높일 수 있다는 장점이 있기 때문에 HDTV에 적합한 방법으로 평가 받고 있다.

현재 디지탈 HDTV는 전세계적으로 그 규격이 확정되지 않았기 때문에 디지탈 HDTV의 세부적인 기능들을 정확하게 정의할 수는 없는 실정이다. 그러나, 현재 연구 개발되고 있는 디지탈 HDTV 시스템은 영상 부호화 기법으로 대부분 움직임 보상, discrete cosine transform (DCT) 그리고 variable-length coding (VLC) 등을 복합적으로 사용하는 움직임 보상 변환 부호화 기법[1-2]을 채택하고 있다. 따라서, 본 고에서는 움직임 보상 변환 부호화 기법을 근간으로 하는, 그림 1에 제시된 바와 같은 구조를 가지는 디지탈 HDTV 시스템을 중심으로 살펴 보겠다. 먼저, 그림 1(a)에 제시된 부호화부(encoder)를 살펴보면, RGB 형태로 표현된 입력 영상 신호는 A/D 변환을 거친 후, YUV 형태의 영상 신호로 변환하고, 색차 신호에 대해서는 간축(decimation)을 통하여 그 크기를 축소시킨다. 그런 다음, 움직임 추정기(motion estimator)를 사용하여 현재 프레임의 영상 신호와 이전 프레임의 영상간의 움직임을 추정하여 그 차이를 얻는 움직임 추정 및 보상(motion estimation and compensation)을 수행한다. 이때 얻어지는 움직임 보상 오차 신호는 DCT된 다음, 양자화되고 VLC를 통하여 가변 길이의 부호로 변환된다. 양자화된 신호는 역 양자화, inverse DCT(IDCT) 및 inverse motion compensation 과정을 거친 후 프레임 메모리에 저장되어, 다음 프레임 영상에 대한 움직임 추정에 사용된다. 한편, 가변 길이 부호로 변환된 영상 신호는 multiplex에서 음성, 데이타 및 text 등의 정보들과 조합된 다음, Reed-Solomon code로 대변되는 forward error correction(FEC) encoder, 변조기(modulator) 및 UHF/VHF 송신기등을 거친 다음, 채널을 통하여 복호화기(decoder)로 전송된다. 그림 1(b)에 제시된 복호화기에서는 부호화기의 역순으로 수신 신호가 복호화되어 복원된 영상 신호를 얻는다.

그림 1과 같은 구조를 가지는 디지탈 HDTV에 관한 연구 가운데 가장 중요한 부분 중의 하나는 HDTV 시스템의 실시간 구현에 관한 것이다. 그러나, HDTV는 그 정보량이 방대하므로 이를 실시간으로



(a)부호화기



(b)복호화기

그림 1. 디지탈 방식의 HDTV 시스템의 간략화된 블럭선도

구현하기 위해서는 고도의 병렬 및 pipeline 연산을 수행하는 전용 VLSI chip, 혹은 application specific intergrated circuit(ASIC)들을 사용하는 것이 필수적인 것으로 판단된다. 영상 부호화 기법을 ASIC들을 이용하여 시스템을 구성한 예는 LSI Logic사의 JPEG chip set[6] 과 H.261을 위한 chip set [7,8], 그리고 [9]에서 소개하는 H.261을 위한 chip set 등이 있다. 그러나, 이러한 예들은 비교적 정보량이 적은 video-phone에 응용하기 위한 시스템들이다. 따라서, 이들을 이용하여 HDTV 시스템을 구현할 때에는 HDTV에서 요구되는 처리 속도를 맞추기 위하여 위에서 언급한 chip set들을 여러개를 사용하여 병렬로 처

리하여야 하기 때문에 시스템이 대단히 복잡해진다는 단점이 있다. 따라서, HDTV 시스템을 보다 효율적으로 구현하기 위해서는 HDTV를 위한 영상 부호화 기법의 처리 속도에 적합한 새로운 전용 ASIC들을 개발하는 것이 절실하다. 이러한 ASIC 들과 더불어 HDTV에 필요한 VLSI 소자들은 향후 반도체 시장의 60~70%를 점유할 것으로 보이는 바, 관련 분야의 발전에 선도적인 역할을 할 것으로 예상되고 있다.

이러한 점들을 참조할 때, 그림 1에 제시된 HDTV 시스템의 각 요소들을 ASIC화 하는 기술은 현시점에서 매우 중요하다고 판단된다. 따라서 본 고에서는 각 요소들에 대한 ASIC 관련 최근 연구 동향을 살펴보고, HDTV를 구현하는데 필수적인 요소들에 대한 ASIC 기법들에 대하여 고찰한다. 일반적으로 그림 1에 제시한 바와 같은 HDTV 시스템의 각 요소들은 방대한 양의 정보들을 처리하기 때문에 빠른 처리 속도를 요구하나, 각 요소들은 비교적 간단한 연산의 반복으로 이루어져 있다. 이러한 경우에, 각 요소들에 대한 VLSI 구조는 지난 10년간 활발히 연구되어져 온 systolic/wavefront array[10,11] 개념을 사용하여 효율적으로 설계할 수 있다. Systolic/wavefront array는 간단한 연산의 반복으로 이루어진 기법들을 병렬 및 pipeline 연산을 통해 처리 속도를 크게 향상시키면서, VLSI chip으로의 설계가 용이하도록 시스템을 구현하는 기법이다. 이 기법은 행렬 연산 및 벡터 내적(inner product) 연산 등과 같은 기본적인 신호 처리에 관한 제 알고리듬들의 처리 속도를 크게 향상시킬 수 있는 방법으로 잘 알려져 있다[10,11]. 이러한 점들을 참조하여, 본 고에서는 HDTV 시스템을 보다 효율적으로 구현하는 방법으로 systolic/wavefront array의 개념을 사용하여 각 요소들을 ASIC화 하는 기법들을 중심으로 살펴보겠다.

본 고의 구성은 다음과 같다. 서론에 이어, 제 2 장에서는 그림 1에 제시한 HDTV 시스템에서 핵심적인 요소들인 움직임 추정기, DCT/IDCT, VLC 및 variable length decoder(VLD), FEC 기법으로 널리 응용되는 Reed-Solomon coder, 그리고 FEC 및 변복조 기법으로 널리 응용되는 trellis coded modulation(TCM) 기법에 사용되는 Vitervi decoder 등 5 요소들의 ASIC 기법들에 대하여 고찰하였다. 그리고 제 3 장에서는 본 고의 결론과 함께 예상되는 연구 방향에 대하여 서술하였다.

Ⅱ. VLSI 구현 기법들

1. 움직임 추정기

영상 신호의 프레임간 잉여 정보(redundency)를 제거하기 위한 움직임 추정 및 보상을 위해 가장 널리 사용되는 방법은 block matching algorithm(BMA)[1,2]이다. BMA는 현재 프레임의 영상을 작은 크기의 블럭으로 나눈 다음, 각 블럭 단위로 이전 프레임의 영상내에서 가장 유사한 블럭을 찾는 것이다. BMA에서 블럭간의 유사성은 다음과 같이 정의되는 mean absolute error(MAE)에 의해 측정된다.

$$MAE(m,n)=\sum_{N-1}^{N-1}\sum_{N-1}^{N-1} |a(i,j)-b(i+m, j+n)| \quad (1)$$

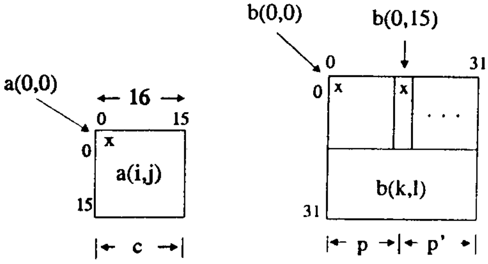
여기서 a(i,j) 는 N×N 크기의 입력 블럭 신호를 나타내고, b(i,j)는 이전 프레임 영상 신호를 나타낸다. 그리고, (m,n)은 예상 움직임 벡터를 나타낸다. 실제적으로 움직임 추정은 예상 움직임 벡터의 범위, 즉 검색 범위(search area)를 P×P 로 제한하여 현재 영상 신호를 이전 프레임내의 영상 신호와 비교하여 최소의 MAE를 가지는 벡터를 최적의 움직임 벡터로 추정한다. 즉, BMA 연산은 다음과 같이 4 개의 loop 로 나타내어지게 된다.

```
For m=-P/2, P/2-1
    For n=-P/2, P/2-1
        For i=0, N-1
            For j=0, N-1
              MAE(m,n)=MAE(m,n)+|a(i,j)
                        -b(i+m,j+n)|    (2)
            End(j)
        End(i)
    End(n)
End(m)
```

식 (2)로 나타내어지는 BMA의 연산은 N×N 크기의 2 개의 블럭간의 MAE를 예상 움직임 벡터의 갯수만큼, 즉 P×P 회 반복하는 것으로 이루어져 있음을 알 수 있다. 여기서 각 예상 움직임 벡터에 대한 MAE 연산은 전형적인 벡터 내적 연산과 유사하게 이루어진다. 이러한 벡터 내적 연산에 대한 VLSI 시스템은 systolic/wavefront array[10,11]등을 이용한 일반적인 전용 VLSI 시스템 구현 기법에서 심도있게 연구되어졌기 때문에, 많은 방식의 BMA를 위한 VLSI 구조 및 시스템들이 소개되고 있다[12-16]. BMA를 위한

VLSI 시스템을 설계할 때에는 처리속도, 복잡도, I/O bandwidth, 그리고 PE 간의 통신 등 일반적으로 VLSI 시스템에 대한 고려 사항[10-11]외에 연속적인 MAE 계산시의 효율성을 고려하여야 한다[15]. 식 (2)를 살펴보면, 예상 움직임 벡터에 대한 MAE를 계산할 때 입력 블럭 {a(i,j)} 는 고정되어 있으나, 이전 프레임의 영상들로 이루어지는 기준 블럭(reference block) {b(i, j)} 는 예상 움직임 벡터의 크기에 따라 좌우, 혹은 상하로 1 화소씩 이동시키면서 MAE 들을 계산할 수 있음을 알 수 있다. 이때, 인접한 예상 움직임 벡터들에 대한 기준 블럭 신호들은 거의 대부분 중복적으로 사용됨을 알 수 있는데, 따라서 이러한 중복되는 기준 블럭 신호를 공유할 수 있는 방법에 관한 것이 BMA의 효율적인 VLSI 구현에 핵심적인 요소중의 하나이다.

기존의 BMA를 위한 VLSI 구조들 중에서 가장 효율적인 기법중의 하나는 그림 2에 제시한 바와 같은 Yang[15]등이 제안한 구조이다. 그림 2는 입력 블럭의 크기가 16×16 이고, 검색 영역 또한 16×16 일 때의 VLSI 구조인데, 3 개의 입력 단자를 통하여 입력 블럭 신호 {a(i,j)} 및 기준 블럭 신호 {b(i,j)}를 받아들인다. 즉, 그림 3(a)에 도시한 바와 같이 입력 블럭 신호 {a(i, j)} 를 a(0,0) 화소부터 a(0,15)을 거쳐 a(15,15)까지 순차적으로 C 단자를 통해 입력한다. 그리고, 기준 블럭 신호들은 그림 3(b)에 제시한 바와 같이 2 부분으로 나누어져, 왼쪽 부분의 기준 블럭 신호는 p 단자를 통하여 b(0,0) 신호부터 입력되며, 오른쪽 부분의 기준 블럭 신호는 p' 단자를 통하여 b(0,15) 신호부터 p 단자 신호보다 16 clocks 지연되어 입력된다. 입력된 신호들은 그림 2에 도시된 바와 같이 Mux 및 delay 소자들을 통해 각 PE들에게 입력되어, 각 PE에서는 특정한 예상 움직임 벡터에 대한 MAE를 계산하게 된다. 이때, PE의 구조는 그림 4에 도시한 바와 같이 절대차들의 누적합을 구하는 구조로 되어 있다. 그림 2에 제시된 구조에서 p개의 예상 움직임 벡터에 대한 MAE 들을 계산할 때, 각 PE에서 데이타 입력 및 데이타 처리 흐름도는 그림 5에 제시한 바와 같다. 그림 5의 흐름도를 참조하면, 그림 5의 VLSI 구조를 사용하여 데이타들을 연속적으로 입력시킬 때, 식 (2)의 BMA 연산을 약 4111 clocks 만에 수행시킬 수 있음을 알 수 있다.



(a)현재블럭 (b)기준블럭

그림 3. 움직임 추정시 현재 블럭 및 기준 블럭의 양식

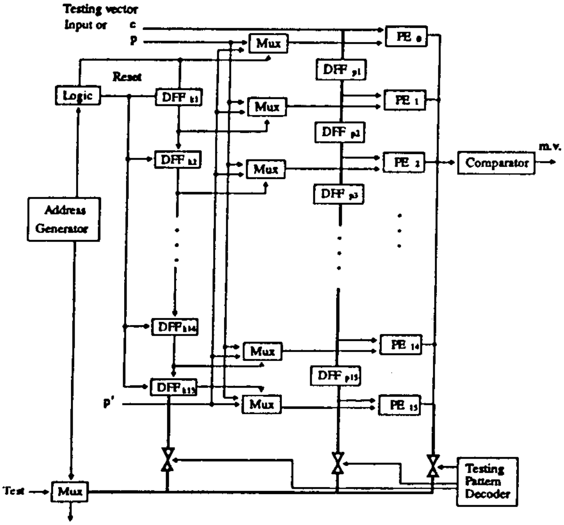


그림 2. 움직임 추정기의 VLSI 구현 예

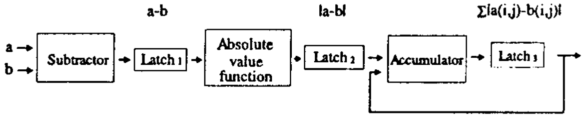


그림 4. 움직임 추정기에 사용되는 PE의 구조

그림 2에 제시된 BMA에 대한 VLSI 시스템의 장점은 3 개의 입력 단자 및 1개의 움직임 벡터 출력등으로 입출력이 구성되어 I/O bandwidth가 비교적 좁다는 것이다. 그리고 16개의 PE 들에 대하여 효율적으로 입력 데이타를 전달하여 연산을 수행하게 함으로써 처리 시간 및 복잡도 측면에서 유리하다. 그러나 기준 블럭 신호 입력들이 모든 PE 들에게 동시에 전파되어야 한다는 단점이 있다. 그림 2의 구조를 가지는 움직임 추정

Cycle time	Data sequences			PE_0	PE_1	PE_2	PE_{14}	PE_{15}
t	c	p	p′	$\Sigma\Sigma\|a(i,j)-b(k,l)\|$	$\Sigma\Sigma\|a(i,j)-b(k,l+1)\|$	$\Sigma\Sigma\|a(i,j)-b(k,l+2)\|$	$\Sigma\Sigma\|a(i,j)-b(k,l+14)\|$	$\Sigma\Sigma\|a(i,j)-b(k,l+15)\|$
0	a(0,0),	b(0,0)		a(0,0)-b(0,0)				
1	a(0,1),	b(0,1)		a(0,1)-b(0,1)	a(0,0)-b(0,1)			
2	a(0,2),	b(0,2)		a(0,2)-b(0,2)	a(0,1)-(0,2)	a(0,0)-b(0,2)		
14	a(0,14),	b(0,14)		a(0,14)-(0,14)	a(0,13)-b(0,14)		a(0,0)-b(0,14)	
15	a(0,15),	b(0,15)		a(0,15)-b(0,15)	a(0,14)-b(0,15)	a(0,13)-b(0,15)	a(0,1)-b(0.15)	a(0,0)-b(0,15)
16+ 0	a(1,0),	b(1,0),	b(0,16)	a(1,0)-b(1,0)	a(0,15)-b(0,16)	a(0,14)-b(0,16)		a(0,1)-b(0,16)
16+ 1	a(1,1),	b(1,1),	b(0,17)	a(1,1)-b(1,1)	a(1,0)-b(1,1)	a(0,15)-b(0,17)		
						a(1,0)-b(1,2)		
								a(0,15)-b(0,30)
16+15	a(1,15),	b(1,15),	b(0,31)	a(1,15)-b(1,15)	a(1,14)-b(1,15)			a(1,0)-b(1,15)
2×16+ 0	a(2,0),	b(2,0),	b(1,16)	a(2,0)-b(2.0)	a(1,15)-b(1,16)	a(1,14)-b(1,16)		a(1,1)-b(1,16)
2×16+ 1	a(2,1),	b(2,1),	b(1,17)	a(2,1)-b(2,1)	a(2.0)-b(2,1)	a(1,15)-b(1,17)		
•		•				a(2,0)-b(2,2)		
•		•						
•		•						
15×16+ 0	a(15,0),	b(15,0),	b(14,16)	a(15,0)-b(15,0)	a(14,15)-b(14,16)			
								a(14,15)-b(14,30)
15×16+16	a(15,15),	b(15,15),	b(14,31)	a(15,15)-b(15,15)	a(15,14)-b(15,15)	a(15,13)-b(15,15)		a(15,0)-b(15,15)
16×16+ 0			b(15,16)		a(15,15)-b(15,16)	a(15,14)-b(15,16)		
16×16+ 1			b(15,17)			a(15,15)-b(15,17)		
							a(15,15)-b(15,29)	
16×16+14			b(15,30)					a(15,14)-b(15,29)
16×16+15			b(15,31)					a(15,15)-b(15,30)

그림 5. 움직임 추정기에서의 데이타 처리 흐름도

기는 최근에 개발되어진 LSI logic 사의 L64720 chip [7,8]에 응용되고 있다. L64720은 그림 2의 구조를 사용하되 PE의 갯수를 32개로 늘려 처리속도를 향상시킴으로써, 화상 회의용 시스템을 위한 움직임 추정을 실시간으로 실행시킬 수 있는 것이다. 이러한 VLSI 움직임 추정기는 검색 범위가 16×16 으로 제한되어 있는데, 움직임 추정기를 복수로 사용함으로써 검색범위를 원하는 범위로 확장시킬 수 있다.

한편, 움직임 추정을 위한 또 다른 방법으로 화소 단위로 반복적으로 움직임을 추정하는 pel recursive algorithm (PRA)[2,17] 이 있는데 PRA에 대한 VLSI 구조는 [18], [19] 등에서 제안된 바 있다.

2. DCT (Discrete Cosine Transform)

영상은 2차원 데이타이원 변환이어야 하지만, 2차원 DCT의 경우에는 다른 직교 변환들과 마찬가지로 1차원 DCT를 행방향과 열방향으로 거듭 수행하여 결과를 얻을 수 있다. 예를 들어 16×16 크기의 2차원 DCT는 그림 6에서 보는 바와 같이 16×1 크기의 1차원 DCT 구조들과 행렬 전치 구조를 조화시켜 수행할 수 있다. 따라서 본 절에서는 1차원, 2차원의 구별없이 DCT의 VLSI 구현 및 고속 알고리듬에 대한 설명을 한다.

DCT의 VLSI 구현 방법은 크게 두가지로 나눌 수 있는데, 첫째는 DCT 계산을 DCT 행렬과 입력 데이타의 곱으로 수행하는 파이프라인 또는 시스톨릭 구조로 구현하는 것이다[20-22]. 둘째는 1차원 DCT 또는 2차원 DCT의 고속 알고리듬을 그대로 VLSI 구현하는 것이다[23-25]. 그러나, 고속 알고리듬들은 계산량은 적어지지만 H/W 구조가 복잡하여 ASIC화 하기 힘들기 때문에 현재까지 상용으로 개발된 DCT 칩들은 대부분 첫번째 방법을 사용하였다. 즉, 그림 6에서와 같이 N×N DCT 계산을 N×1 DCT 구조를 행렬과 벡터의 곱으로

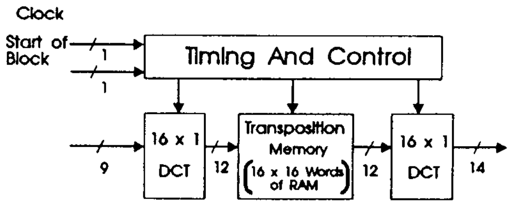
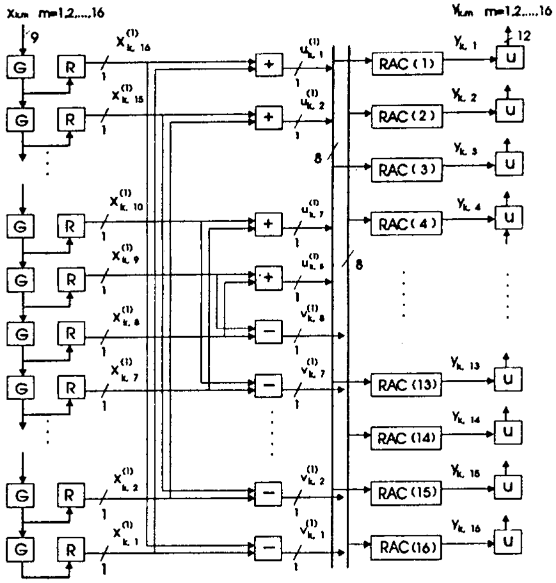


그림 6. 1차원 DCT를 이용한 2차원 DCT의 연산

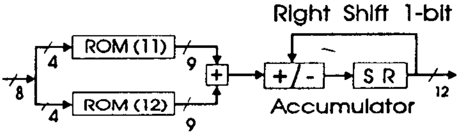
구현하는 것을 기본으로 하였다. 이 방법은 고속 알고리듬을 VLSI 구현하는 경우보다 필요한 곱셈기와 덧셈기의 수가 많다는 단점이 있지만, 구조가 매우 규칙적이므로 VLSI 구현시 설계 비용이 적고 집적도를 높일 수 있다는 장점이 있다. 특히 [20]에서의 방법을 이용한 Bellcore의 DCT 칩은 DCT를 행렬과 벡터의 곱으로 계산하면서, 여기에 필요한 곱셈을 ROM을 이용한 look-up table로 처리하는 distributed arithmetic 기법을 사용함으로써 곱셈기를 전혀 사용하지 않는다는 장점을 갖고 있다. 즉, 그림 6에서의 1차원 DCT 계산을 그림 7에서 보는 바와 같이 register, 덧셈기, ROM and accumulator(RAC)만을 이용하여 필요한 곱셈의 결과를 ROM에서 읽어 이들을 더해서 결과로 출력시키도록 한 것이다. 그 외에도 그림에서 보는 바와 같이 이 구조는 상당히 규칙적이므로 VLSI 구현에 적합함을 알 수 있다. [20]의 방법을 이용한 DCT 구조 외에도 1차원 DCT를 이용하여 현재 개발된 2차원 DCT 칩의 종류에는 Inmos사의 IMS A121, SGS-Thompson사의 STV 3208, STV 3200, LSI Logic사의 L64730 등이 있다.

고속 알고리듬을 직접 VLSI 구현하는 경우에는 필요한 곱셈기나 덧셈기의 수를 크게 줄일 수 있다. 그러나 DFT의 경우와 마찬가지로 고속 DCT 알고리듬에는 복잡한 데이타 교환이 필요하므로 VLSI 구현시 각 processing element(PE) 사이의 데이타 전달이 매우 복잡해진다. VLSI 회로에서는 각 계산 구조 사이의 데이타 전달에 많은 비용이 필요하므로 이러한 복잡한 데이타 교환이 고속 알고리듬을 VLSI 구현하는데 가장 큰 장애 요소중의 하나이다. 따라서 데이타 교환이 줄어들도록 알고리듬을 수정하거나 optical processing을 도입한 VLSI 등으로 데이타 교환에 필요한 비용을 감소시킴으로써 이러한 문제를 해결해야 한다.

위에서 소개한 방법외에도 최근에는 DCT 알고리듬의 곱셈수를 줄이는 동시에 파이프라인 구현이 가능한 비교적 규칙적인 격자형 구조로 구현하는 방법들이 연구되고 있다[26-27].



(a) 16×1 DCT 구조



(b) RAC (ROM and accumulator)의 구조

그림 7. ROM을 이용한 1차원 DCT의 구조

3. 가변 길이 부호화

영상 부호화에 사용되고 있는 엔트로피 부호화[28,29] 혹은 가변 길이 부호화에는 호프만 부호화(Huffman coding)[30], 줄길이 부호화(run-length coding)[31], 산술 부호화(arithmetic coding)[32], 그리고 Lempel-Ziv 부호화[33-37] 등이 있다. 이 중에서 호프만 부호화와 줄길이 부호화는 우수한 성능을 나타낼 뿐만 아니라 비교적 간단한 구조를 가지고 있기 때문에 MPEG, HDTV 등에 널리 사용되고 있다. 따라서 본 절에서는 호프만 부호화 및 줄길이 부호화의 구현 기법에 관하여 고찰하겠다.

줄길이 부호화의 VLSI 구조는 시프트 레지스터(shift register)를 근간으로 단순하게 구현할 수가 있다. 그리고, VLC 부호기는 입력으로, 고정된 길이의 데이타가 입력되어서, 각각에 해당되는 가변 길이 부호로 대응되므로 병렬 및 pipeline 처리 기법을 도입함으로써 기존 VLSI 기술내에서 원하는 높은 동작 속도를 얻을

수가 있다. 그러나 VLC 복호기의 경우는 입력되는 비트열(bit stream)에서 복호화 되기 이전에는 각각 심볼의 길이를 예측할 수 없으므로, 이로 인한 데이타 의존 문제(dependency problem)가 발생하게 되어, 병렬 및 pipeline 연산 기법의 도입이 불가능하다[41,47-48]. 그러나 일반적인 데이타 전송에 있어서는 한 영상을 여러 부분으로 나누어서 독립적으로, 블럭단위로 VLC를 수행하기 때문에[3,5,38,39], 독립적인 가변길이 부호화 열들을 동시에 복호화 하는 것이 가능하다. 이러한 원리로써 VLSI 구현의 복잡도를 약간 높이면서 더 빠르게 수행할 수 있도록 제안된 것이 동시처리 VLC 복호기(concurrent VLC decoder)[47,48] 이다. 그러나 동시에 처리하는 갯수를 늘리는 경우 상태 변환표의 구현 복잡도가 기하급수적으로 증가되므로 여러가지 제한된 여건에서 사용되어야 한다[47,48].

한편, [41-45]에서는 부호단어들을 ROM 또는 PLA에 저장하여 병렬 처리로 복호화 하는 것으로 매우 빠른 동작속도를 가지는 VLC를 제안하였다. 이러한 VLC 구현 기법의 대표적인 예로는 Lei와 Sun이 제안한, 그림 8과 같은 VLSI 구조를 들 수 있다. 그림 8의 구조에서는 먼저 레지스터 D1, D2에 부호화된 부호열들이 저장되며, 복호화가 진행됨에 따라 새로운 부호열을 받아들이게 된다. 다음단의 Barrel shifter는 입력 부호열 중에서 복호화될 비트들을 이미 계산된 위치에 맞추어서 PLA에 입력시켜 주게 된다. 이때 PLA는 복호화된 심볼이외에 그 심볼의 길이를 동시에 출력하게 되며, 이 값은 덧셈기 역할을 하는 Barrel shifter BS1에 입력되어, D1, D0로 부터 가지고 와야할 데이타의 위치를 BS0에 지정해주게 된다. 그림 8의 구조에서 최대 처리 속도는 데이타가 2개의 Barrel shifter와 하나의 PLA에 의해 처리되는 시간, 혹은 처리 지연 시간 (processing delay time)에 의해 결정된다. [43]에서는 이러한 처리 지연 시간을 감소시켜 처리 속도를 향상시킨 구조가 제안된 바 있다. 이러한 VLC는 Mulga라고 하는 1μm double metal CMOS 기술을 이용하여 최고 75 MHz의 동작속도를 얻을 수 있다고 알려져 있다[43]. 만약 부호단어들의 갯수가 적고, 이들을 임의로 변경해야 하는 시스템에서는 위에서 사용한 PLA 대신에 CAM(content addressable memory)를 사용할 수가 있는데, 이 방법은 동작 속도가 느려지며, 시스템이 복잡해 진다는 단점이 있다[44].

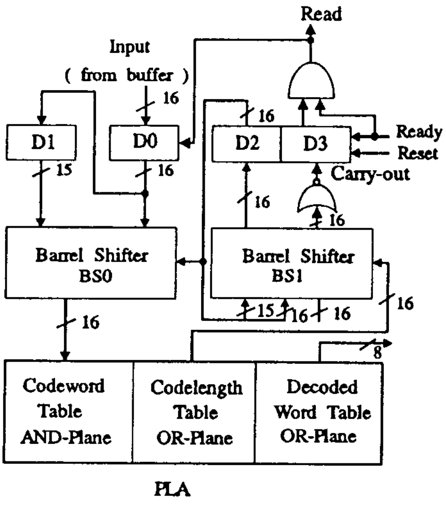


그림 8. VLC 복호기[41]

한편, 위에서 설명한 호프만 부호화 외에 여러가지 가변 길이 부호화 방법들에 대한 VLSI 구현 기법들이 소개되고 있다. 특히, 기존의 호프만 부호화와는 달리 정보당 1bit 이하로 감축시킬 수 있는 가변 길이 부호화 기법인 블럭 부호화, 산술 부호화 등에 대한 VLSI 구조에 대한 연구가 많이 진행되고 있다. 그 한 예로 여러개의 sample의 조합인 벡터 단위로 부호화하는 블럭 부호화 방법[28,29]에 대한 VLSI 구조는 [46]에 제안되어 있다. 그러나, 이러한 방법은 시스템의 복잡도가 높아질 뿐만 아니라, 산술 부호화의 경우 정보들을 즉시 복호화 할 수 없다는 단점이 있기 때문에, 현 기술로는 HDTV 시스템에 응용할 수 있도록 충분한 처리 속도를 얻을 수 있는 VLSI 시스템을 구현하기 힘들 것으로 판단된다.

4. Reed-Solomon 부호화

디지탈 통신 시스템에서는 여러 요인에 의하여 필연적으로 에러가 발생한다. 이와 같은 에러를 검출하고 정정하기 위해서 많이 사용되고 있는 에러 정정 부호가 Reed-Solomon(RS) 부호이다[49-52]. RS 부호의 부호기는 shift 레지스터를 이용하여 간단하게 하드웨어로 구현할 수 있다[49,53]. 그러나 RS 복호기의 경우는 부호기와 같이 간단하게 구현되지 않고 방대한 하드웨어를 필요로 한다. RS 부호의 복호화 과정에서 에러 위치에 대한 정보를 포함하고 있는 에러 위치 다항식(error locator poly- nomial)을 계산하는 과정이 가장 복잡하고 많은 시간을 필요로 한다. 이러한 에러 위치 다항식을 구하기 위해서 많이 사용되고 있는 알고리듬으로는

Berlekamp 알고리듬과 Euclid 알고리듬을 들 수 있다. 최근까지 위의 알고리듬들을 이용해서 RS 복호기를 VLSI로 구현하는 연구들이 많이 이루어져왔다[54-62].

먼저 Massey는 [54]에서 최소 길이의 linear feedback shift register(LFSR)을 합성하는 것이 Berlekamp 알고리듬에서 해를 구하는 과정과 동일하다는 것을 보임으로써 이 알고리듬을 하드웨어로 용이하게 구현할 수 있다는 것을 보였다. 이러한 LFSR 구조를 근간으로 하여 Liu는 [55]에서 Berlekamp 알고리듬에 기초한 RS 복호기의 VLSI 구조를 제안하였다. 이 구조는 신드롬 발생부, Berlekamp 알고리듬 수행부, 그리고 에러 교정부로 이루어진 3단계의 pipeline 형태로 구성되어 있다. 이러한 파이프 라인 구조에서 복호화 속도는 가장 느린 pipeline 단계에 의하여 결정이 된다. Berlekamp 알고리듬 수행부에서는 단위 클록내에서 세번의 곱셈이 수행되어야 하므로 복호화 속도가 제한되는 단점이 있다. 그러나 Liu는 [7]에서 자신이 제안한 RS 복호기를 VLSI로 구현하기 위해서는 같은 복호기를 SSI나 MSI를 사용해서 구현했을 경우에 비해 약 1/10 정도의 소자만이 필요하다는 것을 보였다. 한편 Brent와 Kung[56] 은 Euclid 알고리듬에 사용될 수 있는 두 다항식 간의 최대 공약수를 구할 수 있는 파이프 라인 구조를 제시하였다. Shao[57] 는 이러한 구조를 근간으로 하여 변환 영역에서 Euclid 알고리듬에 기초한 RS 복호기의 VLSI 구조를 제안하였다. 이 구조 역시 신드롬 계산부, Euclid 알고리듬 수행부, 에러값 계산부 등이 파이프 라인 형태로 구성이 된다. 나중에 [58]에서 Shao는 기존의 Euclid 알고리듬을 개선하여 RS 복호기를 VLSI로 구현할 때 필요한 면적을 줄일 수 있는 새로운 구조를 제안하였다. [58]의 VLSI 구조에 Shao가 적용한 기본 개념은 여러개의 cell을 동시에 사용하는 대신에 한 개의 cell을 순환적으로 사용한다는 것이다. 또한 여기에서 에러 값을 계산하는데 있어서 시간 영역에서의 기법이 변환 영역에서의 그것보다 더 효율적이라는 결과를 제시하였다. 그러나 이 경우 역시 실질적으로 하드웨어를 구현할 때 여전히 많은 chip 면적을 필요로 한다. 그러므로 RS 부호를 VLSI로 구현하고자 할 때 chip의 면적을 줄일 수 있는 VLSI 구조에 관한 연구는 필수적이다.

Euclid 알고리듬에서 각각의 반복(iteration)시 변경되어져야할 계수들의 수는 주어진 RS 부호가 정정할 수 있는 에러의 수 t와 같다. 반면에 Berlekamp 알고리듬에서는 매 반복시 변경되어야할 계수의 수가 차례로 증가하여 알고리듬의 마지막 반복 과정에 이르렀을 때 비로소 t와 같아진다. 그러므로 [54]에서 적용한 것과 같은 순환적인 기법(recursive technique)을 Berlekamp 알고리듬에 적용한다면 하드웨어 구현시 Euclid 알고리듬에서 필요로 하는 클럭 수보다 더 적은 클럭으로 구현할 수 있을 것이다. 이것은 결국 하드웨어 구현시 면적의 복잡도가 감소한다는 것을 의미한다. [59]에서는 이와 같은 개념을 바탕으로 기존의 Berlekamp 알고리듬을 개선하여 RS 복호기 구성시 chip 면적을 줄일 수 있도록 하였고 개선된 알고리듬을 이용하여 RS 복호기의 새로운 VLSI 구조를 제안하였다. 그림 9에 Berlekamp 알고리듬을 근간으로한 순환적 VLSI 구조를 제시하였다. 이 구조에서는 각 연산기를 순환적으로 사용하므로 기존의 구조 [55]에서 보다 필요한 cell의 수를 감소시킬 수 있게 되었다. 또한 각각의 곱셈기가 레지스터들에 의하여 분리되어 있으므로 하나의 심볼 클럭 내에 한 번의 곱셈만이 수행되도록 함으로써 복호화 속도를 향상시킬 수 있음을 보였다.

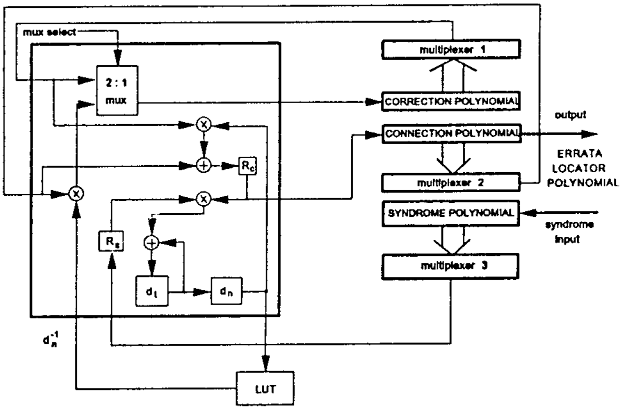


그림 9. Berlekamp 알고리듬에 근간을 둔 순환적인 VLSI 구조

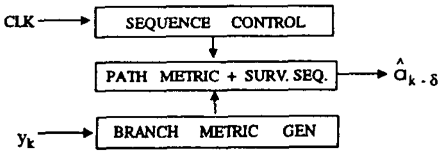
5. Viterbi 복호화

디지탈 데이타는 전송할 때 MSB나 LSB가 똑같은 정도의 잡음 영향을 받기 때문에 전송하는 채널이 좋지 못하면 아날로그에 비해 성능이 더 나빠진다. 그래서 디지탈 통신에서는 채널에 의해 발생한 데이타 오류를 바로 잡기 위해 여러가지 기법이 고안 되었는데 그 중 블럭 코딩(block-coding)과 컨벌루션 코딩(convolutional-coding)이 대표적이다. 컨벌루션 코딩 방법은 전송하고자 하는 데이타와 그 전후의 데이타를 일정한 규칙으로

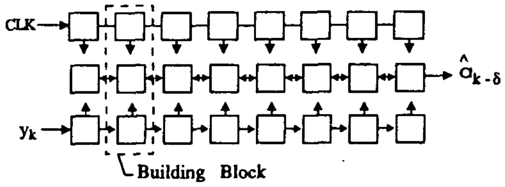
조합하여 잉여분(redundancy)을 둠으로써 채널 오류를 정정하고자 하는 방법이다. 컨벌루션 코딩의 복호에는 Viterbi 알고리듬[63]이 최적 복호 알고리듬으로 알려져 있다. Viterbi 알고리듬은 수신된 비트 신호열을 sequential하게 복호하는 MLSE(maximum likelihood sequence estimation)를 효율적으로 구현한 알고리듬인데, 이는 덧셈, 비교, 선택(add-compare-select:ACS), 트레이스-백(trace-back)의 간단한 과정으로 이루어져 있다. 그러나 각 단계마다 많은 수의 기본 state에서 행해지는 연산들 때문에 이를 구현하는 것은 간단하지 않다.

Viterbi 알고리듬을 구현하는 가장 간단한 방법으로는 하나의 마이크로프로세서를 사용하여 순차적으로 모든 스테이트의 계산을 하게 하는 것이다. 이 같은 구현은 입출력 갯수에 제한을 많이 받고 한 심벌 당 처리하는 속도가 매우 느리다는 단점이 있다. 또 다른 방법으로는 하나의 state에 하나의 프로세서를 배정하고 모든 프로세서들을 연결하여 처리하는 방법이다. 이 방법은 각 프로세서들을 연결하는데 너무 많은 면적이 필요하므로 매우 비효율적이다[68].

그래서 최근에는 Viterbi 알고리듬을 VLSI 구조로 구현하는 방법이 많이 연구되고 있는데[68-70], 이웃하는 프로세서끼리만 선형적으로 연결시켜 각 프로세서를 병렬 혹은 pipeline 식으로 수행하도록 하면서 프로세서 연결 면적은 비교적 적게 하는 systolic array 구조가 주로 연구되고 있다[68-69]. 그림 10에는 각 프로세서를 선형적으로 연결한 Viterbi 디코더의 VLSI 구조의 한 예를 제시하였다[68].



(a) 기능 블럭 선도



(b) PE 들의 배열

그림 10. Viterbi 복호기에 대한 VLSI 구조의 한 예

이 구조에서 주 연산은 중간 행의 PE에서 실행하고 상하 행은 이를 보조하는 하드웨어로 생각할 수 있다. 이 중에서 상위 행은 클럭 신호(CLK)를 입력으로 받아들여 중간 행의 각 프로세서가 필요한 일련의 제어신호를 출력해 내고 하위 행은 입력 신호 y_k를 받아들여서 중간 행의 각 프로세서가 필요한 적당한 branch metric을 생성한다. 이 branch metric은 각 심벌 구간마다 path metric 레지스터를 갱신하는데 사용된다. 중간 행에 있는 각 프로세서들은 각각 논리 회로, path metric register, survivor sequence register, add-compare-select 회로 등을 포함하는데 최종 출력으로 δ시간 만큼 지연된 복호된 데이타가 추정되어 나온다.

그림 10에서와 같은 시스톨릭 어레이 구조를 사용할 때의 장점은 각 단계에서 전체 트레이스-백 과정이 끝나기를 기다리지 않고 pipeline 식으로 일련의 레지스터에서 연속적으로 트레이스-백 동작이 수행될 수 있다는 점이다. 따라서, 그림 10의 구조와 같이 systolic array를 이용한 Vitervi 복호기 구현 방법은 앞에서 예를 든 두가지 극단적인 경우의 단점을 어느정도 보완하면서 아울러 빠른 디코더를 만들 수 있는 것이기 때문에 HDTV 시스템에 응용이 가능하리라고 기대된다.

Ⅲ. 검 토

본 고에서는 HDTV를 위한 핵심적인 5 요소들의 VLSI 구현 기법들에 대하여 고찰하였다. 제 2 장에서 설명한 각 요소들에 대한 VLSI 구조들은 systolic / wavefront array 개념에 근거한 것들로 VLSI 구현이 용이할 뿐만 아니라 병렬 및 pipeline 연산을 효율적으로 수행하므로 최대의 처리 속도를 얻을 수 있다는 장점이 있다. 또한, 본 고에서 언급한 VLSI 구조들은 비교적 간단한 연산의 반복으로 이루어져 있기 때문에 최근의 VLSI 기술 발전 추이로 보아 향후 수년내에 HDTV 처리 속도에 실시간으로 동작시킬 수 있는 VLSI chip을 구현할 수 있을 것으로 예상된다. 이러한 각 요소들에 대한 VLSI 시스템은 그림 1에 제시된 다른 부분과 조합하여 실제 HDTV 시스템을 구현하게 된다. 이때 HDTV 시스템의 다른 요소들은 본 고에서 설명한 설계 기법들을 응용한다면 쉽게 구현할 수 있을 것으로 판단된다. 그러나, 완전한 HDTV 시스템을 구

현할 때에는 이러한 각 요소들을 적절히 접속시키는 문제 또한 시스템 성능에 큰 영향을 주리라 예상된다. 따라서, 각 요소들에 대한 VLSI 설계와 함께 이에 관련된 연구도 병행되어야 할 것으로 판단된다.

參 考 文 獻

[1] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*, Plenum Press: New York, 1988.

[2] H. G. Musman, P. Pirsch and H. -J. Grallert, "Advances in picture coding." *Proc. IEEE*, vol. 73, pp. 523-548, Apr. 1985.

[3] DIGICIPHER HDTV SYSTEM, submitted by General Instrument, Co., VideoCipher Division, June 1990.

[4] Advanced Digital Television System Description, submitted to the FCC/ACATS by the Advanced Television Research Consortium, Feb. 1991.

[5] Channel Compatible DigiCipher HDTV System, submitted by Massachusetts Institute of Technology, May 14, 1992.

[6] JPEG Chip Set Technical Manual, LSI Logic, Dec. 1991.

[7] CCITT Video Compression Data Book, LSI Logic, Sept. 1991.

[8] P. A. Ruetz, et al., "A high-performance full-motion video chip," *IEEE Trans. Cir. and Syst. for Video Technology*, vol. 2, pp. 111-122, June 1992.

[9] H. Fujiwara, et al., "An all ASIC implementation of a low bit-rate video codec," *IEEE Trans. Cir. and Syst. for Video Technology*, vol. 2, pp. 123-134, June 1992.

[10] H. T. Kung, "Why systolic architecture?," *IEEE Computer Mag.*, vol. 17, pp. 37-46, June 1984.

[11] S. Y. Kung, "VLSI array processors," *IEEE ASSP Mag.*, vol. 2, pp. 4-22, July 1985.

[12] L. De Vos, M. Stegherr, T. G. Noll, "VLSI architectures for the full-search block-matching algorithm," Proc. ICASSP 1989, Glasgow, Scotland, pp. 1687-1690, 1989.

[13] L. De Vos and M. Stegherr, "Parameterizable VLSI architectures for the full-search block-matching algorithm," *IEEE Trans. Cir. and Syst.*, vol. 36, pp. 1309-1316, Oct. 1989.

[14] T. Komarek and P. Pirsch, "Array architectures foı block matching algorithm," *IEEE Trans. Cir. and Syst.*, vol. 36, pp. 1301-1308, Oct. 1989.

[15] K. -M. Yang, M. -T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," *IEEE Trans. Cir. and Syst.*, vol. 36, pp. 1317-1325, Oct. 1989.

[16] C. H. Hsieh and T. P. Lin, "VLSI architecture for block motion estimation algorithm," *IEEE Trans. Cir. and Syst. for Video Technology*, vol. 2, pp. 169-175, June 1992.

[17] S. U. Lee, "VLSI architecture and implementation III: Motion estimation," in Proc. ISCAS'92 Workshop 4: Signal Processing and VLSI Implementation for High-Definition Television, Singapore, pp. 122-137. June 1991.

[18] R. C. Kim and S. U. Lee, "A VLSI architecture for a pel recursive motion estimation algorithm," *IEEE Trans. Cir. and Syst.*, vol. 36, pp. 1291-1300, Oct. 1989.

[19] E. D. Frimout, J. N. Driessen, and E. F. Deprettere, "Parallel architectures for a pel recursive motion estimation algorithm," *IEEE Trans. Cir. and Syst. for Video Technology*, vol. 2, pp. 159-168, June 1992.

[20] M. T. Sun, T. C. Chen and A. M. Gottlieb, "VLSI implementation of a 16×16 discrete cosine transform," *IEEE Trans. Cir and Syst.*, vol. 36, pp. 610-617, April 1989.

[21] L. W. Chang and M. C. Wu, "A unified systolic array for discrete cosine and sine transforms," *IEEE Trans. ASSP*, vol. 39, pp. 192-194, Jan. 1991.

[22] U. Totzek, F. Matthiesen, S. Wohlleben and T. G. Noll, "CMOS VLSI implementation of the 2-D DCT with linear processor array," in Proc. 1990 ICASSP, Albuquerque, New

Mexico, pp. 937-940, 1990.

[23] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. ASSP*, vol. 32, pp. 1243-1245, Dec. 1984.

[24] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. ASSP*, vol. 35, pp. 1455-1461, Oct. 1987.

[25] N. I. Cho and S. U. Lee, "Fast algorithm and implementation of 2-D DCT," *IEEE Trans. Cir and Syst.*, vol. 38, pp. 297-305, Mar. 1991.

[26] C. T. Chiu and K. J. Liu, "Real-time parallel and fully pipelined two-dimensional DCT lattice structure with application to HDTV systems," *IEEE Cir. and Syst. for Video Tech.*, vol. 2, pp. 25-37, Mar. 1992.

[27] K. J. Liu, C. T. Chiu, R. K. Kolagotla and J. F. JaJa, "Optimal unified architectures for the real-time computation of time-recursive discrete sinusoidal transforms," submitted to IEEE Trans. Cir. and Syst. for Video Tech.

[28] R. G. Gallager, *Information Theory and Reliable Communication*, New York: John Wiley and Sons, Inc., 1968.

[29] R. E. Blahut, Principles and *Practice of Information Theory*, New York: Addison Wesley, 1987.

[30] D. A. Huffman, "A method for the construction of minimum redundancy-codes," *Proc. IRE*, vol. 40, pp. 1098-1101, Sept. 1952.

[31] W. K. Pratt, *Digital Image Processing*, New York: Wiley, 1991, 2nd Ed.

[32] F. Rubin, "Arithmetic stream coding using fixed precision registers," *IEEE Trans. Inform. Theory*, vol. 25, pp. 672-675, Nov. 1979.

[33] J. Ziv and A. Lempel, "Compression of individual sequences via variable- rate coding," *IEEE Trans. Inform. Theory*, vol. 24, pp. 530-536, Sept. 1978.

[34] A. Lempel and J. Ziv, "Compression of two-dimensional data," *IEEE Trans. Inform. Theory*, vol. 32, pp. 2-8, Jan. 1986.

[35] ISO/IEC, "Digital compression and coding of continuous-tone still images, Part 1, Requirements and guidlines," ISO/IEC JTC 1, DIS 10918-1, Nov. 1991.

[36] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, pp. xvii-xxxiv, Feb. 1992.

[37] W. A. Finamore and P.R.R.L. Nunes, "A modified Lempel-Ziv algorithm and its application to image compaction," in Proc. 1991 ICASSP, Toronto, Canada, pp. 2769- 2772, 1991.

[38] ISO/IEC, "Coded representation of picture and audio information," ISO/IEC JTC 1/SC 2/WG 8, MPEG 89/189, Oct. 1989.

[39] ISO/IEC, "Second draft of proposed standard on information technology- coding of moving pictures and associated audio for digital storage media up to about 1. 5 Mbit/s," ISO/IEC JTC 1/SC 2/WG 11, N0043, MPEG 90/001, Sept. 1990.

[40] P. G. Neumann, "Efficient error-limiting valiable-length codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 292-304, July 1962.

[41] S. -M. Lei and M. -T. Sun, "An entropy coding system for digital HDTV applications," *IEEE Trans. Cir. and Syst. for Video Tech.*, vol. 1, pp. 147-155, Mar. 1991.

[42] S. -M. Lei, M. -T. Sun, K. Ramachandran, and S. Palaniraj, "VLSI implementation of an entropy coder and decoder for advanced TV applications," in Proc. Int. Symp. on Circuits and Systems, New Orleans, LA, pp. 3030-3033, 1990.

[43] M. -T. Sun, "VLSI architecture and implementation of a high-speed entropy decoder," in Proc. Int. Symp. on Circuit and Systems, Singapore, pp. 200- 203, May 1991.

[44] M. -T. Sun, K. -M. Yang, and K. -H, Tzou, "A high-speed programmable VLSI for decoding variable-length codes," Applications of Digital Image Processing XII, A. G. Tescher, ed., Proc. SPIE 1153, Aug. 1989.

[45] M. -T. Sun and S. -M. Lei, "Parallel variable-length-code decoder for advanced television application," 3rd International Workshop on HDTV, Italy, Aug. 1989.
[46] S. -M. Lei, M. -T. Sun, and K. -H. Tzou, "Design and hardware architecture of high-order conditional entropy coding for images," *IEEE Trans. Cir. and Syst. for Video Tech.*, vol. 2, pp. 176-186, Jun. 1992.
[47] S. -F. Chang and D. G. Messerschmitt, "Designing high-throughput VLC decoder part I-Concurrent VLSI architectures," *IEEE Trans. Cir. and Syst. for Video Tech.*, vol. 2, pp. 187-196, Jun. 1992.
[48] H. D. Lin and D. G. Messerschmitt, "Designing high-throughput VLC decoder part II - Parallel decoding methods," *IEEE Trans. Cir. and Syst. for Video Tech.*, vol. 2, pp. 197-206, Jun. 1992.
[49] S. Lin and D. J. Costello, Jr., *Error Control Coding.*, Englewood Cliffs: Prentice-Hall, 1983.
[50] G. C. Clark, Jr. and J. B. Cain, *Error-Correction Coding for Digital Communications*, New York: Plenum Press, 1981.
[51] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd Ed., Cambridge, MA: MIT Press, 1972.
[52] R. E. Blahut, *Theory and Practice of Error Control Codes*, Reading, MA: Addison-Wesley, 1983.
[53] E. R. Berlekamp, "Bit-serial Reed-Solomon encoders," *IEEE Trans. Inf. Theory*, vol. 28, pp. 869-874, Nov. 1982.
[54] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. 15, pp. 122-127, 1969.
[55] K. Y. Liu, "Architecture for VLSI design of Reed-Solomon Decoders," *IEEE Trans. Comput.*, vol. 33, pp. 178-189, Feb. 1984.
[56] R. P. Brent and K. T. Kung, "Systolic VLSI array for polynomial GCD computation," *IEEE Trans. Comput.*, vol. 33, pp. 731-736, Aug. 1984.
[57] H. M. Shao, T. K. Truong, L. J. Deutsch, J. H. Yuen, and I. S. Reed, "A VLSI design of a pipeline Reed-Solomon decoder," *IEEE Trans. Comput.*, vol. 34, pp. 393-403, May 1985.
[58] H. M. Shao and I. S. Reed, "On the VLSI design of a pipeline Reed-Solomon decoder using systolic arrays," *IEEE Trans. Comput.*, vol. 37, pp. 1273- 1280, Oct. 1988.
[59] Y. H. Kim, Y. M. Chung, and S. U. Lee, "A VLSI Reed-Solomon decoder," to appear in SPIE VCIP'92, Boston, MA, Nov. 1992.
[60] Y. R. Shayan, T. Le-Ngoc, and V. K. Bhargaba, "A versatile time-domain Reed-Solomon decoder," *IEEE J. Select. Areas in Commun.*, vol. 8, pp. 1535-1542, Oct. 1990.
[61] B. Arambepola and S. Choomchuay, "VLSI array architecture for Reed-Solomon decoding," in Proc. Int. Symp. on Circuits and Syst., Singapore, pp. 2963- 2966, June 1991.
[62] S. R. Whitaker, J. A. Canaris, and K. B. Cameron, "Reed-Solomon VLSI codec for advanced television," *IEEE Trans. Cir. and Syst. for Video Tech.*, vol. 1, pp. 230-236, June 1991.
[63] A. J. Viterbi, "Error bounds for convolutional coding and an asymptotically optimim decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260- 269, Apr. 1967.
[64] C. M. Rader, "Memory management in a Viterbi decoder," *IEEE Trans. Commun.*, vol. 29, pp. 1399-1401, Sept. 1981.
[65] P. G. Gulak and E. Shwedyk, "VLSI structures for Viterbi receivers: Part I-General theory and applications," *IEEE J. Select. Areas in Commun.*, vol. 4, pp. 142-154, Jan. 1986.
[66] C. Y. Chang and K. Yao, "Viterbi decoding by systolic array," in Proc. Twenty-Third Annu. Allerton Conf. Commun., Cont. Comput., Allerton House, Monticello, IL, pp. 430-439, Oct. 2-4, 1985.
[67] A. J. Viterbi and A. J. Omura, *Principles of*

Digital Communication and Coding, New York: McGraw-Hill, 1979.
[68] P. G. Gulak and T. Kailath, "Locally connected VLSI architectures for the Viterbi algorithm," *IEEE J. Select. Areas in Commun.*, vol. 6, pp. 527-537, Apr. 1988.
[69] G. Fettweis and H. Meyr, "High-rate Viterbi processor: A systolic array solution," *IEEE J. Select. Areas in Commun.*, vol. 8, pp. 1520-1534, Oct. 1990.
[70] T. K. Truong, M. Shih, I. S. Reed and E. H. Satorius, "A VLSI design for a trace-back Viterbi decoder," *IEEE Trans. Commun.*, vol. 40, pp. 616-624, Mar. 1992.

筆 者 紹 介

李 商 郁
1949年 8月 11日生
1973年 서울대학교 전기공학과(학사)
1976年 미국 Iowa 주립 대학교 전기공학과(석사)
1980年 미국 USC 대학교 전기공학과(박사)

1973年 ~ 1974年 한국과학기술연구소 연구원
1980年 ~ 1981年 미국 General Electric Co. 연구원
1981年 ~ 1983年 미국 M/A COM 연구소 연구원
1983年 ~ 현재 서울대학교 제어계측공학과 부교수

주관심분야 : 디지탈 신호처리, 음성 및 영상처리, VLSI신호처리, 적응신호처리, 신경회로망, 디지탈통신 등

金 麟 澈
1963年 2月 27日生
1985年 서울대학교 제어계측공학과(학사)
1987年 서울대학교 제어계측공학과(석사)
1992年 서울대학교 제어계측공학과(박사)

1992年 4月 ~ 1992年 7月 서울대학교 자동화시스템 공동연구소 Post-Doc.
1992年 8月 ~ 현재 대우전자 영상연구소 연구 4팀 선임연구원

주관심분야 : 음성 및 영상 부호화 기법, VLSI 신호처리 등