

# 비트 변환방식을 이용한 데이터 취득에 관한 연구

박 상 길

(부산수산대학)

(1986년 1월 17일 수리)

## A Study on the Data Acquisition by Bit Conversion Method

Sanggil PARK

National Fisheries University of Pusan

(Received January 17, 1986)

This paper deals with a new bit conversion method. When 12 bit A/D converter is adapted to 16 bit micro-computer, complicated data acquisition method is not necessary to acquire the A/D converted data into memory of computer.

However, when the 12 bit A/D converter is adapted to the 8 bit micro-computer 12 bit data should be divided into 4 bit data and 8 bit data. Therefore the old data-dividing method made 4 bit/2byte of memory space wasted.

On the contrary, using the new bit conversion method suggested in this paper the two of 12 bit data are converted into 3 byte of data without extending the A/D conversion time.

### 서 론

계측과 제어분야에 점차로 마이크로 컴퓨터가 많이 이용되고 있다. 이때 이용되는 A/D변환기는 8비트, 12비트, 16비트가 주류를 이루고 있는데 이 중 8비트 A/D변환기는 분해능(resolution)이 저조하고 16비트 A/D변환기는 분해능은 좋으나 변환시간과 가격면에서 볼 때 12비트 A/D변환기에 못미치므로 많이 응용되지 않는다. 따라서 분해능과 변환시간, 가격면에서 유리한 12비트 A/D변환기가 많이 응용되고 있다<sup>1)</sup>.

12비트 A/D변환기를 16비트 계산기에 응용할 경우에는 복잡한 처리를 하지않고 간단히 데이터(data)를 메모리 내에 격납할 수 있다<sup>2)</sup>. 그러나 현재로서는 8비트 마이크로 컴퓨터에 의해 데이터를 처리하는 경우가 많다. 이 경우 12비트의 데이터를 2바이트(byte)의 데이터로 분리하여 저장하는 방식을 채용하게 된다<sup>3)</sup>. 이때 4비트는 필연적으로 데이터가 없는 빈 공간으로 허비하게 된다. 이와 같은 메모리 공간의 낭비를 절약하기 위하여 본 연구에서는 12비

트의 데이터 2개를 A/D변환기의 변환시간에 제약을 주지 않고 3바이트의 메모리 공간에 격납하는 비트 변환방식을 제안한다.

### 비트 변환기의 작동원리

비트 변환기의 작동원리도는 Fig.1과 같다. 먼저 인접한 2개의 12비트 데이터를 MSB(most significant bit) 측으로부터 5번째 비트에서 분리하여 4비트와 8비트의 데이터로 만든다. 그러면 Fig.1의 두번째 단계와 같이 1니블(nibble)과 1바이트의 데이터가 각각 2개 만들어진다. 이 때 만들어진 1니블 데이터 2개를 합하여 1바이트의 데이터로 만들게 된다. 이와 같은 과정을 거치게 되면 각각의 2개의 12비트 데이터는 3개의 8비트 데이터로 변환되게 된다. 이와 같은 방식을 12비트 A/D변환기에 적용하여 8비트의 데이터로 변환하게 되면 8비트 계산기로 메모리의 허비없이 신호처리를 행할 수 있다. Table 1은 12비트의 데이터를 8비트의 데이터로 변환시키는 예를 나타낸다.

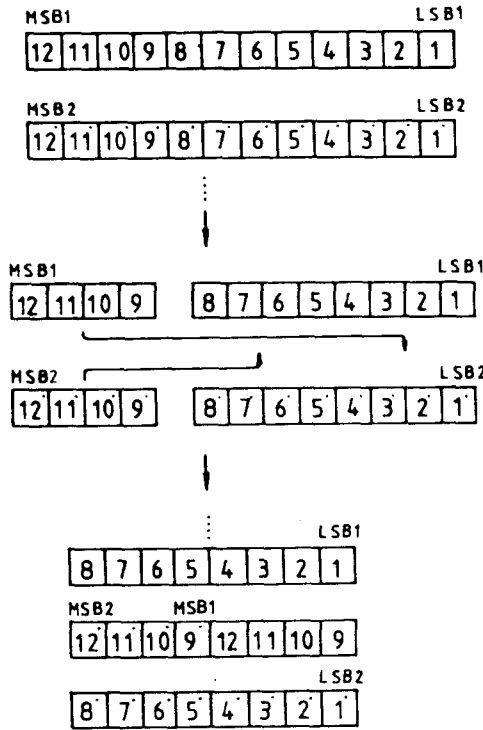


Fig. 1. Operational principle diagram of bit converter.

Table 1. Example of bit conversion

12 bit data	8 bit data
A10	10
	EA
E11	11
B09	09
	EB
EAF	AF
⋮	⋮

이와 같은 방식에 의하여 메모리에 격납된 데이터는 소프트웨어(software)에 의해서 원하는 처리를 할 수 있다.

이상과 같은 원리에 부합하도록 하기 위해서는 Fig. 2와 같은 시스템(system)이 필요하다. 이 시스템의 모든 레지스터(register)는 3 state output이 가능해야 할 필요가 있다. 이 그림에서 SOC(start of conversion) 신호가 가해져서 홀딩(holding)된 아날로그(analog) 신호를 A/D변환기가 디지털 신호로 변환시키게 되면 A/D변환기는 EOC(end of conversion) 신호를 발생한다. 이 신호를 MMV(monostable multivibrator)에 통과시켜서 12비트 레지스터에 가하여, 디지털 신호를 다음 EOC가 발생하기 까지 기억하도록 한다. 이 데이터 중 하위 8비트는 바로 8비트 레지스터에 공급하고, 상위 4비트는 공통으로 2개의 4비트

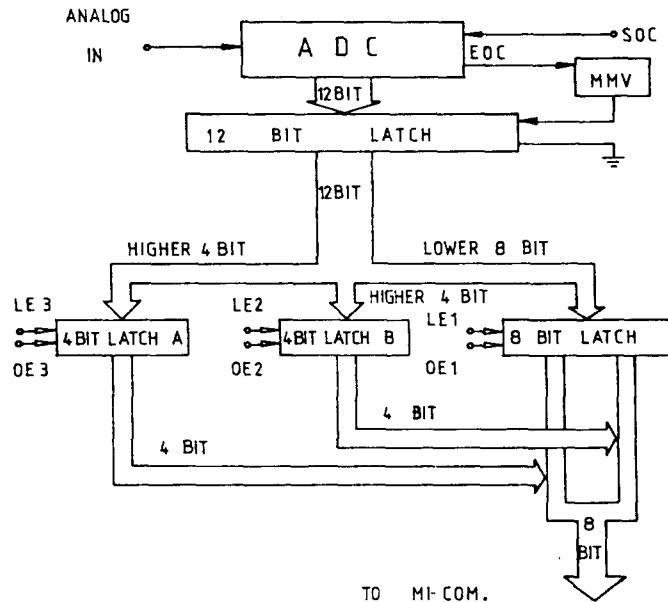


Fig. 2. System diagram of bit converter.

박 상 길

레퀴에 공급한다. 다음에는 8비트 래치와 4비트 래치 A의 LE 단자에 펄스를 가하여 데이터의 하위 8비트와 상위 4비트를 기억시킨다. 그리고 OE1 단자에 펄스를 가하여 계산기의 메모리에 데이터를 격납시킨다.

다음의 EOC 신호가 발생되어 새로이 A/D변환된 12비트 데이터가 12비트 래치에 기록되면 LE1과 LE3 단자에 펄스를 가하여 상·하위 데이터를 각각 기억시킨다. 그 후 OE2, OE3에 펄스를 가하여 먼저번 12비트 데이터의 상위 4비트와 새로운 12비트 데이터의 상위 4비트를 합한 8비트의 데이터를 계산기의

메모리에 전송한다. 그 후 다음 EOC 신호가 들어오기 전까지 OE1에 펄스를 가하여 새로운 12비트 데이터 중의 하위 8비트를 계산기에 전송하여 결국 2개의 12비트 데이터를 3개의 8비트 데이터로 변환하여 계산기의 메모리에 격납할 수 있다.

실 험

이상과 같은 원리에 의하여 설계된 A/D변환기 및 비트변환기의 회로도는 Fig. 3와 같다. 실험에 사용된 계산기는 Apple II였다. 이 기종의 6번 슬롯(slot)

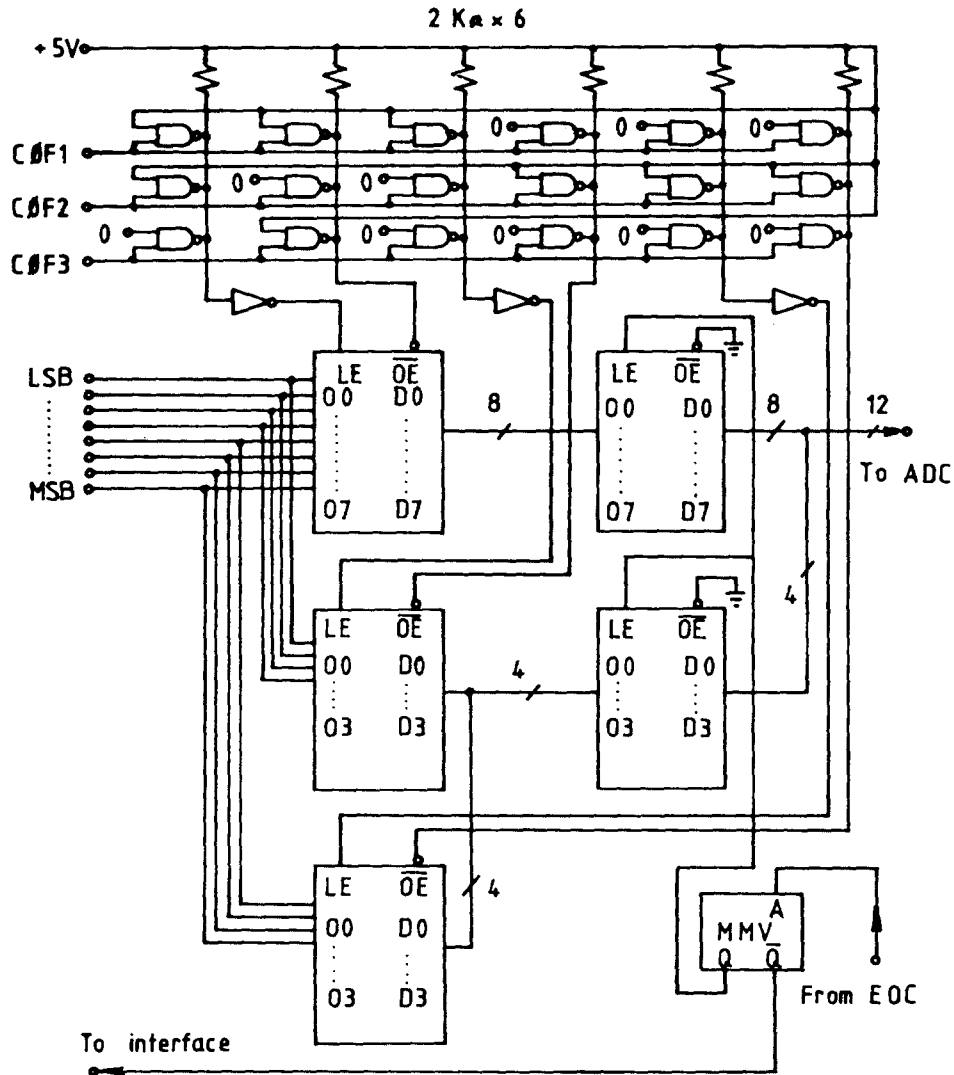


Fig. 3. Circuit diagram of bit converter.

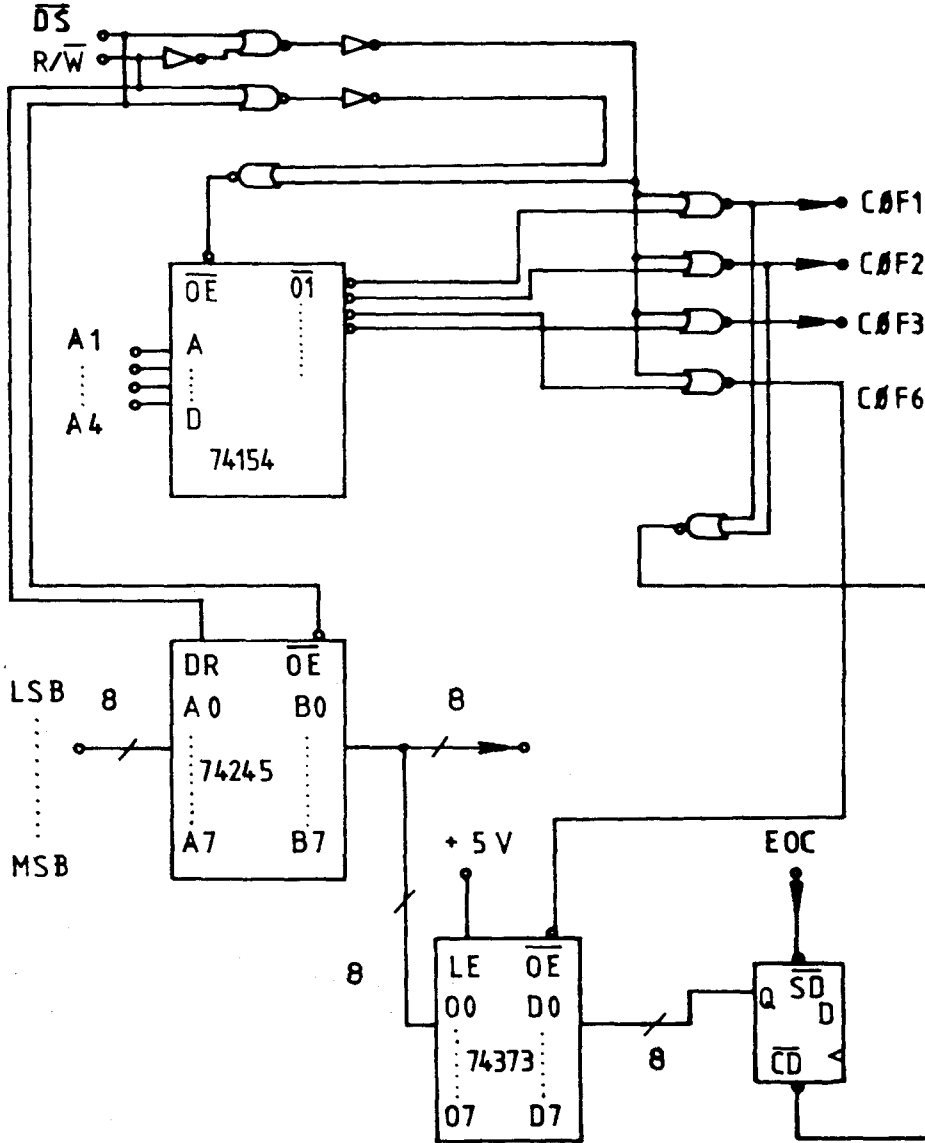


Fig. 4. Circuit diagram of the interface.

전용 인터페이스(interface)를 제작하였을 때의 회로도는 Fig. 4와 같다. 이와 같은 회로를 조합하여 구성한 실제실험에 이용된 시스템은 Fig. 5와 같다.

인터페이스회로에서는 74154 디코더(decoder)에 의해서 4비트의 번지(address)를 디코딩(decoding)한 후 Apple의 DS(device selector) 및 read/write 단자의 신호를 조합하여 각각의 제어신호(control signal)를 선별하여 정하였다.

즉 C/F1이 호출될 때 하위 8비트 데이터가 계산기로 격납되고 C/F2가 호출될 때 2개의 상위 4비트 데이터가 계산기로 격납된다. 그리고 C/F3가 호출

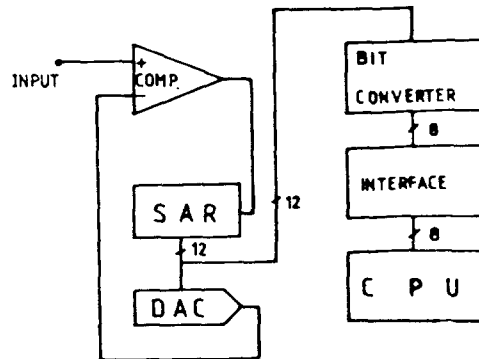


Fig. 5. Experimental system diagram.

박 상 길

될 때 다음 12비트 데이터의 8비트가 격납되게 된다.  
인터페이스회로의 버스 트랜시버(bus transceiver)는 버퍼(buffer)역할도 아울러 하도록 되어 있다.

이와 같은 데이터 격납작용을 위한 비트 변환기의 타이밍(timing)을 표로 만들어 보게 되면 Table 2와 같이 된다.

Table 2. Timing table of bit converter

Address	Lower 8bit	Upper 4 bit of latch A	Upper 4 bit of latch B
Process 1 C0F1	(LE, X)	(LE, X)	(X, X)
Process 2A C0F2	(LE, X)	(X, OE)	(LE, OE)
Process 2B C0F3	(X, OE)	(X, X)	(X, X)

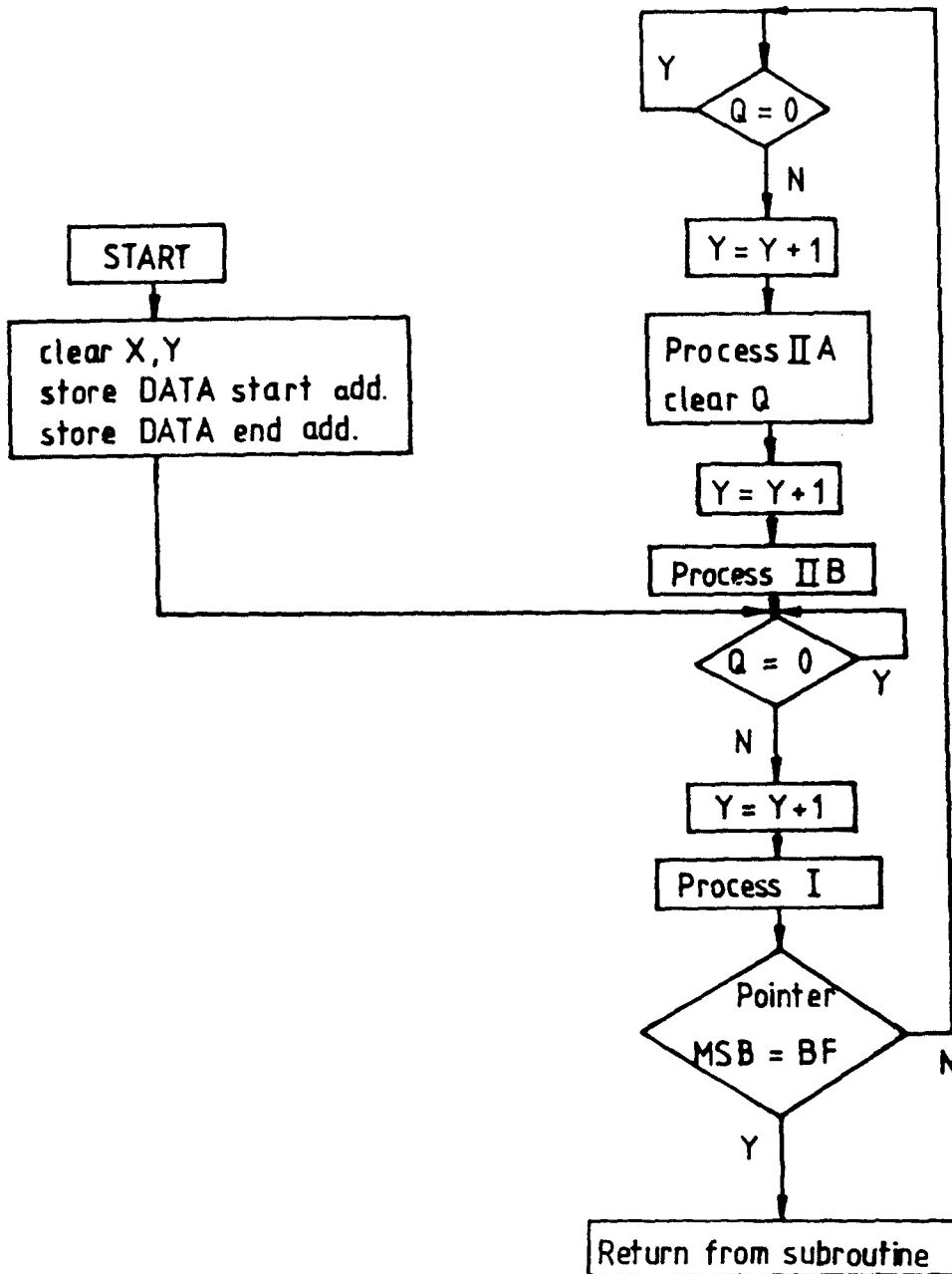


Fig. 6. Flow sheet of the system diagram.

이상과 같이 구성된 시스템을 가동시키기 위한 어셈블러 프로그램(Assembler program)의 플로우 시이트(flow sheet)는 Fig.6과 같다. 이 프로그램 중에는 포인터(pointer)를 사용한 indirect addressing mode를 이용하여 프로그램의 실행시간을 가능한 한 단축하였다<sup>6)</sup>. 사용된 포인터의 번지는 00FA, 00FB였다.

이 프로그램에서는 하나의 8비트 데이터가 계산기에 격납되면 Q를 1로 만들도록 되어 있는데 이는 다음의 데이터 호출 프로그램이 A/D변환기가 데이터를 결정하기 이전에 실행될 경우를 방지하기 위함이다.

즉, 프로그램 실행중 Q가 0이면 루이핑(looping)을 계속하다가 A/D 변환기가 데이터를 결정하여 Q가 1이 되면 프로그램이 계속되어 데이터를 계산기의 메모리에 격납하게 된다.

일단 프로그램이 실행되면 먼저 Q의 값을 테스트(test)하고 Q가 1이면 다시 Y를 1만큼 증가시킨다. 그리고 2개의 상위 4비트를 메모리에 격납하고 Q를 1로 만든다. 다시 Y를 1만큼 증가시키고 Q의 값에 관계없이 8비트 데이터를 격납하고 원래의 위치로 되돌아가서 프로그램을 계속 실행한다.

여기서 A/D 변환기는 +2.5V를 FFF, -2.5V를 000, 0V를 800로 결정(calibration)하였다.

### 실험결과 및 고찰

실험시 제작된 A/D 변환기의 최고 속도는 40  $\mu\text{sec}$ 였으며 정도는  $\pm 1$  LSB였다. 이는 1 chip A/D 변환소자의  $\pm 1/2$  LSB에 비해서 정도가 떨어지는데 그 이유는 잡음에 의한 것으로 본다. 프로그램의 각 데이터취득 프로세스(process) 중 최대의 것이 44  $\mu\text{sec}$ 였다. 이것은 Apple의 동기 주파수에 의한 시간과 기계어 사이클(cycle)의 곱인  $1/1.023 \text{ MHz} \times 44 \text{ machine cycle}$ 과 매우 근사하게 일치함을 알 수 있었다. 실험시 최대 데이터 취득속도는 22.7 K samples/sec였는데 이는 프로그램의 속도제한 시간인 44  $\mu\text{sec}$ 의 역수인 22.727 K samples/sec와 잘 일치함을 알 수 있었다.

A/D 변환기의 속도제한 요인을 시험하기 위해 각 부의 시간지연을 시험한 결과 비교기(comparator)의 비교신호가 입력된 후 원활한 판단을 하기까지 2.9  $\mu\text{sec}$  가소요되었는데 이 기간이 제일 길었다. 따라서 successive approximation 형의 A/D 변환기의 변환속도는 주로 비교기의 응답시간 지연에 기인한다는 것을 알 수 있었다.

계산기 자신이 주변기기를 활용하는 동시에 A/D 변환을 행할 경우에 인터럽트 요구(interrupt request) 방식을 사용하면 수 msec의 시간이 소요되나<sup>7)</sup> 소프트웨어(software)와 하드웨어(hardware)가 상호보완을 한 본 A/D 변환방식의 경우는 44  $\mu\text{sec}$ 의 변환시간이 소요되었으므로 속도면에서 인터럽트를 사용하는 방식에 비해 유리한 것을 알 수 있었다.

그리고 본 시스템을 사용하면서 자기 디스크를 사용할 경우, Apple II의 구조상 디스크 드라이버(disk driver)에 가동명령을 한 이후 BASIC 언어를 사용가능케 하기 위해서는 03F2에 03를, 03F3에 E0를 격납하고 FB6F에서 시작되는 프로그램을 실행하여 소프트웨어 엔트리 벡터(software entry vector)를 고정할 필요가 있었음을 부연하여 둔다.

### 결 론

하드웨어(hardware)적인 복잡함을 피하기 위해서 인터럽트 요구방식이나 DMA(direct memory access)방식을 채용하지 않고, 소프트웨어의 도움에 의한 A/D 변환을 행하였으며, bit 변환장치를 이용하여 메모리의 낭비를 방지한 결과 다음과 같은 결론을 얻었다.

- i) 어셈블러 프로그램을 이용한 데이터 취득방식을 이용하여 44  $\mu\text{sec}$ 의 취득시간(acquisition time)이 가능했다.
- ii) successive approximation 형 A/D 변환기의 변환속도는 주로 비교기의 비교속도에 의존한다.
- iii) 12비트 데이터를 8비트 계산기의 메모리에 격납할 경우, 비트변환방식에 의한 격납방식이 종래의 단순분할방식에 비해 1/4의 메모리영역을 절약할 수 있었다.

### 참 고 문 헌

- 1) Hae-sung Lee and David A. Hodges(1983): Self-Calibration Technique for A/D Converters. IEEE Transactions on Circuit and Systems, Vol. CAS-30, No. 3, 188-190.
- 2) Marcelo Tapia(1983): A Microprocessor-Controlled Flying-Spot Scanner with an Intelligent A/D Converter unit. IEEE Transactions on Instrumentation and Measurement, Vol. IM-32, No. 4 491-496.
- 3) 小倉勝・佐野陽子(1985): マイクロコンピュータ

박 상 길

- を用いたエンジン燃焼解析装置の開発. 内燃機関  
24巻, 10號, 9-14.
- 4) Robert L. Forgacs and Ber Abraham (1981):  
Microprocessor-Controlled Drill Analyzer. IEEE  
Transactions on Instrumentation and Measure-  
ment Vol. IM-30, No. 4 258-262.
- 5) 五味努・本望行雄(1977): 壓力平衡形指壓計. 内  
燃機関の實驗と計測. 105-121.
- 6) Rodney Zaks(1980): Programming the 6502.  
Sybex, 3rd ED. 112-187.
- 7) Richard C. Hallgren(1982): Interface Projects  
for the Apple II. Prentice-Hal, Inc. 50-69.