

論文93-30B-10-1

# 재구성 가능한 다중 프로세서 시스템을 이용한 혼합 영상 부호화기 구현에 관한 연구(연구 I : H/W 구현)

## (A Study on Hybrid Image Coder Using a Reconfigurable Multiprocessor System(Study I : H/W Implementation))

崔 翔 勛\*, 李 光 基\*, 金 濟 翊\*\*, 尹 承 喆\*\*, 朴 圭 泰\*

(Sang Hoon Choi, Kwang Kee Lee, Je Ik Kim, Seung Chul Yoon and Kyu Tae Park)

### 要 約

복합 영상부호화 알고리즘(H.261, MPEG 또는 Digital HDTV)을 고속으로 수행할 수 있는 멀티프로세서 시스템을 제안하였다. 효율적이고 고성능의 부호화시스템 구현을 위하여 병렬처리가 용이한 32-bit 마이크로프로세서, DCT(Discrete Cosine Transform) 프로세서와 이동량검출 프로세서를 결합하여 새로운 프로세싱 모듈을 설계하였다. 그러나 설계한 모듈 1개만으로는 복합 부호화 알고리즘을 고속으로 수행할 수 없으므로 병렬처리 기법이 도입가능하게 구현하였고, 시스템내의 프로세싱 모듈간의 연결을 변화시켜 여러 병렬처리 구조를 실현하도록 crossbar 스위치를 도입하였다.

### Abstract

A multiprocessor system for high-speed processing of hybrid image coding algorithms such as H.261, MPEG, or Digital HDTV is presented in this study. Using a combination of highly parallel 32-bit microprocessor, DCT(Discrete Cosine Transform), and motion detection processor, a new processing module is designed for the implementation of high performance coding system. The system is implemented to allow parallel processing since a single module alone cannot perform hybrid coding algorithms at high speed, and crossbar switch is used to realize various parallel processing architectures by altering interconnections between processing modules within the system.

### 1. 서론

영상 신호를 디지털방식으로 표현하면 저장, 통신과 처리등에서 유리하나, 많은 양의 정보가 발생하기 때문에 영상 데이터의 양을 줄이는 압축 기술이 필요

하다. 이에 따라 1980년대 후반 이후 자연영상 압축 알고리즘에 대한 국제 표준화 작업을 ISO, CCITT 등의 기구에서 진행하고 있다. 이들 기구에서 추진하고 있는 대상은 크게 정지영상 압축 알고리즘과 동영상 압축 알고리즘으로 시스템의 호환성을 보장하고, VLSI 기술을 이용한 하드웨어 실현을 유도하는 작업을 진행 중이다.

정지영상 압축 알고리즘은 CCITT의 SG VIII/NIC (New Image Communication)와 ISO TC97/

\*正會員, \*\*準會員, 延世大學校 電子工學科  
(Dept. of Elec. Eng., Yonsei Univ.)  
接受日字 : 1993年 3月 23日

SC2/WG8/PEG이 공동으로 조직한 JPEG에서 표준화 작업이 진행되었으며, 주로 팩시밀리 전송, 영상 데이터 베이스와 전자 출판 등에 응용된다. [1] 동영상 알고리즘은 CCITT Working Party XV/4에서 영상 전화, 영상 회의 시스템에 응용을 목적으로 표준화 작업을 진행한 H.261과 ISO/IEC/JTC1/SC2/WG11에 의해 설립된 MPEG(Moving Pictures Experts Group)에서 CD(Compact Disk)와 DAT(Digital Audio Tape) 등 디지털 저장매체에 저장을 목적으로 하는 표준화 작업을 진행되었다. [2] 그 외에 MPEG-II와 digital HDTV등의 분야에서 동영상 부호화기법을 기반으로 표준화 및 규격을 정의하는 연구를 진행하고 있다.

위의 알고리즘중에서 JPEG 정지영상 압축 알고리즘은 C-Cube사와 LSI Logic사 등에서, 그리고 계산량이 비교적 적은 H.261 동영상 압축 알고리즘은 LSI Logic사와 IIT사 등에서 VLSI 기술을 이용하여 하드웨어로 실현하였다. 그러나 MPEG 알고리즘과 같은 복잡한 동영상 알고리즘은 부호화기만을 C-Cube사와 IIT사가 발표하였고, 그 외의 부호화 알고리즘의 하드웨어 실현은 많은 기관에서 연구 중에 있다.

일반적으로 부호화기를 하드웨어로 구현하기 위해서는 크게 두 가지 방법이 있다. 첫째는 고성능 신호처리 프로세서를 이용한 실현이고, 두번째는 고집적도의 ASIC(Applications Specific Intergrated Circuits)을 이용한 실현이다. 위의 방법에서, 신호처리 프로세서를 이용한 방법은 기존 신호처리 응용에 이용되는 고속 소자를 영상 부호화에 적용하는 방법이다. 이 방법은 알고리즘 개발과 확장이 용이하나, 완성된 시스템의 성능이 제한적이고, 시스템을 구현하는 비용이 비교적 높다. ASIC을 이용한 구현은 VHDL 등의 VLSI 설계도구를 이용하거나, 기존의 특정한 기능만을 수행하도록 개발된 소자를 이용하는 방법이다. 이 방법은 고속으로 부호화를 진행할 수 있고 대량생산에 유리한 장점이 있으나, 개발노력이 크며, 알고리즘의 추가와 변경이 어렵다는 단점이 있다. [3,4]

본 연구에서는 위의 두 가지 구현 방법의 장점을 살려서 이동량 검출과 DCT 연산같은 단순 반복적인 처리에는 ASIC으로 설계된 소자를 이용하고, 제어가 복잡하고 데이터의 구조와 수행이 크게 변화하는 처리는 융통성이 있는 신호처리 프로세서를 이용하여 알고리즘의 고속 수행과 더불어 수정이 용이하도록 프로세싱 모듈을 설계하여 동영상 부호화를 수행하였다.

## II. 동영상 복합 부호화 알고리즘

국제 표준 동영상 부호화에 쓰이는 압축기술은 영상신호의 시간적인 중복성을 제거하기 위한 이동보상 예측(motion compensated prediction) 기법과 공간적인 중복성을 제거하기 위한 이차원 DCT 변환이다. [2] 그리고, 부호화 효율을 더욱 높이기 위해 통계적 방식의 무손실 부호화인 엔트로피 부호화를 행한다.

### 1. 이동보상

효율적인 동영상 부호화를 위해 시간축에서 영상정보의 중복성을 제거하여 많은 데이터양을 감축할 수 있다. 이를 위하여 초기에는 배경예측과 조건부 화소보충 부호화(conditional replenishment coding) 기법을 이용하였으나, 이 경우 움직임이 있는 영상에 대해서는 부호화 효율이 크게 감소한다. 따라서 이전 화면에 정의한 탐색영역에서 현재 블록과의 오차가 가장 작은 블록을 탐색하여 현재 화면의 블록을 모델링하는 이동보상 예측 부호화(motion compensated predictive coding) 기법을 많이 이용한다. [5]

$$Motion\ Vector(V_x, V_y) = \underset{x,y}{Min} \sum_{i=1}^{16} \sum_{j=1}^{16} |f(i,j) - g(i+x, j+y)| \quad (1)$$

f(i, j) : current picture pixel value  
g(i+x, j+y) : displaced previous picture pixel value

이동량 검출시 예측오차는 전 탐색(full search)이 가장 작으나, 계산량의 감소를 위하여 여러 탐색기법이 연구되었다. 이동량 검출시 오차거리 함수로 곱셈 계산이 생략된 MAE(Minimum Absolute Error)를 이용하고, 과정은 식 (1)과 같다. [6] 그림 1은 이동보상 예측 부호화를 위하여 현재 블록의 이동벡터를 구하는 과정이다.

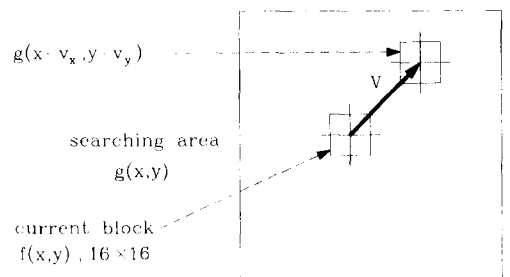


그림 1. 이전화면에서의 이동량 검출  
Fig. 1. Motion estimation from the previous picture.

동영상 부호화에서 이동량검출은 휘도 성분의 16×16크기 블럭만을 이용하여 검출한다.

2. 동영상 블럭 부호화형태 결정

동영상 부호화에서 블럭을 부호화하기 위하여 이동량검출에 의한 예측부호화기법만을 이용하지 않는다. 예를 들어 이동량검출후의 예측오차가 큰 경우에는 현화면의 블럭만을 그대로 부호화하고, 또한 이동벡터를 이용하여 구한 블럭간의 차이와 영 벡터를 이용하여 구한 블럭간의 차이가 유사한 경우 영 벡터를 이용하여 블럭간의 차이를 부호화하고, 이동벡터는 전송하지 않는다. 즉, 그림 2와 그림 3의 시험을 통하여 가장 높은 부호화효율을 갖도록 블럭의 부호화형태를 결정한다.<sup>[8]</sup>

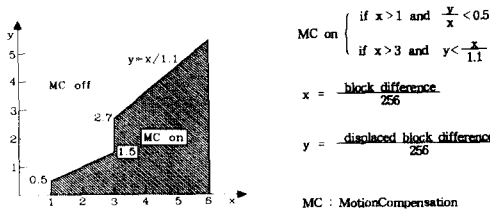


그림 2. MC / no MC 결정  
Fig. 2. MC / no MC decision.

첫째는 그림 2와 같이 부호화할 매크로블럭을 영 이동벡터에 대한 블럭간의 차이의 합과 이동벡터를 이용하여 구한 블럭차이의 합을 비교하여 화면간 부호화의 형태를 결정한다.

둘째는 부호화할 매크로블럭을 화면내 부호화를 수행할 것인가 또는 첫번째 시험을 거친 화면간 부호화를 이용할 것인가를 그림 3을 이용하여 결정한다.

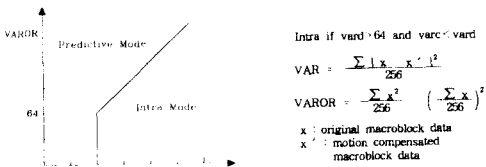


그림 3. 화면내 / 화면간 부호화 결정  
Fig. 3. Intra / non-intra decision.

3. DCT

DCT는 공간상의 신호를 주파수 영역으로 변환시키는 여러 연산중에서 상관계수(correlation coefficient) ρ가 0.9 정도인 자연 정지영상에 적용

하여 부호화에 좋은 효율을 나타내고, 변환후에 에너지를 주로 저주파에 집중시키는 성질이 있고, 인간의 시각 특성이 덜 민감한 고주파 부분에는 작은 크기의 계수를 위치하게 한다. 따라서 이러한 성질 때문에 오늘날 많은 영상부호화 알고리즘에서 DCT를 이용하고 있다.

영상 압축 국제 권고안에서는 8×8의 크기로 분할된 블럭에 대하여 DCT를 수행하는데, 한 블럭의 화소값 f(m, n)의 2차원 DCT는 식 (2)와 같이 주어진다.<sup>[11]</sup>

$$F(u, v) = \frac{4C(u)C(v)}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n) \cos\left(\frac{(2m+1)u\pi}{2N}\right) \cos\left(\frac{(2n+1)v\pi}{2N}\right) \quad (2)$$

$$C(W) = \begin{cases} \frac{1}{\sqrt{2}} & w=0 \\ 1 & w=1, 2, \dots, N-1 \end{cases} \quad \text{단, } u, v = 0, 1, \dots, N-1$$

역변환 [DCT(Inverse DCT)]는 식 (3)과 같이 된다

$$f(m, n) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos\left(\frac{(2m+1)u\pi}{2N}\right) \cos\left(\frac{(2n+1)v\pi}{2N}\right) \quad (3)$$

그러나 위의 DCT연산은 이동량검출후 현재화면과 이전화면의 차 영상에 대한 변환의 경우에는 자연 정지영상의 변환 보다는 영상압축에 불리하다.

4. 양자화 및 zigzag scanning

양자화는 DCT로 변환된 계수를 실제로 데이터 압축하는 과정으로, 주어진 출력 비트량 내에서 좋은 화질을 얻는 것이 목적이다. 따라서 인간의 시각적 특성을 고려한 양자화가 고려되며, 이를 위하여 인간 시각이 둔감한 고주파 영역에서는 DCT 계수를 거칠게 양자화하여 상대적으로 덜 중요한 정보의 양을 줄인다. DCT를 이용하는 부호화기에서의 양자화는 일반적으로 양자화 행렬값에 입력되는 영상의 특성과 버퍼상태 등 여러 요소를 고려하여 결정한다. 식 (4)와 식 (5)는 MPEG 부호화에서 화면내 부호화와 화면간 부호화의 양자화 과정을 나타낸다.<sup>[2]</sup>

화면내 양자화 :

$$Q[u][v] = \frac{8 \cdot F(u, v)}{\text{quantizer\_scale} \cdot \text{intra\_quant}[u][v]} \quad (4)$$

화면간 양자화:

$$Q[u][v] = \frac{8 \cdot F(u, v)}{\text{quantizer\_scale} \cdot \text{non\_intra\_quant}[u][v]} \quad (5)$$

위의 양자화에서 차이점은 양자화 행렬의 값이다.

표 1. MPEG 휘도 성분의 양자화 행렬 예

(a) 화면내 부호화 휘도 성분

(b) 화면간 부호화 휘도 성분

Table 1. The examples of quantizer matrix in MPEG luminance component.

(a) Intra luminance matrix

(b) Inter luminance matrix

intra\_quant [u] [v] ,

non\_intra\_quant [u] [v] .

v \ u	0	1	2	3	4	5	6	7
0	8	16	16	19	22	26	29	34
1	16	16	22	24	27	29	34	37
2	19	22	26	27	29	34	34	38
3	22	22	26	27	29	34	37	40
4	22	26	27	29	32	35	40	48
5	26	27	29	32	35	40	48	58
6	26	27	29	34	38	46	56	69
7	27	29	35	38	46	56	69	83

(a)

v \ u	0	1	2	3	4	5	6	7
0	16	16	16	16	16	16	16	16
1	16	16	16	16	16	16	16	16
2	16	16	16	16	16	16	16	16
3	16	16	16	16	16	16	16	16
4	16	16	16	16	16	16	16	16
5	16	16	16	16	16	16	16	16
6	16	16	16	16	16	16	16	16
7	16	16	16	16	16	16	16	16

(b)

양자화 행렬은 화면내 부호화와 화면간 부호화에 대하여 각각 정의되고, 휘도 성분과 색차 성분에 대하여 각각 정의된다. 표 1은 MPEG 알고리즘에 적용되는 양자화 행렬의 예이다.<sup>8)</sup>

위의 화면내 부호화의 양자화 행렬은 자연 영상의 DCT 변환후에 인간시각 특성이 고주파에 둔감한 점을 반영하였고, 화면간 부호화의 양자화 행렬은 차 영상에 대해 인간의 시각특성을 적용시키는 연구가 거의 이루어지지 않아서 모두 같은 값을 할당한다. 실제로 영상 부호화시에 위의 행렬을 이용하지 않고 새로 정의할 수 있으나, 본 논문에서는 위의 행렬을 이용하여 부호화를 수행한다.

형태로 수행한다. 이 경우 단순히 행간행(row by row) 또는 열간열(column by column) 순서로 계수를 부호화하는 것 보다 부호화되는 계수의 종류를 나타내는 EOB(End of Block) 부호를 먼저 삽입할 수 있어 부호화 효율을 증대 시킬 수 있다.

### 5. 엔트로피 부호화

자연 영상을 DCT후 양자화를 통하여 많은 압축률을 얻을 수 있다. 이러한 양자화를 통한 압축방법을 손실 부호화(lossy coding)라 한다. 국제 표준 영상 부호화기법에서는 이와 더불어 발생하는 정보의 통계적 성질을 이용하는 무손실 부호화(lossless coding)를 수행하여 좀 더 높은 데이터의 압축율을 얻는다.<sup>11)</sup> 영상 압축기법에서 이용되는 무손실 압축 부호화 기법에는 가변 길이 부호화(VLC : Variable Length Coding)기법과 영 길이 부호화(Zero-Run Length Coding)기법이 있다. VLC는 평균 비트율을 낮추기 위해 자주 발생하는 데이터에 보다 작은 길이의 코드를 부여한다. RLC는 DCT 계수를 양자화하면 많은 계수가 영의 값을 갖고 있는 성질을 이용하는 부호화 방법이다. RLC 부호화과정은 영 값이 아닌 계수 앞에 연속되는 영 값을 길이로 표현함으로써 데이터를 압축한다. 동영상 복합 부호화기법에서는 양자화된 DCT 계수뿐 아니라 이동벡터와 기타 정보도 가변 길이 부호화를 한다.

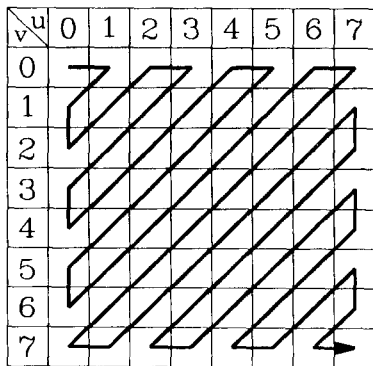


그림 4. 양자화된 계수의 zigzag scanning

Fig. 4. Zigzag scanning of the quantized coefficients.

양자화후에 나타나는 계수의 부호화 효율을 높이기 위하여 그림 4와 같이 계수의 부호화 순서를 zigzag

동영상 부호화기 하드웨어 구현은 알고리즘의 구현과 수정의 용이함과, 고속처리를 목적으로 한다. 본

### Ⅲ. 동영상 부호화기의 구성

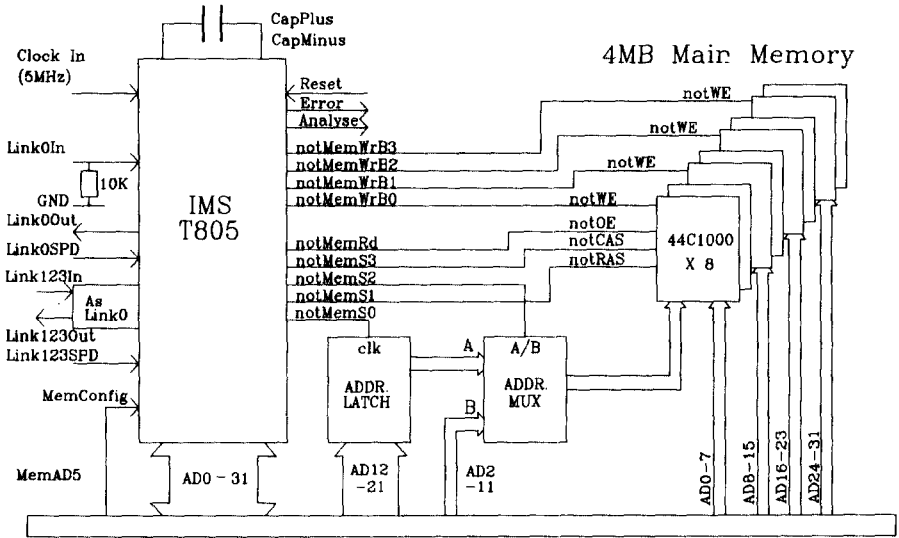


그림 5. 주 프로세서부 구성도  
Fig. 5. Block diagram of the main processor part.

연구에서는 복합 부호화 알고리즘을 효율적으로 수행하기 위하여 고속 프로세서와 기존에 개발된 ASIC을 이용하여 프로세싱 모듈을 설계하였다.

1. 주 프로세서

일반적으로 동영상 부호화기를 구현시에 알고리즘 복잡성과 향후 알고리즘 발전과 수정을 위하여 고성능 프로세서를 기반으로 한 설계가 적합하다. 다음은 주 프로세서부 설계시에 고려하여야 할 주요사항이다.<sup>13)</sup>

- 1) 주 프로세서의 계산능력
- 2) 프로세서간의 고속통신 기능
- 3) 하드웨어 개발의 용이성

위의 조건을 모두 만족하는 프로세서를 선정하는 것이 중요하다. 기존에 영상부호화에 이용되는 고성능 소자는 크게 VSP(Video Signal Processor), DSP(Digital Signal Processor) 소자와 Transputer 등이 있다.

우선 VSP는 동영상 부호화에 가장 적당하게 설계되어 가장 효과적으로 시스템을 구성할 수 있으나, 기존의 상용시장에는 출현하지 않고 소수의 설계자만이 이용하고 있다.<sup>16)</sup>

DSP 소자는 알고리즘의 복잡도에 따라 1개 또는 2개 정도를 주로 음성 CODEC(CODer/DECoder)에

이용되었다. 그러나 보다 고속으로 복잡한 알고리즘 처리를 요구하는 영상부호화에서는 여러 개의 DSP 소자를 병렬처리기법으로 시스템을 구성하여야 한다. 그러나 기존의 DSP 소자는 병렬처리에 관한 고려가 없으므로, DSP 소자를 기반으로 한 시스템은 복잡하고, 효율성도 크게 부족하다.

Transputer는 병렬처리를 위해 설계된 32-bit RISC(Reduced Instruction Set Computer)구조를 갖는 범용 마이크로프로세서이다.<sup>17)</sup> Transputer는 기존의 몇 가지 설계에서 DSP 소자를 이용한 경우 보다 영상부호화기 설계에 보다 적합한 점을 나타내었다. 그러나 Transputer는 DSP 소자에 비교하여 계산능력이 부족하기 때문에 보다 많은 수를 시스템내에 포함시켜야 한다.

본 연구에서는 계산능력의 부족은 DCT 프로세서와 이동량 검출 프로세서를 도입하여 해결하고, 병렬처리에 적합한 Transputer를 주 프로세서로 이용한다.

그림 5는 주 프로세서부의 구성도이다.

다음은 주 프로세서로 이용하는 Transputer T805의 제원이다.<sup>19)</sup>

- i) instruction throughput : up to 30MIPS (Million Instructions Per Second), 1.5 MFLOPS(Mega FLoating point Operations Per Second)

- ii) data bus width : 32bits, address space : 4Gbytes
- iii) on-chip memory : 4Kbytes
- iv) communication channel : 4, 20Mbps links
- v) internal hardware scheduler
- vi) 2-D block move instruction, CRC generation instruction support

위의 제원중에서 병렬처리를 위한 기능은 4개의 20Mbps link와 internal hardware scheduler이다. 이 기능은 다른 프로세서와의 통신을 고속으로 용이하게 하고, 프로세서 내에서 다중 처리를 가능하게 한다. 또한 2-D block move 명령은 2차원 데이터를 고속으로 이동하는 명령으로 동영상 부호화에서 이동보상에 유용하게 이용할 수 있다.

주 프로세서에서는 (I) DCT와 이동량검출 연산을 제외하고, 양자화 및 역양자화와 이동량검출후에 매크로블럭의 부호화 형태를 결정하는 알고리즘 처리를 담당한다.

다음은 식 (4)와 식 (5)에서 주어진 양자화에 대한 계산량이다.

$$T_Q \approx (3T_{load} + 1T_{save} + 2T_{mul} + 1T_{div}) \cdot BS^2 \quad (6)$$

$$T_{load} = \text{data load time from memory} (= 8 \text{ cycles})$$

$$T_{save} = \text{data save time to memory} (= 8 \text{ cycles})$$

$$T_{mul} = \text{multiply operation time} (= 38 \text{ cycles})$$

$$T_{div} = \text{divide operation time} (= 39 \text{ cycles})$$

$$BS^2 = \text{block size}$$

위의 식에서  $T_{load}$ 와  $T_{save}$ 는 임의의 2 차원 데이터 구조와 프로세서간에 데이터를 이동하는데 소요되는 시간이다. 따라서 위의 각 부분 연산의 사이클 개수를 대입하여 계산량  $T_Q$ 를 구하면 8,380 사이클 정도가 요구된다. 그림 2와 그림 3을 이용하여 블럭의 부호화 형태를 구하기 위하여 소요되는 연산량을 고려하면 다음과 같다. 먼저 MC/no MC 시험의 경우 block difference의 값은 이동량검출 프로세서에서 이동량 검출시에 구한 값을 이용하므로 연산량은 다음과 같으며,  $T_{MC, No. MC}$ 의 계산 결과값은 약 120 사이클이다.<sup>[10]</sup>

$$T_{MC, No. MC} = 2 \times (2 \cdot T_{comp} + 2 \cdot T_{load} + T_{div}) \quad (7)$$

$$T_{comp} = 2 \text{ variables compare time} (= 2 \text{ cycles})$$

Predictive/Intra 결정은 VAR과 VAROR 계산과

한 차례의 비교 연산이 소요된다. 그러나 위의 VAR과 VAROR 계산에서 제곱연산은 미리 계산하여 값을 저장한 lookup table을 이용하여 곱셈연산을 수행하지 않고 단순 데이터 인출로 대체할 수 있다.

$$T_{Predictive/Intra} = T_{VAR} + T_{VAROR} + 2 \cdot T_{comp} \quad (8)$$

$$T_{VAR} = T_{abs} + BS^2 \cdot T_{index} + (BS^2 - 1) \cdot T_{add} + T_{div}$$

$$T_{VAROR} = T_{var} + T_{div} + T_{mean} + T_{div} + T_{mul} + T_{sub}$$

$$T_{abs} = 2 \cdot BS^2 \cdot T_{load} + BS^2 \cdot T_{sub}$$

$$T_{var} = BS^2 \cdot T_{load} + BS^2 \cdot T_{index} + (BS^2 - 1) \cdot T_{add}$$

$$T_{mean} = BS^2 \cdot T_{load} + (BS^2 - 1) \cdot T_{add}$$

$$T_{index} = \text{Lookup table indexing time} (= 2 \cdot T_{load} + T_{save})$$

$$T_{add} = \text{addition time} (= 1 \text{ cycle})$$

$$T_{sub} = \text{subtraction time} (= 1 \text{ cycle})$$

식 (8)의  $T_{Predictive/Intra}$ 를 계산하면 약 5,530 사이클이다.

## 2. DCT 프로세서부

기존의 국제표준 자연영상 압축 알고리즘에는 DCT가 포함되어 있다. 여기에서 이용되는 DCT는 8×8 크기의 2차원 연산으로 1024 MAC(Multiplications and Accumulations) 정도의 연산량이 요구된다. H.261의 CIF(Common Intermediate Format) 영상을 처리하는 경우에는 약 73 MMAC (Million MAC)가 소요되고, 고속 알고리즘을 이용하여도 14 MMAC의 연산이 필요로 한다. 동영상 부호화 알고리즘에서는 이동량 탐색을 위한 이전화면을 구성하여야 하므로 같은 계산량의 IDCT 연산이 필요로 하고, MPEG 부호화기법에서는 더욱 많은 연산이 요구된다.<sup>[6]</sup>

본 연구에서는 단순 반복적으로 많은 계산이 요구되는 DCT 연산을 위해 INMOS사의 IMSA121을 도입하였고, 제원은 다음과 같다.<sup>[10]</sup>

- 1) 8×8 transform size
- 2) 8×8 calculation time = up to 3.2 μsec
- 3) multi-function(DCT, IDCT, filtering, matrix transpose)
- 4) Dx port, clipping capability

위의 특징중에서 동영상 부호화에 유리한 것은 Dx port의 이용이다. 즉, 부호화시에는 Dx port는 이동 검출된 이전화면을 입력 받아 현화면과의 차영상에 대하여 DCT를 수행할 수 있고, 복호화시에는 DCT

계수를 IDCT하여 이전화면의 8×8 블록데이터와 합하여 0과 255사이의 값으로 클리핑하여 현화면을 구성할 수 있다. 그림 6은 IMSA121을 이용한 DCT부의 구성도이다.

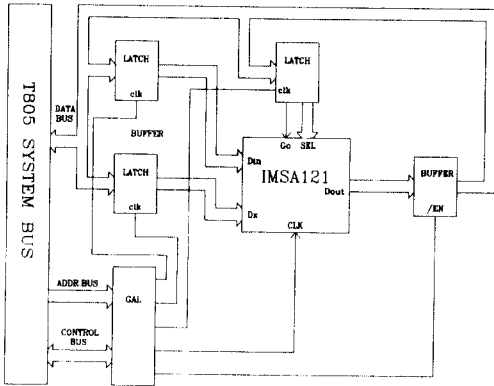


그림 6. DCT부의 구성도  
Fig. 6. Block diagram of the DCT part.

IMSA121을 이용하여 (I)DCT를 수행하는 경우, 덧셈이나 곱셈 같은 산술 연산은 배제되고 프로세서와 데이터를 이동하는 과정만이 필요하다. 연산은 8×8 크기의 블록을 입력한 후, 128 사이클의 파이프라인 지연후에 결과가 출력된다. 따라서 처음 블록에 대하여만 128 사이클 동안 결과가 출력되지 않고, 그 후에는 데이터의 입력과 동시에 두 블록 이전의 결과가 출력된다. 따라서 한 블록을 (I)DCT 수행할 때 소요되는 시간은 다음과 같다.

$$T_{DCT\_intra} = T_{Din} + T_{Dout} \quad (9)$$

$$T_{Din} = 64 \cdot T_{out}$$

$$T_{Dout} = 64 \cdot T_m$$

$$T_{out} = \text{data output time} (= 18 \text{ cycles})$$

$$T_m = \text{data input time} (= 18 \text{ cycles})$$

식 (9)의 연산은 화면내 부호화시에 소요되는 시간이고, 화면간 부호화시에는 Dx port를 통하여 입력받은 이전화면의 값과 Din port의 값과의 차이를 구하여 DCT를 수행할 수 있고, IDCT의 경우에는 이전 화면을 Dx port로 입력받아 차영상의 IDCT 결과와 합하여, 0~255 사이의 값으로 클리핑하여 Dout port에 출력시킨다. 위의 연산에 대하여 소요되는 시간은 다음과 같다.

$$T_{DCT\_inter} = T_{Din} + T_{Dout} + T_{Dx} \quad (10)$$

$$T_{Dx} = 64 \cdot T_{out}$$

위의 식에서  $T_{in}$ 과  $T_{out}$ 은 임의의 2 차원 데이터 구조와 DCT 프로세서간에 데이터를 이동하는데 소요되는 시간이다. 위의 연산시간을 산출하면,  $T_{DCT\_intra}$ 는 약 2,300 사이클이고,  $T_{DCT\_inter}$ 는 3,500 사이클이 소요된다.

### 3. 이동량 검출 프로세서

동영상 부호화에서 가장 많은 계산량이 필요한 부분은 이동량 검출이다. 이동량 검출은 뺄셈과 절대값 연산, 누적 덧셈의 단순한 연산으로 구성되어 있다. 그러나 이러한 반복적인 연산이 이전화면의 탐색영역으로 정의된 부분에서 식 (1)의 연산을 반복한다. 예를 들어 -8 ~ +7 탐색영역내에서 CIF 영상에 대한 이동량 검출을 위한 연산량은 780 MOPS(Million Operations Per Second)가 필요하다. 따라서 이전에 소개된 DCT 연산 보다 많은 연산량이 필요하다. 이러한 연산은 프로세서내에서 고속 수행이 불가능하므로 전용 ASIC의 도입이 필요하다.

본 연구에서는 이동량 검출을 위하여 내부에 256개의 시스톨릭 어레이 모듈이 포함된 SGS-THOMSON사의 STI3220을 도입하였고, 제원은 다음과 같다. [9]

- 1) pixel rate : up to 18MHz
- 2) block size : 8×4n, 16×4n
- 3) search range : -8 ~ +7
- 4) error function : full search with MAE

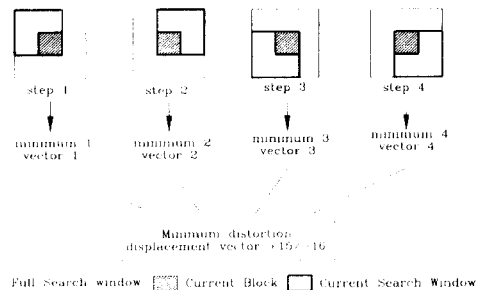


그림 7. STI 3220에서 -16 ~ +15 탐색의 예  
Fig. 7. An example of searching -16 ~ +15 with STI 3220.

STI3220의 제원에서 탐색범위는 -8 ~ +7이다. 그러나 보다 정확하게 이동량을 검출하기 위해서 -16 ~ +15의 범위내에서 탐색을 수행하여야 하고, MPEG 알고리즘으로 부호화시 이동량 검출을 세 화면 이전을 기준으로 수행하는 경우 탐색범위가 -48

~ +47이므로 STI3220의 탐색범위를 벗어난다. 따라서 본 연구에서는 그림 7과 같이 -8 ~ +7 범위의 탐색을 4번 수행하여 -16 ~ +15 범위의 영역을 수행하고, 화면간격이 1이상인 경우에는 telescopic 탐색기법을 이용한다.<sup>[8]</sup>

STI3220을 이용하여 16×16 크기의 블록을 -16 ~ +15 범위에서 이동벡터를 구하고자 할 때, 그림 7과 같이 -8 ~ +7 범위를 4회 탐색하여 전체 이동벡터를 구할 수 있다.

그림 8은 STI3220을 이용한 이동량 검출부의 구성도이다.

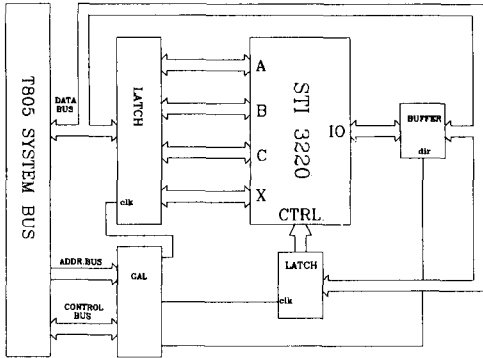


그림 8. 이동량 검출부의 구성도  
Fig. 8. Block diagram of the motion estimation part.

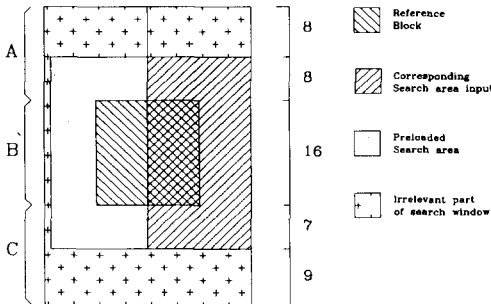


그림 9. STI3220의 데이터 입력  
Fig. 9. The data input to the STI3220.

STI3220을 이용하여 16×16 크기의 블록을 -8 ~ +7 범위에서 이동량을 검출하는 경우 초기화 과정(initial sequence)과 블록 과정(block sequence)로 나누어서 그림 9와 같이 데이터를 입력시킨다. 초기화 과정은 이전화면만을 입력하기 위한 과정으로 16 사이클의 소자 초기화에 이어 31 row × 15

column 크기의 이전 배경화면을 A, B와 C port를 통하여 입력한다. 블록 과정은 오른쪽의 31 row × 16 column 크기의 배경화면과 16×16 크기의 현화면을 A, B, C와 X port를 통하여 입력한다. 이 입력 후 이동벡터는 38 사이클 후에 출력되고, 지연은 첫 블록에만 존재하는 파이프라인을 채우는데 소요되는 시간이다. 연속되는 블록에 대해서는 DCT 프로세서와 같이 두번째 블록이후 부터 데이터 입력과 동시에 출력이 가능하다.

위의 모든 상황을 고려하여 16×16크기의 블록을 -8 ~ +7의 범위에서 이동량 검출에 소요되는 시간  $T_{ME8}$ 은 다음과 같다.

$$T_{ME8} = T_{initial8} + T_{block8} + T_{MV\_in8} \quad (11)$$

$$T_{initial8} = (16 + 31 \times 15) \cdot T_{out}$$

$$T_{block8} = (31 \times 16 + 16 \times 16) \cdot T_{out}$$

$$T_{MV\_in8} = T_{in}$$

그러나 16×16 크기의 블록을 -16 ~ +15 범위에서 그림 7과 같이 이동벡터를 구하는 경우, subwindow 2와 subwindow 4의 초기화 과정은 subwindow 1과 subwindow 3의 블록 과정으로 생략 가능하고, 4개의 벡터중에서 가장 이동량이 적은 벡터를 선택하는데 소요되는 시간을 고려하여 전체 계산시간을 다음과 같이 구할 수 있다.

$$T_{ME16} = T_{initial16} + T_{block16} + T_{MV\_in16} + T_{comp4} \quad (12)$$

$$T_{initial16} = 2 \times T_{initial8} \text{ cycles}$$

$$T_{block16} = 4 \times T_{block8} \text{ cycles}$$

$$T_{MV\_in8} = 4 \times T_{MV\_in8} \text{ cycles}$$

$$T_{comp4} = 3 \times T_{comp} \text{ cycles}$$

위의 식에 값을 대입하여  $T_{ME8}$ 과  $T_{ME16}$ 을 구하면 22,000 사이클과 72,000 사이클 정도이다.

#### 4. Reconfigurable architecture 구현

Transputer는 다른 프로세서에 비교하여 병렬처리를 용이하게 구현할 수 있다. 그 이유는 Transputer 내부에는 고속 H/W scheduler와 20Mbps link 4개가 내장되어 있다. H/W scheduler는 프로세서 내부에서 여러 작업의 병렬 수행을 효율적으로 지원하고, link는 멀티프로세서시스템을 구현시에 각 프로세서간에 정보전달을 용이하게 한다.<sup>[9]</sup>

본 연구에서는 master 프로세서에서 소프트웨어에 의하여 프로세서의 link의 연결구조를 변경할 수 있도록 INMOS사의 32×32 crossbar switch C004를



시스템내에 포함시켰다. 이 crossbar switch는 임의의 프로세서간의 데이터 교환을 위한 통신경로를 제공하여, 여러 컴퓨터 구조를 별도의 하드웨어 변경없이 소프트웨어에 의하여 쉽게 변경할 수 있고, 알고리즘의 수행도중 인접하지 않은 프로세서간의 데이터 교환시 이용할 수 있다.

그림 10은 시스템내의 프로세서들과 C004와의 연결을 표시하는 구성도이다.

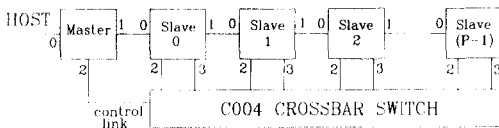
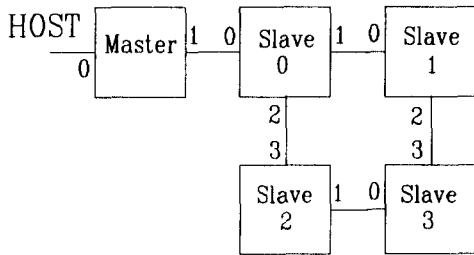
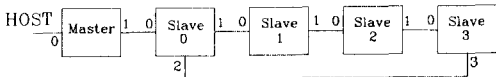


그림 10. 전체 시스템 구성도  
Fig. 10. Block diagram of the overall system.

그림 10에서 프로세서에 할당된 번호는 link의 번호이다. 그림 11은 동영상 부호화기에 효율적으로 이용할 수 있는 구조의 예이다.



(a)



(b)

그림 11. 프로세서 연결의 예  
(a) 2차원 mesh 구조  
(b) 1차원 ring 구조  
Fig. 11. Examples of processor connection.  
(a) Two dimensional mesh structure,  
(b) One dimensional ring structure.

IV. 실험 및 결과고찰

본 연구에서 구현한 시스템의 성능을 평가하기 위하여 352×240 SIF영상에 대하여 이동량검출과

DCT알고리즘을 그림 9에 표시한 2차원 mesh 구조 상에서 프로세서의 갯수를 변화시키면서 수행시간과 가속성(speed up)을 구하였다.

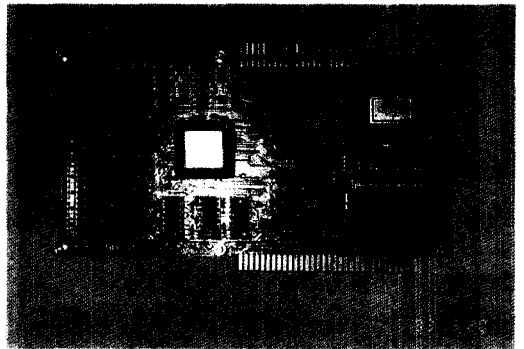


그림 12. 구현된 주 프로세서부  
Fig. 12. The implemented main processor part.



그림 13. 구현된 이동량 검출부와 DCT부  
Fig. 13. The implemented motion estimation part and DCT part.

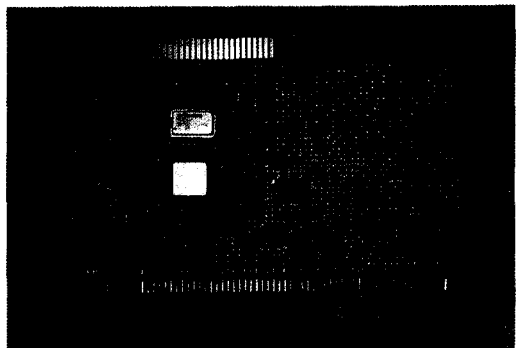


그림 14. 구현된 C004 crossbar 스위치부  
Fig. 14. The implemented C004 crossbar switch part.

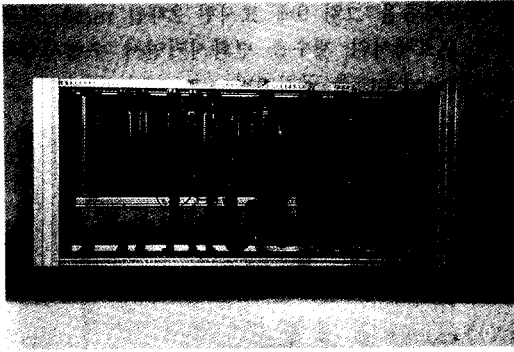


그림 15. 구현된 동영상 부호화 시스템

Fig. 15. The implemented motion image coder system.

표 2. DCT 연산과 이동량 검출 연산 시간 비교

Table 2. The comparison of the DCT & ME operations.

DCT 연산				
	SUN	1PE	2 PEs	4 PEs
Processing Time	1.61	0.353	0.202	0.101

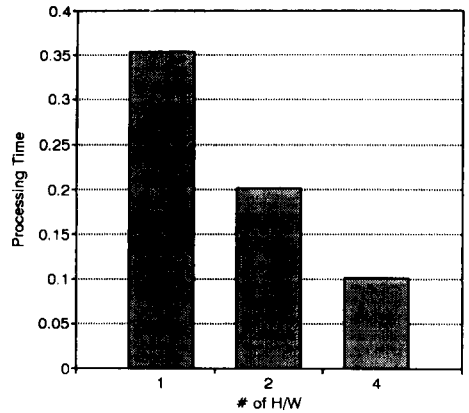
  

이동량 검출 연산				
	SUN	1PE	2 PEs	4 PEs
Processing Time	55.98	3.412	1.908	0.904

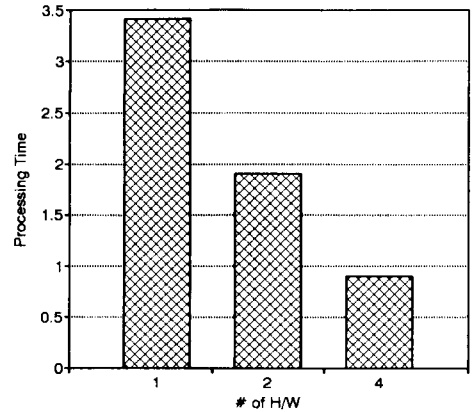
PE : Processing Element

그림 12는 Transputer와 4M RAM을 결합한 주 프로세서 보드이고, 그림 13은 DCT 프로세서와 이동량 검출 프로세서를 결합한 보드이고, 그림 14는 C004를 도입한 crossbar switch 보드이고, 그림 15는 위의 설명한 보드를 결합하여 구성된 고속 동영상 시스템이다.

본 시스템에서의 수행시간의 평가는 Transputer 내에 존재하는 타이머를 이용하였다. 이 타이머는 프로세서의 우선순위(priority)에 따라 1 $\mu$ sec(high priority) 또는 64 $\mu$ sec(low priority)로 동작한다. 본 시스템의 유용성을 나타내기 위하여 SUN-SPARC STATION II에서 알고리즘을 수행하여 비교하였다. SUN에서의 수행시간은 getusage()를 이용하여 측정하였고, system service time을 제외한 user time만을 고려하였다.



(a)



(b)

그림 16. 구현된 시스템의 가속성

(a) DCT 연산 (b) ME 연산

Fig. 16. The speed-up of implemented system.

(a) DCT operation, (b) ME operation.

SUN에서 수행하는 DCT는 미리 COSINE 계수를 구한 후에 알고리즘을 수행 하였고, 이동량검출은 전 탐색을 수행 하였다. 표 2는 측정된 수행시간을 나타내었고, 그림 16은 본 시스템에서 가속성을 표시한다.

이동량 검출 수행시간이 DCT 수행시간에 비교하여 SUN에 대한 가속성이 우수한 것은 이동량 검출 프로세서내에 256개의 시스톨릭 어레이를 포함하여 고속으로 알고리즘을 수행할 수 있기 때문이다.

### V. 결론

복합 동영상부호화 알고리즘을 효율적으로 수행할

수 있고, 프로그램에 의해 임의로 프로세싱 모듈간의 연결을 쉽게 바꿀 수 있는 멀티프로세서 시스템을 제안하고, 구현하였다. 알고리즘의 추가 변경의 용이성과 고속수행을 위하여 INMOS사의 32-bit 마이크로 프로세서, DCT프로세서와 이동량검출 프로세서를 결합하여 새로운 프로세싱 모듈을 설계하였다.

구성된 시스템의 성능을 평가하기 위해 DCT와 이동검출 알고리즘을 직렬구조에서 프로세서 갯수를 변화하며 비교하였다. 실험결과 352×240 SIF영상을 단일 프로세싱 모듈로 차영상 DCT연산의 경우 0.35초를 나타냈고, 이동량검출에서는 3.5초를 나타냈고, 프로세싱 모듈을 증가시키기에 따라 거의 선형적인 성능을 보였다. 위의 모든 수행시간은 SUN SPARC STATION II에 비교하여 월등이 성능이 우수함을 입증하였다.

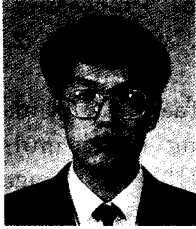
그러나 위의 성능은 실제로 동영상 부호화를 실시간으로 진행시키기에는 제약이 있으므로 좀 더 고속으로 DCT와 이동량 검출을 수행할 수 있는 프로세싱 모듈 설계에 관한 연구가 필요하다.

마지막으로 위의 프로세싱 모듈에서 양자화 및 Huffman 부호화와 같이 medium level 처리를 효과적으로 처리하기 위해 고속 전용곱셈기와 길이와 관계없이 단일 클럭에 shift연산을 수행하는 barrel shifter가 포함된 고성능 프로세서 도입이 요구된다. 이 경우 양자화등의 특정연산에서 수행시간을 1/10이상 줄일 수 있어 알고리즘 수행효율을 크게 증대시킬 수 있다.

#### 参 考 文 献

- [ 1 ] Gregory K. Wallace, "The JPEG Still Picture Compression Standard," Communications of the ACM, vol. 34 no. 4, pp.31-44, April 1991.
- [ 2 ] Didier Le Gall, "MPEG : A Video Compression Standard for Multimedia Applications," Communications of the ACM, vol. 34 no. 4, pp.47-58, April 1991.
- [ 3 ] Cornelis HOEK, et al, "An Array Processor Approach for Low Bit Rate Video coding," Signal Processing: Image Communication, vol. 1 no. 2, pp.213-223, Oct. 1989.
- [ 4 ] Sung Nam Choi, Kyu Tae Park, et al, "Efficient video coding using multiprocessor system," ICCT Proc. Image Processing I pp. 02.01.1-02.01.4 1992.
- [ 5 ] INMOS, "Introduction to Motion Picture Coding and the CCITT Coding Algorithm," INMOS, Feb. 1990.
- [ 6 ] Ming L. Liou, "Visual Telephony as an ISDN Application," *IEEE Communications Magazine*, pp.30-38, Feb. 1990.
- [ 7 ] R. A. Loftufo, et al, "Hough Transform for Transputer Array," IEE Third international conference on conference publication number 307 Image Processing and Its Applications, pp.122-130, July 1989.
- [ 8 ] ISO/IEC JTC1/SC2/WG11, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbits/s," ISO/IEC JTC1/SC2/WG11, Nov. 1991.
- [ 9 ] INMOS, *The Transputer Databook*, Second Edition, INMOS, 1989.
- [ 10 ] SGS Thomson Microelectronics, *Image Processing Data Book*, 1st Edition, SGS Thomson Microelectronics, Oct. 1990.

著 者 紹 介



崔 翔 勛(正會員)

1965年 8月 27日生. 1987年 2月  
아주대학교 전자공학과 졸업(공학  
사). 1989年 2月 연세대학교 전자  
공학과 졸업(공학석사). 1993年 8  
月 연세대학교 전자공학과 졸업  
(공학박사). 주관심분야는 영상압

축, 컴퓨터 아키텍처 등임



李 光 基(正會員)

1963年 7月 9日生. 1986年 2月  
연세대학교 전자공학과 졸업(공학  
사). 1988年 8月 연세대학교 전자공  
학과 졸업(공학석사). 1993年 8月  
연세대학교 전자공학과 졸업(공학  
박사). 주관심분야는 영상압축 등

임.



尹 承 喆(準會員)

1965年 7月 28日生. 1990年 2月  
연세대학교콜 전자공학과 졸업(공학  
사). 1992年 8月 연세대학교 전자  
공학과 졸업(공학석사) 주관심분  
야는 영상압축 등임.



金 濟 翊(準會員)

1970年 1月 14日生. 1992年 2月 연  
세대학교 전자공학과 졸업(공학사).  
1992年 3月 ~ 현재 연세대학교 전  
자공학과 공학석사 재학중. 주관심분  
야는 영상압축 등임.

朴 圭 泰(正會員) 第 28卷 B編 第 11號 參照

현재 연세대학교 전자공학과 교수