

## TMS320C31을 이용한 QPSK 모뎀 구현

## Implementation of QPSK Modem using TMS320C31

김광호 · 김종욱 · 조병모\* · 김영수

Kwang-Ho Kim · Jong-Wook Kim · Byung-Mo Cho\* · Young-Soo Kim

## 요 약

본 논문에서는 TI(Texas Instrument)사의 범용 DSP 프로세서인 TMS320C31을 이용하여 통신 시스템에서 많이 사용되는 QPSK 방식의 모뎀을 구현하였다. 지금까지 거의 모든 시스템의 신호 변환 과정은 하드웨어로 구성되어 있지만, 본 논문에서 구현된 시스템은 QPSK 신호의 변조과정에서 IF단의 DAC를 통과하기 이전까지의 과정과 복조과정에서 IF단의 ADC를 통과한 이후의 과정을 프로그램으로 구성하고, 신호의 입·출력부와 처리부분을 하드웨어로 구성하였다. DSP 프로세서를 이용한 모뎀 출력 결과를 PC 상에서 시뮬레이션 결과와 비교하여 제작한 모뎀이 정상적으로 동작됨을 확인하였다.

## Abstract

In this paper, we implemented QPSK(Quadrature Phase-Shift Keying) modem which is widely used for communication systems, using a general Digital Signal Processor(DSP), TMS320C31. Up to now, almost all of communication systems consist of hardware. However, the implemented system herein is composed of software and hardware part. Software part includes the modulation process, before passing D/A(Digital-to-Analog Converter) and the demodulation process, after passing A/D(Analog-to-Digital Converter) in IF(Intermediate Frequency) node. Hardware part is related to input, output and process of signal. To demonstrate the successful implementation of modem, the output results obtained from DSP processor are compared with the simulated result on the personal computer.

## 1. 서 론

오늘날 사회 구조가 정보화 사회로 변모되어감에 따라 통신의 역할이 종합적인 정보 시스템으로 변화되어 가고 있으며, 이를 위한 신호 처리 및 전송 방법이 점차 디지털화되어 가고 있다. 아날로그 통신에 비해 신뢰도가 높고 기밀 유지가 용이한 디지털 통신으로 전환되고 있다. 이 디지털 통신은 대체로 디지털 회로 또는 디지털 신호 처리의 영역을 RF(Radio Frequency) 주파수까지 직접적으로 확대

하고, RF부는 단순한 신호 증폭과 주파수 변환만을 수행하도록 그 범위를 가능한 축소시킨 것이다. 즉 RF부분을 제외한 동조, 채널 선택, 그리고 원신호 복구 등의 모든 부분을 디지털 신호 처리 기술을 이용하여 구현하고 있다.

언제, 어디서나, 누구에게나 다양한 통신 수단을 통해 정보를 제공하는, 즉 정보의 입·출력의 역할을 하는 것이 유·무선의 모뎀이다. 특히 무선 통신에서 일시에 불특정 다수인에게 대량의 정보를 동시에 제공하는 방송이나 정확한 음질과 다양한 정보의 부가

경희대학교 전자공학부(School of Electronics & Information, Kyunghee Univ.)

\*순천대학교 전자공학과(Dept. of Electronic Engineering, Suncheon National Univ.)

· 논문 번호 : 20010528-071

· 수정완료일자 : 2001년 6월 28일

서비스를 제공하는 이동통신에서는 데이터를 변조하고 복조하는 모듈의 역할은 점차 증대하고 있다.

현재 이동통신 환경에서 사용되는 모듈은 변조와 복조부로 나누어져 있으며 이 부분들이 하드웨어적으로 구성이 되어있다. 하지만 최근에 새로이 부각되는 SDR(Software Defined Radio) 개념에서는 이러한 부분들을 전부 소프트웨어적으로 구성을 한다. 따라서 무선 시스템의 경우 신호의 송신과 수신 과정에서 ADC(Analog-to-Digital Converter), DAC(Digital-to-Analog Converter), Memory, 그리고 신호 처리와 응용프로그램의 운영을 위한 프로세서를 제외한 나머지 신호의 변·복조의 과정을 프로그램으로 구성한다. 디지털 이동통신 시스템에서 사용되는 변복조 방식에는 QPSK, OQPSK, MSK, GMSK 등이 있으며, 이는 모두 반송파의 위상을 변화시키는 위상 편이변조(PSK) 방식에 속한다. PSK 과는 일정한 포락선 (또는 진폭)을 갖는 파형이기 때문에 전송로 등에 의한 진폭변동의 영향이 적으며 심볼 에러(symbol error) 측면에서도 우수하다. 뿐만 아니라, 피변조파는 양측대파 신호이기 때문에 타이밍 정보 및 주파수 정보를 포함하고 있어, 변복조 회로가 간단한 장점이 있다<sup>[1]</sup>.

본 논문에서는 선형변조 방식의 PSK 계열 중 가장 기본적인 이론인 QPSK 신호의 변·복조 과정을 DSP 프로세서를 이용하여 구현한 내용을 기술한다. 이를 위하여 제2장과 제3장에서는 시스템의 전체적인 프로그램과 제작에 관련된 내용을, 제 4장에서는 시뮬레이션과 실제 보드의 실험 결과를 기술하고, 끝으로 제 5 장에서 결론을 맺는다.

## II. 시스템 구성(응용 프로그램의 구조)

### 2-1 QPSK 변·복조과정의 프로그램 구성

그림 1은 본 논문에서 고려한 QPSK 변·복조 부분의 흐름도이다. 발생된 데이터를 I채널과 Q채널로 디코딩하여 나눈 다음 그레이 코드로 변환된 데이터를 0 → +1, 1 → -1 즉, bipolar 신호형태로 변환된다. 변환된 신호는 interpolator를 통하여 4배 up-sampling된 후, 제곱근 상승코사인 나이퀴스트(SRCN : Square-root Raised Cosine Nyquist)필터에

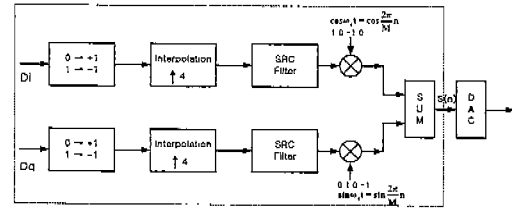


그림 1. QPSK 변조 과정

Fig. 1. QPSK modulation.

인가된다. 여기서, 4는 SRCN 필터를 Nyquist sampling rate보다 2배 높게 샘플링한 샘플치로 구현하기 위해 즉, 한 주기(T)를 4 샘플을 갖도록 하기 위해 선택된 값이다.

SRCN 필터의 탭 계수들은 필터 탭 중앙에 대해서 대칭성을 가지므로 선형위상의 특징을 가진다. 일반적인 필터의 출력 값은 입력데이터  $x[n]$ 와 필터 계수  $h[n]$ 의 컨볼루션으로 이루어진다. N탭 상승코사인 필터의 출력  $y[n]$ 은 식 (1)과 같이 표현된다.

$$y[n] = \sum_{i=0}^{N-1} h[i] x[n-i] \quad (1)$$

사용된 제곱근 상승코사인 나이퀴스트 필터의 임펄스 응답식은 식 (2)와 같다.

$$g[n] = \frac{\sin\left[\frac{\pi n}{T}(1-\alpha)\right] + \frac{4\alpha n}{T} \cos\left[\frac{\pi n}{T}(1+\alpha)\right]}{\frac{\pi n}{T} \left[1 - \left(\frac{4\alpha n}{5T}\right)^2\right]} \quad (2)$$

여기서 T는 심볼 주기이며, α는 초과 대역폭 파라미터 값으로 본 논문에서는 0.2를 사용하였다.

QPSK 변조부 모델이 그림 1에 나와 있다. 그림에서 반송파 IF주파수가 심볼율의 4배가 되도록 정하면 변조된 신호는 아래의 수식으로 나타낼 수 있다.

$$S[n] = I[n] \cos\left(\frac{\pi}{2} n\right) + Q[n] \sin\left(\frac{\pi}{2} n\right) \quad (3)$$

여기서 S[n]은 변조된 IF 신호이고 I[n]과 Q[n]은 각각 제곱근 상승코사인 필터를 거친 I 채널과 Q 채널 신호를 나타낸다. 식 (3)을 자세히 관찰하면, 코사인과 사인 신호들은 1, 0, -1, 0,.... 그리고 0, 1,

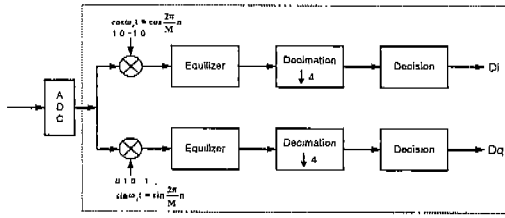


그림 2. QPSK 복조 과정  
Fig. 2. QPSK demodulation.

0, -1, ...의 값을 가지므로 변조되는 신호는 +(I 채널 신호), +(Q 채널 신호), -(I 채널 신호), -(Q 채널 신호),.....의 순서대로 출력됨을 알 수 있다. 그러므로 IF 대역으로의 믹싱은 곱셈연산이 되는 것이 아니라 간단히 해당 필터 출력의 부호를 바꾸어 주기만 하면 된다. 또한  $n$ 값이 짝수일 경우에는  $S[n]$ 은 I 채널 신호만 사용되고,  $n$ 이 홀수일 경우에는  $S[n]$ 은 Q 채널의 신호만 사용한다. 이를 이용하면 I 채널의 필터와 Q 채널의 필터를 반으로 줄일 수 있다<sup>[2]</sup>.

그림 2는 프로그램으로 구성될 복조 과정의 흐름도이다. 먼저 A/D 변환기를 통과한 신호에 변조단에서 했던 것처럼 I채널과 Q채널에 1, 0, -1, 0,.... 그리고 0, 1, 0, -1,....의 값을 곱하여 I 채널과 Q 채널로 데이터를 분리한다.

분리되어진 신호를 상승코사인 필터와 시시각각 변하는 채널로 생기는 왜곡과 잡음의 효과를 교정하기 위해 적응 등화기를 사용하였다. 등화기는 탭 계수를 조절할 수 있는 LMS(Least Mean Square) 알고리즘을 사용하였다. 그림 3은 LMS 알고리즘을 이용한 등화기의 구조이며 식 (4)는 알고리즘 수식이다.

이렇게 나온 값을 Decimation을 한 후 나온 값을 "0"보다 크면 "1"의 값을 "0"보다 작으면 "-1" 값으로 Decision해서 원래의 I 채널과 Q 채널의 데이터를 복원하는 과정을 전부 프로그램으로 구성하였다.

$$\begin{aligned}
 y[n] &= x[n]^T \hat{w}[n] \\
 e[n] &= a[n] - y[n] \\
 \hat{w}[n+1] &= \hat{w}[n] + \mu e[n] x[n]
 \end{aligned}
 \tag{4}$$

$y[n]$  : 등화기의 출력

$w[n]$  :  $k$ 번째 탭에서의 가중치  
 $e[n]$  : 오차신호  
 $a[n]$  : 원하는 출력  
 $\mu$  : 스텝 크기 매개변수

$$\begin{aligned}
 x[n] &= [x[n+N], \dots, x[n+1], x[n], \\
 &\quad x[n-1], \dots, x[n-M]]^T \\
 &: \text{등화기 탭 입력}
 \end{aligned}$$

### 2-2 호스트 프로그램

호스트 프로그램은 크게 두 가지 역할을 하는데 데이터 전송과 DSP 프로세서에서 실행되는 소프트웨어와의 통신을 맡는다.

호스트 프로그램은 NI(National Instrument)사의 LabWindows/CVI 프로그램 툴을 이용하여 메뉴 구조를 구성하였다. "DownLoad" 메뉴 아래에는 "Program"과 "Data" 메뉴가 있으며 이 메뉴 아래에서 프로그램이나 데이터를 Download 할 수 있다. 데이터는 Binary, Hexa 혹은 Float 형식을 다운로드 할 수 있다. "Upload" 메뉴 아래에는 "Data to Display"와 "Data to File" 서브메뉴가 있으며 DSP 보드에서 처리된 데이터를 화면에 디스플레이 하거나 또는 PC에 파일로 저장할 수 있게 구성되어 있다. 데이터 형식에 따라서 Binary, Hexa 혹은 Float 형식을 선택하여 저장할 수 있다. 그리고 DSP 프로세서를 Run, Stop, Reset 시킬 수 있는 메뉴들이 있다. 아래에 DSP 보드를 구동시킬 때의 초기상태를 표시해 주는 창을 나타내었다.

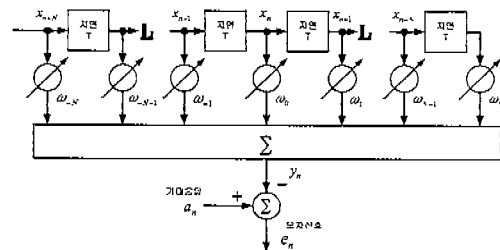


그림 3. 적응 등화기 요소  
Fig. 3. Adaptive equalizer component.

### 2-3 타겟 프로그램 구성

타겟 프로그램의 구성은 DSP 프로세서에서 동작하는 것으로서 그림 4에서 보는 것과 같이 초기화 부분에서는 Memory의 초기화, Timer 설정, Serial port의 설정을 하고 PC에서 발생한 데이터를 QPSK 변조를 한다. 그 다음에 D/A 변환기에 공급될 타이머에 대한 인터럽트를 체크하여 만약에 인터럽트가 걸리면 D/A 변환기를 통과한 데이터는 잡음을 첨가하여 다시 A/D 변환기를 통과하여 나온 리턴값을 가지고 다시 QPSK 복조 과정을 거쳐서 Decision 값으로 원래의 데이터를 복원한다.

### 2-4 호스트와 DSP 보드간의 데이터 통신

PC에서 발생시킨 데이터를 DSP 보드에서 변조한 후 메모리에 저장하고 D/A 변환기를 통하여 채널로 보내진 후 다시 A/D 변환기를 사용하여 획득한 데이터를 DSP 보드에서 호스트 응용프로그램으로 전송하는 프로그램이다. 다음에서 그 과정을 순차적으로 설명하겠다.

#### ◆ DSP 보드에서의 동작 과정.

- ① DSP 보드, 변수를 초기화한다.
- ② 미리 지정된 샘플링 주파수에 대해 타이머를 셋업한다.

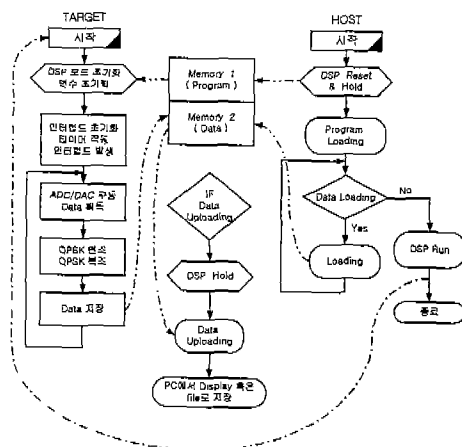


그림 4. 타겟과 호스트 프로그램 흐름도  
Fig. 4. Flowchart of host and target program.

- ③ PC로부터 데이터와 프로그램을 DSP 보드 메모리로 읽어들인다.
- ④ 읽어들인 데이터를 QPSK 변조시켜 DAC로 내보낸다.
- ⑤ 채널을 통과하고 ADC를 거친 데이터를 얻는다.
- ⑥ ADC를 통과한 데이터를 QPSK 복조로 데이터를 복원한다.
- ⑦ ③번으로 가서 반복한다.

#### ◆ 다음은 호스트 프로그램에서의 동작 과정 <프로그램 및 데이터 Downloading 과정>

- ① DSP 보드를 Reset 시키고 Hold 시킨다.
- ② 타겟 프로그램을 DSP 보드의 프로그램 메모리에 로딩시킨다.
- ③ 데이터를 DSP 보드의 데이터 메모리에 로딩시킨다.(만일 전송할 데이터가 없는 경우 DSP를 동작시킬 수 있는 제어 신호를 타겟 쪽으로 보낸다.)
- ④ ③번으로 가서 반복한다.

#### <데이터 Uploading 과정>

- ① DSP를 Hold 시킨다.
- ② DSP 보드의 데이터 메모리로부터 데이터를 가져온다.

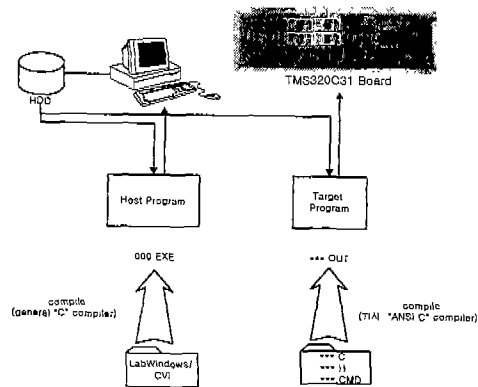


그림 5. DSP 보드에서 사용된 프로그램의 전반적인 구조  
Fig. 5. General structure of source program used in DSP board.

② PC에서 화면에 Display한다.(데이터를 메모리로부터 File로 저장도 가능하다.)

그림 5는 DSP 보드에서 사용된 프로그램의 전반적인 구조를 보여주고 있다.

### III. 시스템 구성

#### 3-1 제작한 DSP 보드의 구성

DSP 보드는 먼저 TI(Texas Instrument)사의 TMS 320C31 50 MHz의 DSP 프로세서를 사용하였고, ADC와 DAC는 Maxim 사의 max538과 max170으로 serial type의 변환기를 사용하였고, 또한 프로그램과 데이터의 저장용의 메모리는 Samsung사의 K6 R1016C1C SRAM을 사용하였다. 그리고 ISA interface, 제어신호, 그리고 데이터의 임시 기억장소로 latch등이 사용되었다.

DSP보드의 메모리는 프로그램 메모리 번지가 00000000H-0000FFFFH까지며, 데이터 메모리는 00010000H-0001FFFFH이다. PC에서 코딩된 타겟 프로그램은 인터페이스보드를 통하여 00000000H - 0000FFFFH 번지로 로딩되어 DSP에 의해서 실행된다. DSP에서 처리된 결과는 00010000H- 0001FFFFH 번지 사이에 저장하여 PC에서 DSP의 처리된 결과를 업로딩하여 파일이나 모니터에 결과를 나타낸다.

ADC와 DAC는 DSP의 직렬포트에 인터페이스되어 있다. 사용된 ADC와 DAC가 12비트이므로 DSP가 AD변환된 데이터를 수신할 때에는 DSP의 수신 레지스터(080804CH)가 32비트이므로 앞의 20비트는 마스크를 하여 DSP에서 사용하지 않는다. 마찬가지로 DSP에서 처리한 결과를 D/A를 통하여 송신할 때에도 DSP의 송신레지스터(0808048H)는 32비트이므로 앞의 20비트는 마스크 처리하여 사용하지 않는다.

DSP를 홀드하거나 리셋할 때에는 PC의 I/O번지에 DSP를 홀드 또는 리셋하는 데이터를 출력하면 DSP에서는 홀드 또는 리셋된다. 이 때 PC에서는 타겟 프로그램을 인터페이스 보드를 통하여 로딩하거나 DSP가 처리한 결과를 PC상의 메모리나 파일로 업로딩할 수 있다.

처리된 결과를 DAC을 통해 송신하거나 수신 신호를

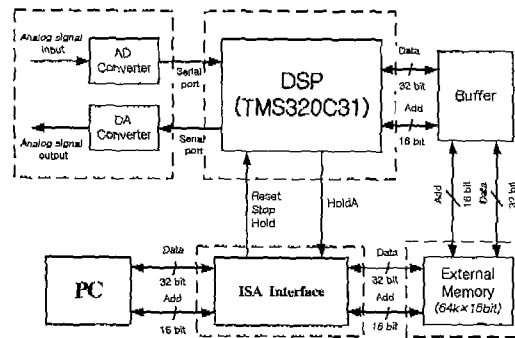


그림 6. 시스템 블록도

Fig. 6. System block diagram.

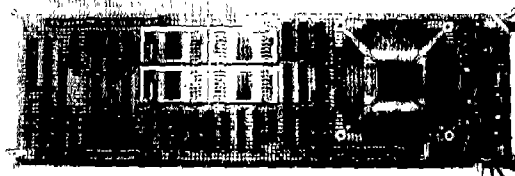


그림 7. 실제 제작한 DSP 보드

Fig. 7. Implemented DSP board.

호를 ADC로 변환하는 샘플링 주기는 타이머를 이용했으며, DSP의 Timer Period 레지스터(808028H)에 쓰여진 값이 0이 될 때마다 주기적으로 DSP에 타이머 인터럽트를 발생시키며, 이 때에 DSP는 인

표 1. 주요 부품 내용 및 특징

Table 1. Main components and characteristics.

List	Company	Part Number	Characteristic
DSP Processor	Texas Instrument	TMS320C31	50 MHz (clock)
AD Converter	Maxim	MAX538	serial type 12 bit (resolution) 125 kHz (sampling)
DA Converter	Maxim	MAX170	serial type 12 bit (resolution)
External Memory	Samsung	K6R1016C1C	64k x 16bit, 12 ns

터럽트 루틴 내에서 DSP에서 처리한 결과를 송신하기 위해 직렬포트의 송신레지스터에 처리된 결과를 출력한다. 그리고 수신신호가 AD변환되어 그 결과가 수신 레지스터에 들어 있으므로 이 레지스터의 값을 읽어서 처리를 한다. 송·수신 프로그램에서 소요된 클럭의 수는 데이터를 입·출력하는 인터럽트 루틴과 메인 프로그램(인터플레이션, 데시메이션과 필터링, 그리고 변·복조)을 포함해서 약 70 클럭이 소요되어 시간으로는  $70 \times 40 \text{ ns} = 2.8 \mu\text{sec}$ 이다. 그림 6은 시스템의 블록도이고, 그림 7은 실제로 제작한 DSP 보드의 사진이다.

#### IV. 실험 및 결과

##### 4-1 송·수신단의 시뮬레이션 환경

실험은 PC상에서 먼저 수행한 결과를 기준으로 DSP 프로세서에서 작동할 QPSK 변·복조 부분, A/D 변환기, 그리고 D/A 변환기의 정상동작을 확인하였으며 마지막으로 잡음을 통과하는 실험을 했다. 잡음은 Hewlett Packard(HP)사의 33120A Function Generator 에서 발생시켰으며, 잡음의 크기를 증가시키면서 SNR에 따른 복조신호의 에러발생률을 알아 보았다. 그리고 A/D 변환기의 샘플링 주파수는 40 kHz 이며 12 bit 직렬형 A/D 변환기를 이용했으며, D/A 변환기도 마찬가지로 직렬형이다. A/D와 D/A 변환기의 전·후에 있는 LPF의 차단주파수는 20 kHz이다.

PC상에서 시뮬레이션은 C언어를 사용하였고, DSP 보드에서는 어셈블리 언어를 사용하여 구현하였다. 여기서 사용된 필터는 필터 길이가 17인 제곱근 상승코사인 필터이며, 이는 심볼당 4개의 표본을 취하였기 때문에 4개의 심볼 구간에 해당된다. 그리고 초과대역 요소(roll-off factor)  $\alpha$ 는 0.2 이다. 먼저 송신단을 보면 PC상에서 프로그램으로 랜덤한 Data I 채널과 Q 채널의 신호를 발생시켰다. 그리고 발생된 신호를 PC에서 floating 형식으로 DSP보드로 다운로드한 후 이를 이용해서 DSP보드에서 인터플레이션(M=4)과 제곱근 상승코사인 필터를 통과한 다음 변조를 시켰다. 신호를 변조한 다음 D/A 변환기와 LPF를 통과한 후 전송된다. 전송된 신호는 다시 LPF

를 통과한 후 A/D 변환기를 이용하여 디지털 신호로 변환하여 데이터 파일로 DSP보드의 데이터 메모리에 저장하였다. DSP보드의 데이터 메모리에 저장한 수신된 신호를 PC에서 업로딩하여 처리하였다. 수신단에서 사용한 적응 등화기의 탭의 길이는 송신단의 필터 길이와 동일한 17을 사용하였다.

##### 4-2 시뮬레이션 결과

###### 4-2-1 PC상에서 C로 수행한 결과

DSP 보드에서 수행을 하기 전에 PC상에서 C언어를 이용하여 변·복조 과정을 잡음과 채널의 효과를 추가하지 않고 수행하였다. 먼저 PC상에서 랜덤하게 발생한 I, Q 채널의 신호를 변조하여 송신단의 출력 데이터를 수신단의 입력데이터로 사용하여 복조를 하였다.

###### 4-2-2 DSP보드를 이용하여 수행한 결과

DSP 보드의 정상 동작을 확인하기 위해서 그림 8과 같이 DSP에서 인터플레이션과 필터를 통과하고 변조하여 D/A 변환기로 출력하지 않고 직접 수행하였다. PC에서 발생시킨 데이터를 DSP 프로세서를 이용하여 수행하였다. 그림 9에 나타난 수신단의 입력신호가 한 샘플정도 지연되었지만 송신단의 신호가 그대로 복원되었다.

###### 4-2-3 D/A와 A/D 변환기를 통과하여 수행한 결과

앞의 두 실험에서 DSP의 정상동작을 확인하였으

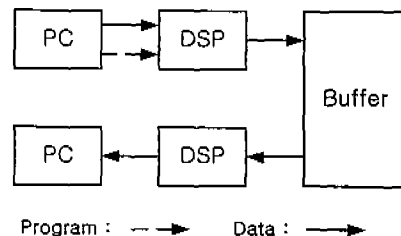
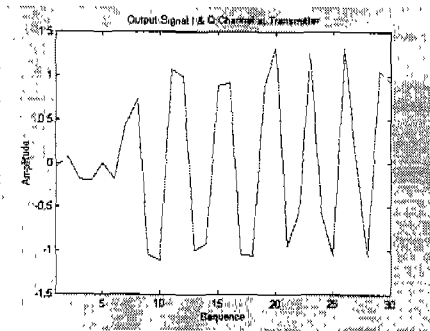
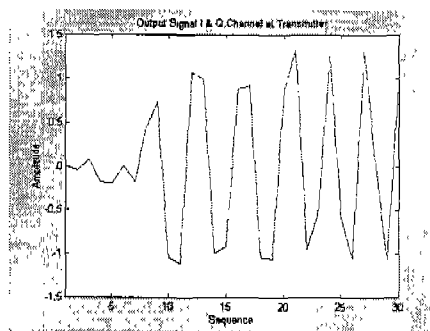


그림 8. DSP 동작 확인 구성도

Fig. 8. Block diagram for validation of DSP operation.



(a) PC상에서 수행할 때  
(a) On PC



(b) On PC  
(b) On DSP board

그림 9. 수신단에서의 입력 신호 비교

Fig. 9. Comparison of input signal in receiver.

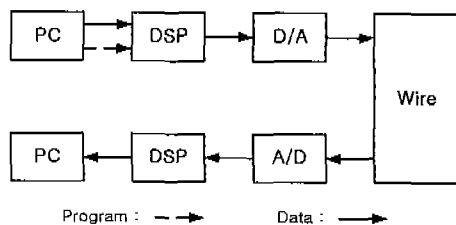


그림 10. D/A와 A/D 변환기 동작확인 구성도

Fig. 10. Block diagram to confirm D/A & A/D converter operation.

며 D/A와 A/D 변환기의 정상동작을 확인하기 위해 그림 10과 같이 수행하였다. DSP에서 인터플레이션, 필터링, 그리고 변조를 하여 변조된 신호를 D/A 변환기를 통하여 송신하고, 송신된 신호는 채널(선으로 직접 연결함)을 통해서 A/D 변환기로 입력되며 입력된 신호를 변환하여 메모리에 저장한 다음 복조하였다. LMS 알고리즘 수렴상수  $\mu$ 의 값을 0, 0.01, 0.0001로 변화시키면서 수행하였다.

그림 11과 그림 12에 나타나듯이 이번 실험의 결과를 보면 등화기를 동작시킬 때( $\mu = 0.0001$ )가 등화기를 동작시키지 않을 때( $\mu = 0$ )보다 눈 모양이 더욱 뚜렷하게 잘 나왔다. 수신단에서 복원된 신호 I, Q 채널의 신호를 가지고 필터의 수렴상수( $\mu$ )가 0.0001일 때의 눈모양을 보면 수렴상수가 0일 때보다 고주파 성분을 많이 통과시키고 있다. 이러한 이유는 결정된 값이  $\pm 1$ 에 가까운 값을 갖도록 필터 계수를

조정하므로 필터 출력값이 결정순간에 1에 가까운 값이 되기 때문이다.

본 논문에 소개된 눈모양은 수신단에서 등화기를 통과한 I, Q 채널의 신호를 그린 것이다.

#### 4-2-4 잡음이 있을 때의 실험 결과

앞에서 수행하였던 실험을 통하여 DSP, D/A, A/D 변환기의 정상동작을 확인하였으며 이번에는 잡음에 의한 효과를 알아보려고 한다. 본 논문에서는 Hewlett Packard(HP)사의 33120A Function Generator로 발생시킨 백색잡음을 사용하였으며, 그림 13에 보는 것과 같이 D/A 변환기와 A/D 변환기 사이 채널에 잡음을 첨가하였다. 신호의 전력은 고정시키고 잡음의 전력을 변화시키면서 수신단에서의 에러 확률을 측정하였다. 수신단에서 복원시 적응 등화기의 수렴상수는 이전에 실험에서 결과가 좋았던 0.01과 0.001로 하여 측정하였으며, 메모리의 용량 때문에 데이터는 30,000개를 가지고 10번을 반복 시행하였다. 즉 I, Q 채널에 각각 150,000개의 데이터를 수신하여 에러 확률을 측정하였다. 그림 14는 측정된 데이터로 얻어진  $E_b/N_0$ 에 따른 에러 확률을 나타내고 있다. 실험 결과 수렴상수  $\mu$ 가 0.001일 때 0.01보다 같은 품질( $E_b/N_0$ )에서 에러 확률이 적음을 알 수 있었다. 그러므로 수신단에서 신호 복구시 에러를 줄이려면 적응 등화기의 수렴 상수값을 수렴 조건 한계를 벗어나지 않는 범위 내에서 실험에 의하여 시스템에

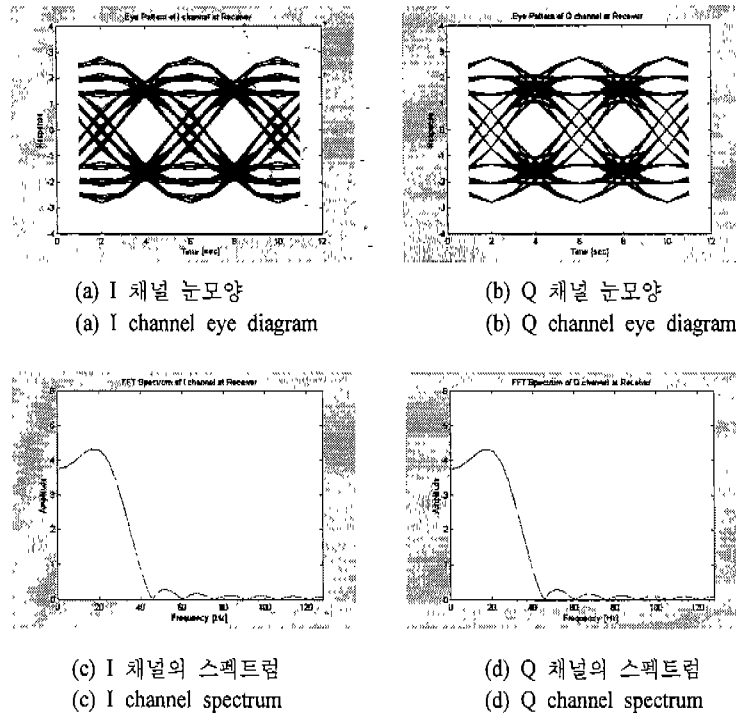


그림 11. D/A와 A/D 변환기를 통과한 신호의 눈모양과 스펙트럼(적응 등화기 사용 안한 경우)  
Fig. 11. Eye diagram and spectrum of signal passed D/A & A/D converter (Without adaptive equalizer)

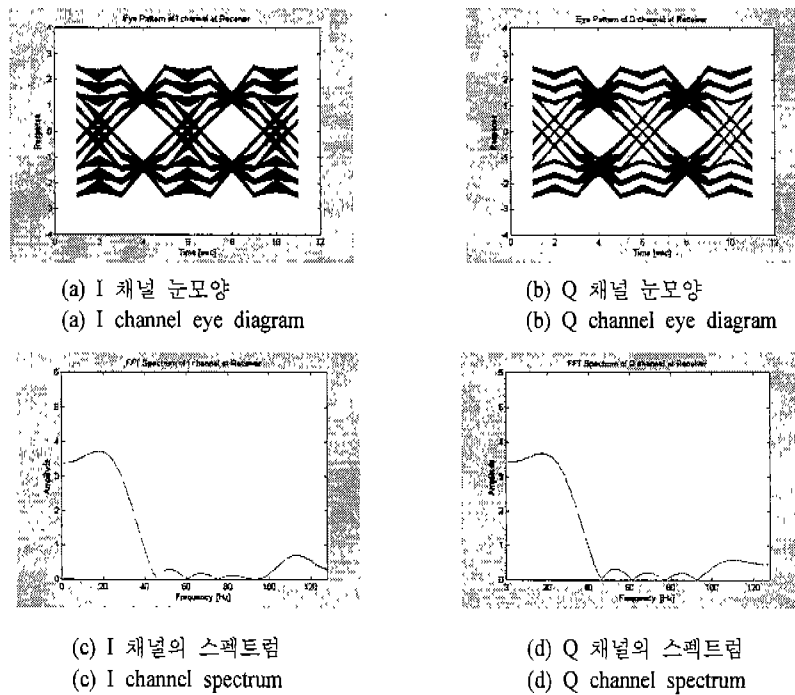


그림 12. D/A와 A/D 변환기를 통과한 신호의 눈모양과 스펙트럼( $\mu=0.0001$ )  
Fig. 12. Eye diagram and spectrum of signal passed D/A & A/D converter ( $\mu=0.0001$ ).



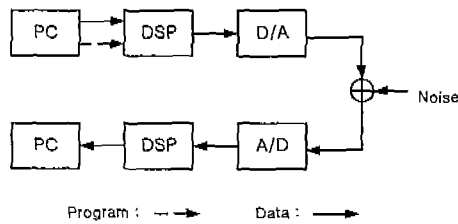


그림 13. 잡음 효과를 위한 구성도  
 Fig. 13. Structure for noise dithering.

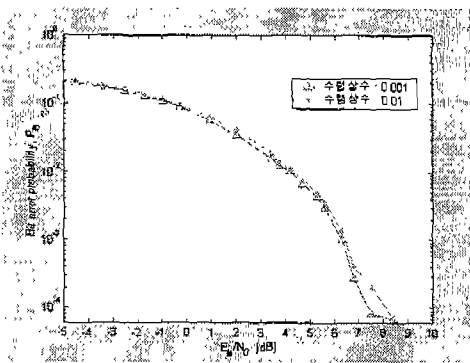


그림 14. 비트 오류 성능  
 Fig. 14. Probability of bit error performance.

맞게 설정하여야 한다.

### V. 결 론

본 논문에서는 TI 사의 TMS320C31 DSP 프로세서를 이용하여 QPSK 모뎀을 구현하였다. DSP에서 수행되는 프로그램은 어셈블리 언어를 이용하여 PC 상에서 코딩하고 컴파일하여 ISA 인터페이스를 통하여 DSP 보드의 프로그램 메모리로 로딩하고 임펄스 응답이나 DSP에서 처리하는데 필요한 데이터나 처리되어진 데이터는 DSP 보드의 데이터 메모리로 로딩할 수 있도록 구성하였다. 이 때 DSP와 PC간의 통신은 PC의 I/O 번지를 통하여 이루어지며, DSP의 Reset와 Hold 단자를 이용하여 DSP의 상태를 확인할 수 있었다.

실험 결과 PC상에서 C언어로 시뮬레이션한 결과와 DSP에서 인터플레이션과 필터를 통과하고 변조하여 DAC로 출력, ADC로 입력하여 디지털로 변환된 신호를 처리한 결과가 같음을 확인하여 DSP가

모뎀으로 정상적으로 구현되었음을 확인하였다. 또한 구현된 QPSK 모뎀의 수신단에서 약 70클럭이 소요되어 시간으로는  $70 \times 40 \text{ ns} = \mu\text{sec}$ 이며, 메모리는 코드 부분, 데이터 부분 각각 60K 워드를 사용하였다. 그리고 수신단의 등화기의 수렴 상수값에 따른 실험 결과로는 등화기를 동작시킬 때가 등화기를 동작시키지 않을 때보다 눈 모양이나 수신 데이터의 복원능력이 우수함을 확인했고, 같은 품질 ( $E_b/N_0$ )에 대해 수렴 상수값의 적절한 조절로 상대적으로 에러가 적은 수신 데이터를 복원할 수 있었다. 그러므로, 시스템에 맞는 적절한 등화기의 수렴 상수값이 실험을 통하여 정해져야 할 것이다.

구현한 QPSK 모뎀을 방향탐지, 잡음제거나 음성 신호처리, 스펙트럼 분석 등의 실제 응용에 사용하기 위해서는 향후 실제 무선 채널의 영향에 대한 연구와 샘플링 주파수의 범위를 높이기 위하여 속도가 빠른 DSP 프로세서의 적용에 대한 연구가 필요할 것이다.

### 참 고 문 헌

- [1] B. Sklar, *Digital Communication Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [2] W. Y. Chen, G. H. Im and J. J. Werner, "Design of Digital Carrierless AM/PM Transceivers", AT&T and Bellcore, Aug. 19, 1992.
- [3] H. Johnson and M. Graham, *High-Speed Digital Design a Handbook of Black Magic*, Prentice-Hall, 1993.
- [4] A. Polydoros and K. Kim, "On the detection and classification of quadrature digital modulations in broad-band noise", *IEEE Trans, Commun*, vol. 38, Aug. 1990.
- [5] G. H. Im and J. J. Werner, "Bandwidth-Efficient Digital Transmission up to 155Mb/s over Unshielded Twisted Pair Wiring", *IEEE J. Selected Areas in Comm.*, vol. 13, pp. 1643-1655, Dec. 1995.
- [6] Texas Instruments, TMS320C3x User's Manual. 1994.

- [7] Texas Instruments, TMS320C3x General- Purpose Applications, 1998.
- [8] Texas Instruments, TMS320 Floating-Point DSP Code Generation Tools, Release 4.60, 1995.
- [9] Simon S. Haykin, *Digital Communications*, Wiley, 1998.

- [10] 박귀태, 이상락, "C 언어로 쉽게 쓰는 TMS 320C31", 대영사, 1998.
- [11] 박상용, "PBC 방식을 이용한 PSK 신호의 자동 식별 시뮬레이터 구현", 경희대학교 석사학위 논문, 2000.

**김 광 호**



1999년 2월: 순천대학교 전자공학과 (공학사)  
 2001년 2월 : 경희대학교 전자공학과 (공학석사)  
 2001년 3월 ~ 현재: (주)한국컴퓨터 기술 연구소 연구원  
 [주 관심분야] 통신신호처리, DSP,

적응 필터 등

**김 중 욱**



2000년 2월: 경희대학교 전자공학과 (공학사)  
 2000년 3월~현재: 경희대학교 전자공학과 석사과정  
 [주 관심분야] 통신신호처리, DSP, SDR 등

**김 영 수**



1981년 2월: 연세대학교 전자공학과 (공학사)  
 1983년 2월: 연세대학교대학원 전자공학과 (공학석사)  
 1988년 12월: Arizona State University 전기 및 컴퓨터공학과 (공학박사)

1985년 5월~1986년 5월: Consultant, Signal-System Technology Inc., U. S. A.  
 1986년 6월~1988년 12월: Research Associate, Arizona State University.  
 1989년 3월~1992년 8월: ETRI 전자기술부, 전자응용연구실, 실장.  
 1992년 9월~1996년 8월: 경희대학교 전자공학과 조교수.  
 1998년~현재: 한국전자과학회 스펙트럼공학 연구회 위원장.  
 1996년 9월~현재: 경희대학교 전자정보학부 부교수.  
 [주 관심분야] 어레이신호처리, 이동통신 시스템, 스펙트럼 추정, 적응필터, SDR 등

**조 병 모**

1982년 2월: 인하대학교 전자공학과 (공학사)  
 1985년 8월: 연세대학교 대학원 전자공학과 (공학석사)  
 1991년 2월: 연세대학교 대학원 전자공학과 (공학박사)  
 1991년 3월~현재: 순천대학교 전자공학과 부교수  
 [주 관심분야] 디지털 신호처리, Multirate 및 필터뱅크, 적응신호처리.