

# 10Mb/s의 傳送率을 갖는 AMBTC를 利用한 映像符號器/ 復號器의 實時間 具現

## (A Real Time Implementation of Picture Coder/Decoder Using AMBTC at the Data Rate of 10Mb/s)

高 亨 和\*, 李 忠 雄\*\*

(Hyung Hwa Ko and Choong Woong Lee)

### 要 約

本 研究에서는 absolute moment block truncation coding (AMBTC) 理論을 利用하여 動畫像데이터 壓縮을 위한 實時間動作시스템을 具現하고, NTSC TV 신호를 사용하여 실험하였다. 全體시스템은 符號器와 復號器로 나누어지고, 符號器는 4 走査線버퍼 메모리부와 데이터 處理部로 構成된다.

復號器는 2 필드버퍼 메모리부와 어드레스 발생부로 나누어진다. 그 밖에 A/D 변환, D/A 변환, 동기분리 및 제어신호 발생부가 첨가된다. 실험結果로 80Mb/s의 전송율을 갖는 原畫像과 10Mb/s의 전송율을 갖는 處理된 畫像을 비교하여 나타냈다. 약간의 劣化가 있으나 그다지 거칠어 지지는 않았다.

### Abstract

This paper describes an implementation of the absolute moment block truncation coding (AMBTC) in real time for the moving picture data compression. We have realized a system composed of the encoder and decoder, and operated it using an NTSC TV signal. The encoder consists of a 4-line buffer memory and a data processing block. Besides, there are signal conditioner and a control signal generator. Experimental results show that the quality of the processed image with a data rate of 10 Mb/s is slightly degraded, but not objectionable, comparing data rate of 80 Mb/s.

### I. 序 論

오늘날 通信分野에서는 離散信號處理技術의 도입과 함께 急激한 變革을 이루고 있다. 社會的으로도 더욱

다양한 通信서비스를 要求하고 있다. 映像通信 分野에 있어서는 ISDN(integrated services digital network)의 도입과 함께 videotex, teleconference, picture phone, 高品位 TV 등 여러가지 새로운 시스템이 등장하고 있다. 이러한 映像通信의 새로운 서비스를 充足시키기 위해서는 디지털화된 映像데이터를 貯藏하고 또 傳送해야 한다.

그런데, 영상신호를 디지털화 할 경우 그의 대역폭이 대폭 늘어나는 結果를 가져오므로 經濟的인 面에서 볼 때 이러한 늘어나는 帶域幅을 줄여주면서 만족할

\*正會員, 光云大學校 電子通信工學科  
(Dept. of Elec. Comm., Eng., Kwang Woon Univ.)

\*\*正會員, 서울大學校 電子工學科  
(Dept. of Elec. Eng., Seoul Nat'l Univ.)

接受日字: 1987年 4月 16日

만한 畫質을 維持하도록 하는 方法의 開發이 必要하다. 1970년대부터 활발하게 研究된 映像帶域壓縮方法은 現在까지도 계속적으로 研究되어질 만큼 큰 關心분야이다.<sup>1),2)</sup> 지금까지의 研究추세는 컴퓨터를 利用한 모의 실험을 통해 非實時間에서 行해졌는데, 그 理由는 실시간處理를 위해서는 超高速의 記憶소자가 막대하게 필요하고, 데이터處理가 빨라야 하기 때문이다. 따라서, 영상압축장치의 實時間具現은 DPCM, Hadamard 변환, conditional replenishment 方法 등을 제외하고는 드문 실정이다. 그러나, 고속의 VLSI 칩이 계속 개발됨에 따라 더 많은 알고리즘의 실시간 구현이 가능해지고 있다.

본 논문에서는 absolute moment block truncation coding (AMBTC) 理論을 利用하여 實時間動作 映像壓縮 시스템을 具現하고 그 結果를 보였다. AMBTC는 다른 영상부호화 방식에 비해 계산량이 훨씬 적고, 블럭별로 처리하기 때문에 메모리량을 줄일 수 있다. 개발된 시스템은 並列處理를 통해 계산시간을 최소화하고자 하였으며 이를 위해 소프트웨어 처리가 전혀없고 게이트 레벨로 設計하여 덧셈, 뺄셈, 나눗셈 등을 수행하도록 하였다. NTSC 신호발생 카메라를 사용하여 실험하였고, 水平 및 수직 동기 신호를 분리하여 전체 회로의 동기를 맞추도록 하였다.

本 論文의 構成은 AMBTC 理論을 간단히 소개한 후, 開發된 하드웨어 시스템의 개요를 설명하고 실험 결과를 보였다.

## II. AMBTC 理論

BTC (block truncation coding) 理論은 1979년 미국의 Delp와 Mitchell<sup>3)</sup>에 의해 처음 發表된 이후 1984년 Lema와 Mitchell<sup>4)</sup>이 計算이 간단하며 性能이 우수한 AMBTC 理論을 發表하였다. 이 방식은 2bit/pel의 데이터 符號率을 가지므로 PCM에 비해 1/4정도로 데이터를 압축시킨다.

먼저 디지털화된 映像을 (N×N) 크기의 겹치지 않는 블럭으로 나눈다. 각 블럭마다 平均( $\bar{m}$ )과 절대중심 모멘트(absolute central moment) ( $\bar{\alpha}$ )를 計算한다. 그 식은 (1)식, (2)식과 같다.

$$\bar{m} = \frac{1}{K} \sum_{i=1}^k X_i \quad (1)$$

$$\bar{\alpha} = \frac{1}{K} \sum_{i=1}^k |X_i - \bar{m}| \quad (2)$$

여기서  $X_i$ 는 각 화소의 gray level이다. 또, K는 한 블럭내의 화소의 수이다. 平均值를 임계값(threshold)으로 하여 각 블럭의 畫素를 1비트 量子化한다. 即, 平均보다 큰 gray level을 갖는 畫素는 1로, 평균보다

작은 gray level을 갖는 화소는 0으로 量子化하여 (1,0)으로 構成된 비트평면을 만든다. 이렇게 符號化된 블럭은 그의 平均( $\bar{m}$ )과 절대중심 모멘트( $\bar{\alpha}$ ) 및 비트평면을 수신측에 전송한다. 수신측에서는 각 블럭의  $\bar{m}$ 와  $\bar{\alpha}$ 가 保存되도록 畫像을 復元한다. 각 블럭마다 비트평면의 0은 'a'로, 1은 'b'로 復元시키도록 하면, 다음의 (3)식과 (4)식이 성립되어야 원래화상의 특성이 변하지 않고 保存된다.

$$K\bar{m} = qb + (k-q)a \quad (3)$$

$$K\bar{\alpha} = q(b - \bar{m}) + (k-q)(\bar{m} - a) \quad (4)$$

여기서 k는 한 블럭내의 총 화소의 수이고, q는 비트평면의 1의 숫자의 합이다.

(3)식과 (4)식을 利用하여 a, b를 구하면 (5)식과 (6)식이 얻어진다.

$$a = \bar{m} - \frac{\bar{\gamma}}{k-q} \quad (5)$$

$$b = \bar{m} + \frac{\bar{\gamma}}{q} \quad (6)$$

여기서  $\bar{\gamma} = K\bar{\alpha}/2$ 이다.

(5)식과 (6)식을 변형시켜 다른 표현식으로 쓰면 다음의 (7)식과 (8)식이 된다.

$$a = \frac{K\bar{m} - \sum_{i=1}^k X_i (X_i \geq \bar{m})}{k-q} \quad (7)$$

$$b = \frac{\sum_{i=1}^k X_i (X_i \geq \bar{m})}{q} \quad (8)$$

윗 식에서 알 수 있듯이  $K\bar{m}$ 은 한 블럭 전체 畫素의 gray level의 합이 되고,  $\sum_{i=1}^k X_i (X_i \geq \bar{m})$ 는 한 블럭내의  $\bar{m}$ 보다 큰 화소들만의 gray level의 부분합이다.

## III. 實時間動作 映像符號器/復號器의 開發

### 1. 시스템 構想

NTSC 표준신호는 1수평주사선의 주사시간은 63.5  $\mu$ s이고, 유효주사시간은 그의 약 80%인 52  $\mu$ s 정도이다. 현재 아날로그 방식의 화질을 유지하기 위해서는 1수평주사선당 500개 정도의 화소를 갖도록 해야 한다. 그렇게 하기 위해서는 약 10MHz의 표본화 주파수가 필요하다. AMBTC는 한 화소당 2비트의 부호화를 필요로 하므로 이 경우 약 20Mbps의 비트전송율이 소요된다. 그런데, 이와같은 해상도를 유지하려면 본 시스템의 처리속도는 표본화주파수의 2배가 필요하므로(III-2참고) 20MHz의 처리속도를 要求한다. 우리가 사용하고자 하는 소자가 汎用 TTL 소자이므로 이미 그 한계를 넘고 있다. 따라서, 본 시스템에서는

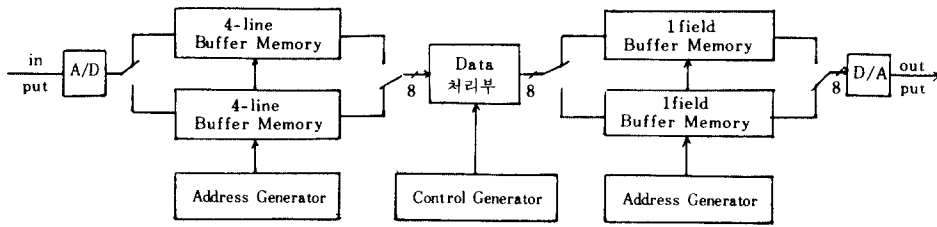


그림 1. 開發된 시스템의 全體블럭圖  
Fig. 1. Overall Block Diagram of the Developed System.

해상도를 반으로 감소시키는 대신에 입수가 용이한 소자를 선택할 수 있도록 하였다. 따라서 표본화 주파수를 5MHz로 하여, 비트전송율은 10Mbps가 된다. 또, AMBTC 理論에 의하면 부호기에서는 1블럭의 평균과 absolute moment를 비트평균과 함께 전송하면, 복호기에서는 이들 정보로부터 a와 b를 계산하도록 되어 있다. 그런데, 실용성의 면에서 볼 때, 復號器는 構成이 간단할 수록 좋기 때문에 a, b를 미리 계산하여 전송하면, 복호기에서는 비트평균의 (0, 1)에 따라 a, b를 절환하여 나타내기만 하면 되므로 復號器의 構成이 간단해진다. 따라서, 부호기 측에서 평균과 절대중심 모멘트 대신 a, b를 전송하는 方式을 채택한다. 데이터 처리가 블럭별로 되므로 블럭사이즈의 선정이 중요하다. 시뮬레이션 결과 (4×4)의 크기가 적당한데 이유는 다른 변환부호화(KLT, DCT, Hadamard 변환등)와 달리 블럭의 크기가 커져도 데이터符號率은 그다지 줄어 들지 않으며, 성능은 오히려 블럭이 커질 수록 열화된다.

5 MHz의 표본화주파수로 샘플링하면 1주사선당 256개의 화소를 얻게 된다. (4×4)의 블럭별로 데이터를 處理하므로 4수평주사선의 총 블럭 수는 64개가 된다. 實時間처리를 위해 1블럭은 3.8μs (63.5μs×4÷64)이내에 처리해야 한다. 3.8μs시간 동안 (7)식과 (8)식의 計算을 완료하는 것은 不可能하므로 並列로 처리해야 한다. (Ⅲ-3 실시시간 처리과정 참조) 데이터 處理部에서는  $\sum_{i=1}^m X_i$ 와  $\sum_{i=1}^m X_i (X_i \geq \bar{m})$ 의 計算을 해야하므로 같은 블럭의 데이터를 2번 읽어야 한다. 따라서, 버퍼메모리에서 읽는 클럭이 쓰는 클럭보다 2배 빨라야 실시간 동작이 가능하다. a와 b를 계산해 주기 위해 除算器가 필요하여 고속 演算소자가 필요할 것처럼 생각되나, 본 연구에서는 특수하게 설계된 binary divider에 의해 게이트 레벨의 하드웨어를 具現하였다. 부호화된 블럭의 데이터를 수신측에 多重化전송시키면 수신측에서는 1 프레임(512개의 주사선으로 구성)의 데이터를 저장하여 필드별로 번갈아 화상을 재현시키도

록 한다. 이것은 정지화상을 표시하기 위한 특별배러이다. 또, 이 모든 회로를 동기시켜 주는 회로가 필요하다.

### 2. 시스템 構成

開發된 시스템은 符號器(엔코더)와 復號器(디코더)로 나눌 수 있다. 符號器는 4走査線버퍼 메모리부와 데이터處理部로 구성되어 있다. 復號器는 2필드 버퍼 메모리와 어드레스 發生部로 構成된다. 그 밖에 A/D 변환, D/A 변환 및 클럭 발생부와 제어신호 발생부가 있다. 본 시스템은 5 MHz의 표본화율과 2bit/pel의 부호화율을 갖는다. 따라서, 10Mb/s의 비트 전송율을 갖는다. 전체블럭圖는 그림 1과 같다.

### 3. 符號器(Encoder)

#### 1) 4走査線 버퍼 메모리부

처리에 필요한 최소한의 기억용량은, (4×4) 처리를 위해 4-line의 기억이 필요하고, 또, 1수평走査線당 256 화소를 각 화소당 8비트로 양자화하면, 총 8킬로비트의 메모리 용량이 필요하다. 處理속도가 10MHz이므로, 고속의 RAM이 필요한데 본 시스템에서는 55ns의 액세스시간을 갖는 NMOS形 SRAM '2148'을 採用하였다. 그의 구조가 (1K×4)이므로 4주사선의 기억에 2개의 IC가 필요하고, 實時間동작을 위해 토글버퍼를 使用해야 하므로 SRAM은 모두 4개가 필요하다.

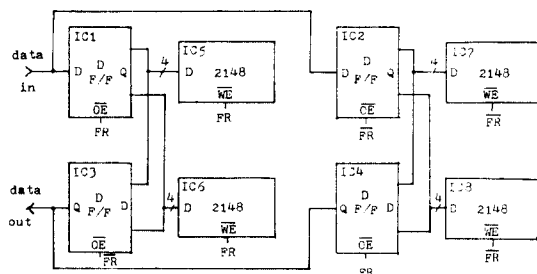


그림 2. 4走査線버퍼 블럭圖  
Fig. 2. Block Diagram of the 4-line Buffer.

그림 2 에 블럭圖를 보였다. IC1과 IC2의 D F/F은 入力데이터 同期용이고 IC3과 IC4의 D F/F은 出力데이터 同期용이다. IC5, 6 및 IC7, 8은 자기 4走査線의 기록 및 읽기를 번갈아 담당한다. 한개의 데이터는 4 주사선 마다 읽기와 쓰기를 교체한다. 이때 출력들이 서로 共通으로 연결되므로 출력 enable 단자를 使用하여 서로 충돌되지 않게 해야 한다. 또, 後段의 데이터 處理部에서 동일 블럭의 데이터를 2번 요구하도록 되어 있어서, 4 주사선 버퍼에 기록하는 클럭 보다 읽어내는 클럭이 2 배 빠른 주파수를 갖는다. 모든 動作이 수평동기 신호에 맞추어 리셋되므로 그에 必要한 제어신호와 어드레스 신호발생이 중요하다. 특히 버퍼에 기록할 때는 순차적으로 이루어지지만, 읽을 때는 블럭별로 읽기 때문에 어드레스 신호발생이 복잡해진다. 이 4走査線 버퍼 메모리에 使用된 주요 IC는 2148, 74574, 74163, 74112, 74257 등이다.

2) 데이터 處理部

符號器의 가장 核心部로 II 장의 (7)식과 (8)식의 計算과, 비트平面的 발생 및 데이터의 多重化(multiplexing)을 擔當한다. 데이터 處理部의 블럭圖를 그림 3 에 보였다. 또, 데이터 처리의 실시간 과정은 그림 4에 보였다. 이 데이터 處理部에서는 4 가지의 計算을 遂行하는데, 먼저  $\sum_{i=1}^{16} X_i$ 를 計算하고, 다음에  $\sum_{i=1}^{16} X_i(X_i \geq \bar{m})$ 를 計算한다. 이 結果를 利用하여 (7)식과 (8)식에 의해 두번의 나눗셈을 시행한다.

그림 3 과 그림 4 를 참고로 實時間動作을 살펴보면, 먼저 최초의 16클럭 기간 동안 첫 블럭의 16畫素의 값이 래치 1 (latch 1)을 통해 ALU에 의해 누적(accumulate)되어 총 화소의 合計가 래치 3 에 기억된다. 다음

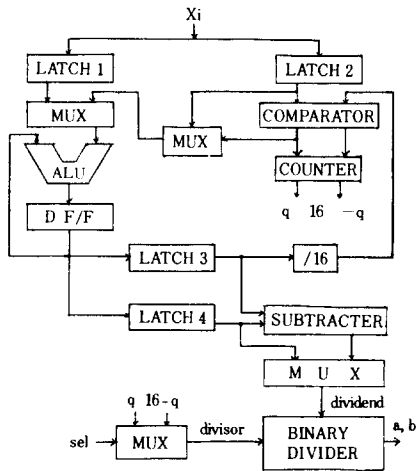


그림 3. 데이터 處理部 블럭圖  
Fig. 3. Block Diagram of the Data Processing Block.

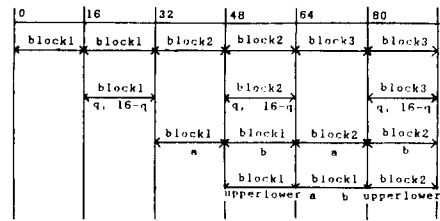


그림 4. 데이터處理의 實時間過程  
Fig. 4. Real Time Sequence of the Data Processing.

16클럭 기간 동안에 첫 블럭의 각 화소값이 이미 구해진 平均과 比較되어 平均보다 큰 畫素만 ALU를 통해 舍해져 래치 4 에 기억된다. 이 16클럭 기간 동안에 (1,0)의 비트平면과 q(平均보다 큰 화소의 갯수) 값이 同時에 얻어진다. 다음 16클럭 기간 동안에는 2 번째 블럭의 데이터가 읽혀져서 앞의 過程을 되풀이 하여 그 블럭의 平均値를 얻는다. 이 16클럭 기간 동안에는 앞의 32클럭 기간 동안 計算된 結果를 利用하여 첫번째 블럭의 a값이 減算器 및 除算器에 의해 구해진다. 특히 減算기는 subtract and shift 方法으로 構成되었다. A 를 計算하는데 16클럭이 必要하다. 다음 16클럭期間 동안에는 첫 블럭의 b를 동일회로에 의해 計算하여 구한다. 이같이 하여 얻어진 a, b 및 비트평면 출력은 多重化하여 복호기 쪽으로 傳送한다. a와 b가 8 비트, 비트평면이 16비트이므로 비트평면을 둘로 나누어 보낸다. 즉, 32클럭 기간 동안 8 클럭씩 나누어서 상위 비트 평면, 하위비트평면, a, b의 순서로 전송한다. 이 4 가지 데이터는 8 비트씩 병렬로 800ns기간 동안 전송한다. 첫번째 블럭이 전송되는 기간은 48번째 클럭부터 79번째 클럭까지이다. 이것은 최초로부터 4.8μs 지난 후이다. 즉, 計算에는 4.8μs가 必要된 것을 알 수 있다. 이 데이터處理部에 使用된 주요 IC는 74574, 74257, 7485, 74181, 74161, 74164, 74174 등이다.

4. 復號器 (Decoder)

그림 5 에 복호기의 블럭圖를 보였다. 복호기는 크게 3 부분으로 나눌 수 있다. 첫째로, 부호기에서 傳送되어온 a, b 및 비트平면을 기억시키는 버퍼메모리部가 있다. 2 필드分の 데이터를 저장하도록 設計하여 動畫像 뿐만 아니라 靜止畫像도 表示할 수 있도록 하였다. 1 프레임당 블럭의 수가 8,192 개이고, 1블럭당 32비트가 必要하므로, 총 내장 메모리 용량은 256킬로 비트가 된다. 이때 메모리의 액세스 시간은 400ns 이하이면 되고, CMOS形 SRAM '6264'를 採用하였다. 그의 構造가 (8K×8)이므로 모두 4 개의 SRAM이 必要하다. 實時間동작을 위해 2 개씩 나누어서 1필드씩의 읽

기와 쓰기를 담당한다. 그의 역할은 수직동기 신호에 의해 교체된다. 메모리의 번지를 효율적으로 할당하여 입, 출력시키는 것이 가장 중요하다. IC1과 IC2는 a, b를 저장하는데 사용되며 IC3과 IC4는 비트平面向을 저장하는데 사용된다.

둘째로, 비트평면 데이터에 의해 a, b를 선택하여 화상을 再現시키는 부분이 있다. 비트평면 데이터를 읽어서 並列-直列 변환해 준 다음 1 화소씩 표시해 준다. a, b는 D F/F에 의해 래치시킨다. 기억시킬 때는 블럭별로 하지만, 再現할 때는 走査線별로 읽어내야 하기 때문에 어드레. 신호발생이 복잡하다. 분주 IC에 의해 발생된 파형을 MUX시켜 사용한다. 水平동기와 垂直동기에 의해 분주기가 클리어되도록 하였고, 符號器로부터 傳送되어 오는 데이터의 동기를 맞춰주는 識別信號가 印加되어야 한다.

셋째로, 標本化주파수가 5 MHz로 낮게 선정된 理由로 해상도의 低下가 예상되어 수평方向으로 補間(interpolation)시켜 주는 회로가 있다. 畫素間的 中央에 兩畫素의 평균을 계산하여 삽입시킨다. 그 밖에 어드레스 발생부와 제어신호 발생부가 있다. 사용된 주요 IC는 6264, 74574, 74257, 74163, 74195, 74112, 74245 등이다.

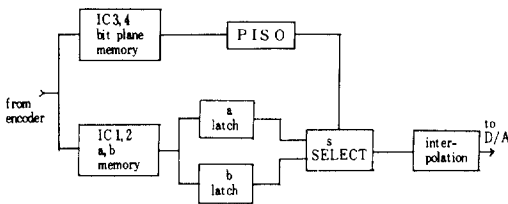


그림 5. 복호기의 블럭圖  
Fig. 5. Block Diagram of the Decoder.

5. 其他 回路

A/D변환, D/A변환, 前置필터, 復元필터, 수평 및 수직동기분리, 클럭발생, 영상신호合成 등이 이루어진다. 클럭은 수평동기신호의 정수배가 되도록 하여 화면상에 直交(orthogonal) 標本化패턴이 되도록 만들어 준다. A/D 변환클럭이 5 MHz이고, D/A 변환클럭은 10MHz이다. 데이터處理 클럭은 10MHz이다. 모든 回路의 동기는 수평 및 수직동기신호에 의해 이루어진다. 부호기의 데이터處理部에 필요한 제어신호는 타이밍이 매우 중요한데 TTL-IC의 게이트 지연시간을 고려하여 細密하게 設計하였다. A/D변환 및 D/A 변환은 삼성반도체에서 만든 단일칩 IC KSV-3100을 사용하였다. A/D 변환과 D/A 변환을 단일칩에서 할 수 있고

A/D 변환은 8 bit, 최대변환 속도가 20MHz 이상인 고속 IC이다.

IV. 시스템 性能評價

AMBTC 부호화 方式은 중간정도의 畫質을 보장하는 영상압축방법으로 2.0 bit/pel의 부호화율로 23~28dB 정도의 S/N을 나타낸다. 본 시스템은 실제의 카메라 신호를 利用했으므로 정확한 S/N은 알기가 어려우나 대체적으로 전산실험과 같은 S/N을 유지한다. 그러나, 약간의 차이는 예상할 수 있는데 그 이유로는 다음과 같은 것을 들 수 있다. 클럭이 H동기에 정확히 동기되어 있는지의 여부, 고주파의 영향으로 생기는 간섭현상, 제어신호의 timing오차, 게이트지연의 不一致 등을 들 수 있다. 화상에 미치는 영향은 edge의 jerk 현상, spike 잡음 발생, 화질의 해상도저하 등이 있다. 본 研究에서 具現한 시스템은 이상의 문제점을 극복하고 다양한 종류의 화상에 대해 共히 우수한 性能을 갖는다는 것을 實驗을 통해 확인하였다. 또, 원래 AMBTC는 1/4의 데이터壓縮을 이루도록 되어 있으나, 본 시스템은 표본화주파수를 1/2로 떨어뜨림으로써 1/8의 데이터압축을 시켰다. 이로 인해 해상도의 감소가 현저하게 발생하였다.

V. 實驗結果 및 檢討

컬러 비디오 카메라에서 발생한 복합영상신호로부터 RC 低域通過 여파기를 통해 약 2MHz로 대역제한된 흑백영상 신호를 얻는다. 이 信號를 A/D 변환하여 AMBTC 符號化하여 전송하고, 수신단에서는 復號器를 거쳐 화상을 재현하였다. 그림6 과 그림8은 原來의 화상을, 그림7 과 그림9는 再現한 화상을 보였다. 原畫像은 디지털적으로 80Mb/s의 전송율을 갖는 화상으로 볼 수 있고, 처리된 화상은 5 MHz로 標本化한 후 2 bit/pel로 符號化했으므로 10Mb/s의 전송율을



그림 6. 原色像  
Fig. 6. Original Image.



그림 7. 再現된 畫像  
Fig. 7. Reconstructed Image.

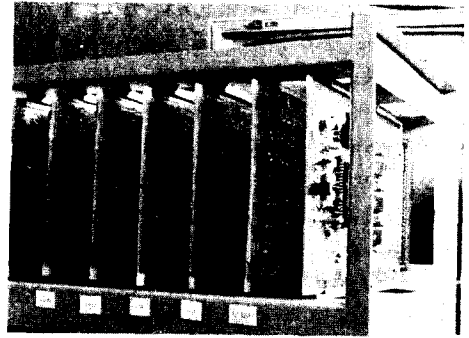


그림 10. 開發된 裝置의 모습  
Fig. 10. A View of the Developed System.

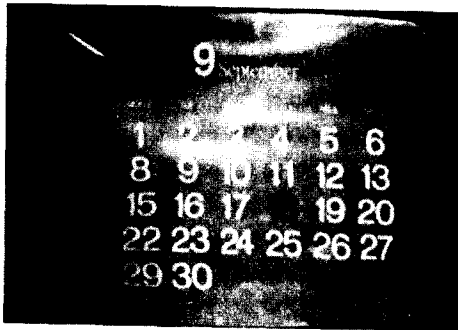


그림 8. 原畫像  
Fig. 8. Original Image.

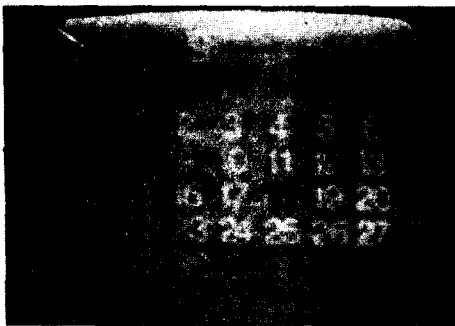


그림 9. 再現된 畫像  
Fig. 9. Reconstructed Image.

시행하여 그중에 문자와 虎圖를 결과로 보았다. 실험 장치의 全부는 그림 10에 보였다. 총 回路는 램핑으로 구성했으며, 7개의 基板과 300여개의 IC로 구성했다. 총 소비전력은 15W (5V, 3A) 정도이다.

VI. 結 論

본 연구에서는 映像데이터 壓縮을 위한 absolute moment block truncation coding을 이용하여 實時間에서 動作하는 하드웨어 시스템을 具現하고, 黑白映像信號를 使用하여 實驗하였다. 10Mbps의 데이터 전송율로 전송하여 비교적 優秀한 畫質을 維持할 수 있었다. 영상부호화 장치의 실시간 구현의 복잡성과 난이성을 극복하고 새로운 시스템을 提案하였다. 소프트웨어 處理가 전혀없이 하드웨어적으로 모든 計算을 수행하며, 프레임메모리를 低速의 SRAM으로 구성하였다. 이는 부호화율이 2bit/pel이고 표본화율이 5 Msample/sec가 되어 10Mbit/sec의 비트전송율을 갖기에 가능하다. 수신측에서는 최종段에 interpolator를 添加하여 해상도低下를 보상하였다. PCM으로 영상을 전송할 때에 비해 1/8의 비트전송율로 그에 相應하는 性能을 얻었고, 장차 수개의 칩으로 집적하면 실용화 할 수 있다.

參 考 文 獻

갖는다. 再現된 畫상을 보면 畫質의 劣化가 있는데 이는 표본화주파수가 낮기 때문에 不可避하다. 또, 줄무늬가 눈에 띄는데 이것은 컬러複合映像 신호로부터 黑白신호를 분리하는 過程에서 컬러 副搬送波 (color sub-carrier)의 不完全한 除去로 인한 現象이며 장치내부에서 발생한 것은 아니다. 畫상은 여러 종류에 대해

[1] A.N. Netravali and J.O. Limb, "Picture Coding: A Review," *Proc. IEEE*, vol-68, pp. 366-406, 1980.  
[2] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*. 2nd ed., vol. 1, Academic Press, pp. 166-208, 1982.  
[3] M.P. Ekstrom, *Digital Image Processing*

- Techniques*. Academic Press, pp. 172-226, 1984.
- [4] E.J. Delp and O.R. Mitchell, "Image Compression using block truncation coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 1335-1342, Sept. 1979.
- [5] M.D. Lema and O.R. Mitchell, "Absolute moment block truncation coding and its application to color images", *IEEE Trans. Commun.*, vol. COM-32, pp. 1148-1157, Oct. 1984.
- [6] H.H. Ko and C.W. Lee, "Real time implementation of block truncation coding for picture data compression," *Proc. Int. Conf ASSP-87*, Apr. 1987.
- [7] W.K. Pratt, *Digital Image Processing*. Wiley, pp. 140-161, 1980.
-