

## Design of a PC based Real-Time Software GPS Receiver

高先俊\* · 元鍾焄\* · 李資城\*\*  
(Sun-Jun Ko · Jong-Hoon Won · Ja-Sung Lee)

**Abstract** - This paper presents a design of a real-time software GPS receiver which runs on a PC. The software GPS receiver has advantages over conventional hardware based receivers in terms of flexibility and efficiency in application oriented system design and modification. In order to reduce the processing time of the software operations in the receiver, a shared memory structure is used with a dynamic data control, and the byte-type IF data is processed through an Open Multi-Processing technique in the mixer and integrator which requires the most computational load. A high speed data acquisition device is used to capture the incoming high-rate IF signals. The FFT-IFFT correlation technique is used for initial acquisition and FLL assisted PLL is used for carrier tracking. All software modules are operated in sequence and are synchronized with pre-defined time scheduling. The performance of the designed software GPS receiver is evaluated by running it in real-time using the real GPS signals.

**Key Words** : Software GPS Receiver, FFT-IFFT, FLL Assist PLL, Real-Time, SSE2, OpenMP

### 1. 서 론

오늘날의 일반적인 상용 GPS(Global Positioning System) 수신기의 신호처리 및 항법데이터 처리의 기능은 ASIC기반 칩셋(chipset)을 사용하여 설계된다. 이러한 ASIC기반 GPS 수신기는 설계의 변경이나 성능개선을 위하여 칩셋의 재설계가 반드시 요구된다. 소프트웨어기반 GPS 수신기는 RF(Radio Frequency)-Front-End를 제외한 모든 처리 과정을 범용 프로세서(processor)에서 구동되는 소프트웨어 모듈로 설계함으로써, 소프트웨어의 교체만으로 개선이 가능하며, 차세대 위성항법시스템으로의 변경이 용이한 장점을 지닌다. 반면, 모든 처리과정이 소프트웨어적인 연산에 의해 수행됨에 따라, 계산부하의 증가로 실시간 운영이 문제가 된다. 따라서 고속연산이 가능한 프로세서의 사용과 연산효율이 높은 신호 및 데이터처리 알고리즘의 개발과 프로세서기반의 코드최적화(code optimization) 설계과정이 요구된다.

본 논문에서는 범용 PC(Personal Computer)를 기반으로 하는 실시간 소프트웨어 GPS 수신기를 구현한다. RF-Front-End로부터 출력되는 중간주파수(Intermediate Frequency; IF) 신호를 데이터획득장치(Data Acquisition; DAQ)를 통해 입력받아 실시간으로 처리하며, 큐(Queue) 구조를 활용한 링버퍼(ring buffer)와 데이터 흐름제어(data flow control)를

통해 중간주파수 신호데이터의 원활한 처리가 가능하도록 구성한다. 입력된 중간주파수 신호데이터는 혼합기(mixer)와 적산기(integrator)의 연산과정에서 바이트형태(byte-type), 즉 8-bit 데이터로 변환하여 처리한다. 바이트형태의 데이터 변환은 표본화 정밀도(sampling resolution)를 증가시켜 신호감도를 개선하고, 추후 소프트웨어모듈의 변경이 필요 없는 직접변환(direct-conversion) 구조를 적용할 수 있게 하기 위함이다. Fast Fourier Transform(FFT)-Inverse FFT (IFFT)를 이용한 신호획득모듈과, Frequency Locked Loop (FLL)와 Phase Locked Loop(PLL)를 결합한 반송파 추적루프 및 Delay Locked Loop(DLL)를 이용한 코드추적모듈을 구성하며, 신호동기모듈, 의사거리 측정치 생성모듈, 그리고 항법필터모듈 등의 설계를 통해 실시간으로 측위정보를 제공하는 소프트웨어 GPS 수신기를 구현한다. 구현된 소프트웨어모듈은 millisecond(ms) 단위로 미리 정해진 시간계획(time scheduling)에 따라 순차적으로 실행되며, 프로세서 처리능력(processing power)의 80% 이내의 실시간 동작특성을 가질 수 있도록 설계한다. 후처리과정을 통해 파악된 각 소프트웨어모듈의 처리소요시간을 바탕으로, 고속처리용 소프트웨어 설계기법중 하나인 Streaming SIMD Extensions2 (SSE2)와 하이퍼쓰레드(Hyper-Thread™; HT)의 병렬처리 구조를 이용한 OpenMP(Multi-Processing) 기법을 활용하여 실시간처리가 가능하도록 각 소프트웨어모듈을 최적화한다 [1][2]. 실제 GPS 신호 환경에서의 측위실험을 통해 구현된 소프트웨어 GPS 수신기의 실시간 성능을 확인한다.

### 2. 소프트웨어기반 GPS 수신기 구조

GPS 수신기는 기능적으로 RF-Front-End부와 상관기

\* 교신저자, 正會員 : 亞洲大學 電子學科 博士課程  
E-mail : sjko@ajou.ac.kr

\* 正會員 : Research Associate in FAF · 博士

\*\* 正會員 : 亞洲大學 電子學科 正教授 · 工博

接受日字 : 2006年 4月 13日

最終完了 : 2006年 5月 12日

(correlator)부, 그리고 프로세서부로 구성된다. RF-Front-End부는 GPS 신호를 수신하는 하드웨어 모듈로서, 안테나로부터 수신한 GPS 위성신호를 하향변환(down-conversion)을 통해 처리가능한 수 MHz 주파수대역의 중간주파수 신호로 변환한다. 상관기에서는 중간주파수 신호를 이용하여 신호획득과 추적, 그리고 동기과정을 수행하며, 프로세서는 상관기 출력치와 항법데이터, 의사거리 측정치, 그리고 GPS 시각 등의 원시측정치를 이용하여 사용자의 위치를 추정한다. 소프트웨어 GPS 수신기에서는 RF-Front-End를 제외한 모든 처리과정을 소프트웨어모듈로 설계한다. 그림 1은 일반적인 소프트웨어 GPS 수신기의 구조를 나타낸다. 중간주파수 신호의 원활한 실시간 처리를 위하여 Dynamic Memory Access(DMA) 제어기와 공유메모리 버퍼(Shared Memory Buffer; SMB)를 활용한 데이터 흐름제어(data flow control)를 이용한다. 이러한 구조는 고속으로 입력된 중간주파수 신호를 효율적으로 처리하여 프로세서의 동작성을 높이고 신호데이터를 분산처리함으로써 프로세서의 계산 부하를 줄일 수 있는 이점을 제공한다. 입력된 중간주파수 신호는 적산시간(Pre-detection Integration Time; PIT) 단위의 데이터블록(data block)으로 분산되어 각 채널마다 독립적으로 동작하는 신호처리모듈을 거치게 되며, 추출된 원시측정치를 이용하여 항법해를 추정하게 된다. 처리과정상에서의 중간주파수 신호는 외부저장장치(external storage device)에 저장하여 후처리 과정을 통한 GPS 신호분석과 수신기의 오동작요인 분석 등에 이용한다.

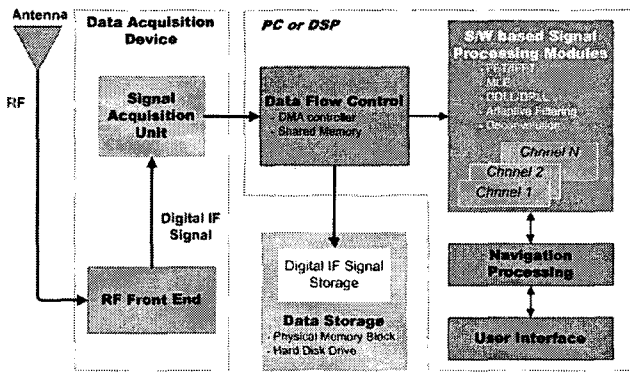


그림 1. 소프트웨어기반 GPS 수신기 구조  
Fig. 1. The Structure of Software based GPS Receiver

### 3. 실시간 소프트웨어 GPS 수신기 설계 및 구현

#### 3.1 RF-Front-End와 중간주파수 신호데이터 제어모듈 설계

RF-Front-End부에서는 약 1.5GHz의 GPS 위성신호를 처리가능한 중간주파수 신호로 하향변환하며, 일반적으로 4MHz 정도의 중심주파수(center frequency)와 2MHz의 신호대역폭(null-to-null bandwidth)을 가지도록 설계된다. 위성신호의 왜곡 없는 변환과 신호잡음(signal noise)의 최소화를 위해 효율적인 주파수계획(frequency plan)과 적절한 필터설계가 요구된다. 또한 기저대역(baseband)과 근접한

중간주파수 신호를 얻기 위하여 적합한 표본화주파수(sampling frequency)의 선택이 필요하다.

본 논문에서는 Zarlink사의 GP2015를 활용하여 RF-Front-End를 설계하였다. GP2015는 3단계 하향변환 구조와 2-bit A/D 변환기를 이용하여 1.405MHz의 디지털 중간주파수 신호를 출력한다. 출력신호는 부호와 절대치를 의미하는 SIGN과 MAG 신호로 분리되어 출력된다[3]. 그림 2는 GP2015에서 수행되는 하향변환과정을 나타낸다.

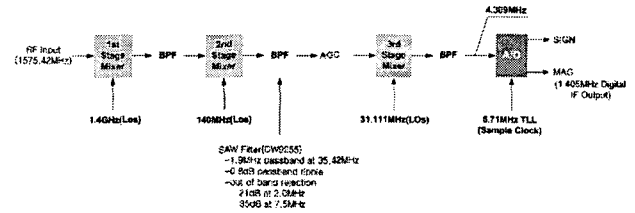


그림 2. GP2015기반 하향변환과정  
Fig. 2. GP2015 based Down-Conversion Scheme

출력된 중간주파수 신호는 소프트웨어모듈에서의 신호처리를 위하여 프로세서의 기본처리단위인 바이트형태로 변환되어야 하며, 2가지 정도의 변환형태가 제시되고 있다. 하나는 SIGN과 MAG신호를 나누어 비트(bit)단위로 구성하는 방식으로서, 1 바이트 내에 8개의 표본신호를 포함하는 이점이 있으나, 비트단위의 국부신호 생성을 위해 신호데이터의 비트 재구성과 복잡한 비트연산이 수반되므로, 연산 효율성이 떨어지는 문제점이 발생된다. 이를 해결하고자 도플러주파수(Doppler frequency)와 C/A 코드지연(code delay)을 일정한 정밀도 간격으로 적용한 국부신호를 미리 생성하여 활용하는 기법이 제시된 바 있으나, 신호추적 정밀도의 한계에 기인한 바이어스(bias) 오차가 발생할 수 있으므로, 측위정확도에 영향을 미칠 수 있다. 바이트단위 중간주파수 신호는 SIGN과 MAG 신호의 조합을 이용하여 변환된  $\pm 3, \pm 1$ 의 물리적 신호값을 바이트단위 데이터로 구성하는 방식으로, 1바이트에 1개의 표본신호로 구성된다. 프로세서의 기본처리단위인 바이트로 직접 처리함으로써, 국부신호 생성시, 데이터 재구성 등의 부수적 과정이 요구되지 않으며, 높은 정밀도와 신호감도를 유지한 상태에서의 신호처리가 가능하다[4]. 반면, 데이터 크기의 증가에 따른 계산부하의 증가를 보완하기 위해 특정 프로세서상에서의 최적화된 소프트웨어 코드설계가 요구된다.

본 논문에서는 바이트형태의 중간주파수 신호를 이용한다. 데이터획득장치로부터 입력된 비트단위의 SIGN과 MAG 신호는  $\pm 3, \pm 1$ 의 바이트형태 값으로 변환되어 공유메모리에 저장된다. 큐구조를 활용한 공유메모리는 512KB 블록으로 구성되며, 순차적으로 입력된 중간주파수 신호의 버퍼링 역할을 수행한다. 각 데이터블록은 연속으로 입력되는 중간주파수 신호의 저장시, 프로세서의 처리부하를 줄이기 위하여 DMA 제어를 통해 제어되며, 신호처리모듈과의 이벤트(event) 신호 교환을 통해 연속된 중간주파수 신호가 전달, 처리되도록 동작한다. PC 내부로 신호데이터를 입력받기 위하여 데이터획득장치를 활용한다. 이용된 데이터획득장치는 PCI 타입의 PC 카드 형태를 취하며, 1개의 외부클록

(external clock) 입력채널과 2개의 데이터 입력채널로 설계되어 있다. 수신기의 표본화주파수와 동기된 SIGN과 MAG 신호를 획득하기 위해 표본화주파수를 외부클럭 입력채널과 연결하며, 입력된 SIGN과 MAG 신호는 2개의 데이터 입력채널에 분리, 할당되어 처리된다. 이와 같은 일련의 데이터 획득과정을 위하여 데이터획득장치 제어용 소프트웨어모듈을 설계하였으며, 그림 3은 구현된 데이터획득장치, RF-Front-End, 그리고 설계된 데이터 저장 프로그램의 동작화면을 나타낸다.

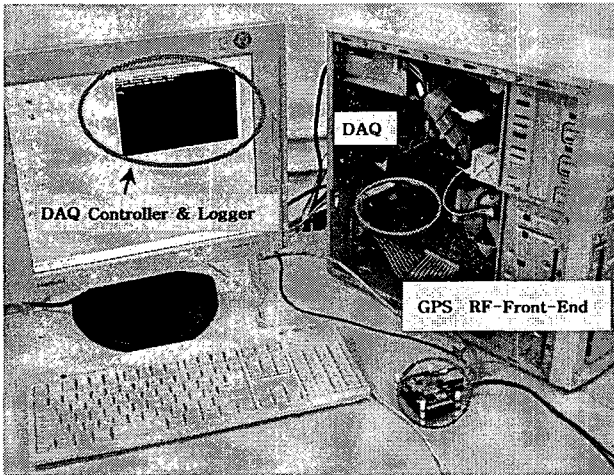


그림 3. 데이터획득장치, RF-Front-End, 그리고 데이터 획득장치용 제어프로그램 구성  
 Fig. 3. The Structure of DAQ, RF-Front-End, and DAQ Controller

3.2 FFT-IFFT기반 신호획득 소프트웨어모듈 설계

신호획득(signal acquisition)은 도플러주파수와 코드지연에 대한 2차원적 검색을 통해 가시위성군의 신호를 검출하는 과정이다. 본 논문에서는 고속의 신호획득 성능을 얻기 위하여 FFT-IFFT기반의 순환상관함수(circular correlation function)를 소프트웨어모듈로 설계하였다[5]. FFT-IFFT 기법은 고속의 초기위치획득시간(Time-To-First-Fix; TTFF) 성능을 위한 기반기술로 이용된다. 입력된 중간주파수 신호와 도플러주파수 및 코드지연을 고려한 국부생성신호는 각각 다음과 같이 표현된다.

$$r_{IF}(t_k) = A d(t_k - \tau) c(t_k - \tau) \cos [2\pi (f_{IF} - f_d)t_k - \phi(t_k)] + \nu_k \quad (1)$$

$$r_c(t_k) = c(t_k - \hat{\tau}) e^{2\pi (f_P - \hat{f}_d)t_k} \quad (2)$$

여기서,

$r_{IF}(t_k)$  : 중간주파수 신호

$r_c(t_k)$  : 도플러주파수와 코드지연을 고려한 국부생성신호

$A$  : 신호전력

$d(t_k - \tau)$  : 항법 데이터

$c(t_k - \tau)$  : C/A 코드

$\tau$  : 코드위상 지연

$f_{IF}$  : 중간주파수신호의 공칭주파수 (nominal frequency)

$f_d$  : 도플러 주파수

$\phi(t_k)$  : 초기 반송파위상

$\nu_k$  : 신호잡음

$t_k$  : 표본화시각

입력신호와 생성신호간의 시간영역(time-domain) 상관함수는 다음과 같이 정의된다.

$$Corr(\hat{\tau}, \hat{f}_d) = \sum_{k=0}^{N-1} r_c(k)r_{IF}(n+k) \quad (3)$$

여기서,  $k$ 는 표본화 시각지수(time index)를 나타내며,  $N$ 은 전체 표본데이터의 수를 의미한다. 순환상관함수는 시간영역 상관함수의 주파수변환, 즉 FFT로 정의할 수 있으며, 이는 주파수영역에서의 순환코드를 이용한 상관함수를 의미한다. 최대상관치(correlation peak)는 식 (3)의 절대치로 판별되며, 다음과 같이 표현된다.

$$|Corr(\hat{\tau}, \hat{f}_d)| = |IFFT\{R_c(f) \cdot R_{IF}^*(f)\}| = |IFFT\{R_{IF}(f) \cdot R_c^*(f)\}| \quad (4)$$

식 (4)를 이용하여 최대상관치를 검출함으로써, 대략적인 도플러주파수와 코드지연치를 얻을 수 있다. 이러한 FFT-IFFT기반 신호획득기법은 1ms 중간주파수 신호만으로 모든 가시위성의 신호검출이 가능함으로써, 신호검색 소요시간의 단축을 통해 고속의 초기위치획득시간 성능을 제공할 수 있다. 본 논문에서는 실시간으로 FFT-IFFT 처리를 수행하기 위해 도플러주파수를 적용한 국부생성신호의 FFT를 미리 연산 및 저장하여 이용함으로써, 중간주파수 신호에 대한 FFT 및 IFFT의 1회 계산만으로 신호검출이 완료될 수 있도록 구성하였다. 그림 4는 구현된 FFT-IFFT기반 신호획득 알고리즘의 구조를 나타낸다[6].

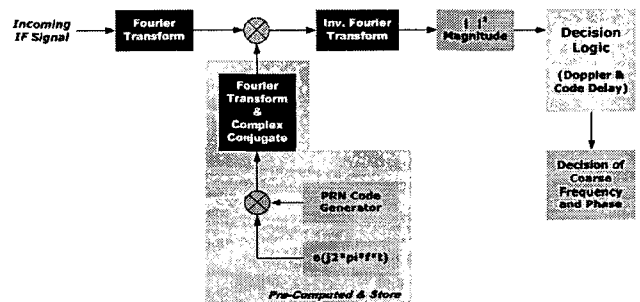


그림 4. FFT-IFFT 기반 신호획득 알고리즘  
 Fig. 4. FFT-IFFT based Signal Acquisition Algorithm

3.3 신호추적 소프트웨어모듈 설계

본 논문에서는 반송파추적(carrier tracking)을 위한 FLL-PLL 결합 추적루프와 코드추적(code tracking)을 위한 1차

DLL 추적루프를 설계하였다. 각 추적루프에 적합한 루프필터(loop filter)를 구성하며, 고전제어기 설계기법에 따른 Jaffe-Rechitin 필터 구조를 활용하여 설계하였다[7].

### 3.3.1 코드추적용 1차 DLL 루프필터

정밀한 코드지연을 추적하기 위해 설계된 1차 DLL 루프필터는 다음과 같다.

$$\tau_{k+1} = \tau_k + 4BT_e_c + (r\hat{T}\ddot{\theta}) + TB_{code\ NCO} \quad (5)$$

여기서,

- $e_c$  : 코드 변별기(discriminator) 출력치
- $B$  : 잡음대역폭(noise bandwidth)
- $T$  : 적산시간
- $r$  : 환산계수(scale factor)
- $\hat{\theta}$  : 도플러주파수 추정치  
(estimate of Doppler frequency)

$B_{code\ NCO}$  : 코드 NCO 바이어스(code NCO bias)

식 (5)에서 코드 NCO 바이어스는 RF-Front-End에 포함된 발진기 성능규격으로 결정되며, 실험적으로 파악된 50Hz ( $\approx 50\mu s/s$ )로 설계하였다. 환산계수는 C/A 코드의 칩변화율(chipping rate)과 L1 방송파 주파수로부터 다음과 같이 정의된다.

$$r = \frac{R_c}{f_{L1}} = \frac{1}{1540} \quad (6)$$

### 3.3.2 반송파추적용 FLL-PLL 결합 신호추적모듈

본 논문에서는 수신기의 빠른 신호추적과 향상된 정밀도를 제공하기 위하여 FLL-PLL 결합 신호추적루프를 소프트웨어모듈로 구현하였다. 정상상태(steady-state)에서 신호추적오차의 완전한 제거를 위하여 2차 FLL 루프필터와 3차 PLL 루프필터를 통합설계하였다. 설계된 FLL-PLL 결합 추적루프의 구현식은 다음과 같다.

$$\theta_{k+1}^{\ddot{}} = \ddot{\theta}_k + \omega_{0,PLL}^3 T e_{\theta} + (\omega_{0,FLL}^2 T e_{\omega}) \quad (7.1)$$

$$\dot{\theta}_{k+1} = \dot{\theta}_k + T\ddot{\theta}_k + 1.1\omega_{0,PLL}^2 T e_{\theta} + (1.4\omega_{0,FLL} T e_{\omega}) \quad (7.2)$$

$$\theta_{k+1} = \theta_k + T\dot{\theta}_k + 2.4\omega_{0,PLL} T e_{\theta} + (T\hat{\theta}_{aiding}) + (TB_{carrier\ NCO}) \quad (7.3)$$

여기서,

- $e_{\theta}$  : PLL 변별기 출력치
- $e_{\omega}$  : FLL 변별기 출력치
- $\omega_{0,PLL}$  : PLL 대역폭
- $\omega_{0,FLL}$  : FLL 대역폭
- $\hat{\theta}_{aiding}$  : 외부속도 보정치
- $B_{carrier\ NCO}$  : 반송파 NCO 바이어스(carrier NCO bias)

일반적으로 추적필터는 추정치의 발산을 방지하기 위하여 과도상태(transient-state) 구간에서는 광대역(wide-band) 필터로 동작하고, 정상상태에서는 협대역(narrow-band) 필터로 동작하도록 설계된다. 본 논문에서는 과도상태에서의 PLL 대역폭을 22Hz로, 정상상태에서는 17Hz로 설정하였으

며, FLL 대역폭은 0.85Hz로, 적산시간은 1ms를 이용하였다. 설계된 각 대역폭은 일반적인 수신기의 경우보다 다소 큰 값을 나타내는데, 이는 충분한 대역폭 확보를 통해 높은 강인성을 지닌 추적필터를 구성함으로써, 수신기 시계오차 및 표분화오차 등에 기인한 신호추적실패의 방지와 중기동 (medium-dynamic) 특성상에서 동작이 가능한 수신기성능의 제공에 기인된다.

## 3.4 신호동기 소프트웨어모듈

### 3.4.1 비트동기

비트동기(Bit Synchronization)는 위성신호와의 비트데이터 동기과정을 의미하며, Carrier Lock Indicator(CLI)와 1ms 계수기(counter)를 이용하여 동기여부를 판단한다. CLI가 임계치이상을 유지하는 시점부터 1ms 계수기가 동작하며, 1ms 계수기 값이 20ms를 나타낼 때, 비트전이(bit-transition) 발생여부를 확인한다. 다음은 CLI의 계산식을 나타낸다[3].

$$CLI = \cos\phi \approx I_k I_{k-1} + Q_k Q_{k-1} \quad (8)$$

여기서,  $\phi$ 는 I-Q 벡터에 의한 위상을 의미하며,  $I_k$ 와  $Q_k$ 는 각각  $k$ 에서의 In-phase와 Quadrature-phase의 적산치를 나타낸다. 코사인(cosine) 함수의 특성에 의하여 비트전이가 발생될 때마다  $\pm\pi$ 의 위상변화가 발생되므로, 20ms마다 코드위상의 변화여부를 판단하여 비트전이를 확인한다. 식 (8)에서와 같이 CLI는 전상태와 현재상태의 I와 Q의 내적으로 표현됨에 따라, 음수가 되는 시점에서 비트반전이 발생됨을 확인할 수 있으며, 이를 검출하여 비트동기를 완성하게 된다.

### 3.4.2 프레임동기

프레임동기(frame synchronization)는 다음 4가지 조건의 검사를 통해 확인된다[8][9].

#### • 프레임동기 조건

- 1) TLM(TeleMetry) 워드(word)내부의 preamble(10001011 or 01110100) 검사
- 2) HOW(HandOver) 워드내부의 subframe ID(1-5번) 검사
- 3) HOW 워드내부의 zero비트 검사
- 4) 패리티(parity) 검사

위 4가지 조건검사를 통해 프레임의 동기여부를 판단하며, 프레임동기의 완료시점부터 HOW 내부의 Z-count를 이용한 GPS 위성시각이 정의될 수 있다. Z-count는 1.5초 정밀도를 가진 GPS 시각정보를 나타내며, 연속된 다음 서브프레임(subframe)의 시작지점과 동기되어 있다. 따라서 프레임동기 완료 후, 약 4.8초 이후부터 정확한 GPS 시각 추정과 의사거리 측정치의 산출이 가능하게 된다.

## 3.5 의사거리 측정치생성 소프트웨어모듈 설계

의사거리 측정치는 프레임동기를 통해 얻은 GPS 위성시각을 활용하여 수신기의 신호 수신시각(received time)과 위

성의 신호 발신시각(transmitted time)을 추정하고, 이를 차분한 전달지연시간(propagation delay time)과 광속(speed of light)의 승산(multiplication)을 통해 계산된다[8][9].

$$\rho(t) = c(t_r(t) - t_s(t - \tau)) \quad (9)$$

여기서,

$t_r(t)$  : 수신기의 신호 수신시각

$t_s(t - \tau)$ : 위성의 신호 발신시각

$c$  : 광속

GPS 위성의 신호 발신시각은 다음과 같이 계산된다 [8][9].

$$\begin{aligned} t_s(t - \tau) = & \text{Z-count} \\ & + \text{비트수} \\ & + \text{C/A 코드수} \\ & + \text{C/A 코드 칩(chip) 수} \\ & + \text{C/A 코드위상} \end{aligned} \quad (10)$$

초기상태에서 수신기의 신호수신시각은 신호추적중인 위성군의 발신시각에 평균적 전달지연시간인 0.076ms를 합산하여 대략적 수신시각을 설정한 후, 수신기 내부클럭의 증가분을 가산하여 얻게 된다[8]. 이때 신호수신시각이 GPS 시스템 시각과 ±0.5초 이내로 동기된 것으로 가정하여 전달지연시간을 보정하면 다음과 같이 표현된다.

표 1. 전달지연시간 보정

Table 1. Correction of Propagation Delay Time

보정식	조건	$t_{propagation} = \text{rem}(t_r, 1) - t_s(t - \tau)$
$t_{propagation} > 0.5$		$t_{propagation} = t_{propagation} - 1$
$t_{propagation} < -0.5$		$t_{propagation} = t_{propagation} + 1$

$t_{propagation}$  : 보정된 전달지연시간

$\text{rem}(\cdot, \cdot)$  : 나머지 연산자

### 3.6 실시간 소프트웨어 설계 및 구현

표 2. Intrinsics 함수를 사용한 동작코드의 예

Table 2. Example Codes using Intrinsics Functions

```

__m128 IntrinsicBuffer #1, IntrinsicBuffer #2, and IntrinsicBuffer #3 declaration
for i = zero to (the number of samples)/4 - 1
{
    // Accumulation
    IntrinsicBuffer #1 = _mm_setr_ps(DATA-a1, DATA-a2, DATA-a3, DATA-a4);
    IntrinsicBuffer #2 = _mm_setr_ps(DATA-b1, DATA-b2, DATA-b3, DATA-b4);
    IntrinsicAccum = _mm_mul_ps(IntrinsicBuffer #1, IntrinsicBuffer #2);
    ...
    total IntrinsicAccum = _mm_add_ps(total IntrinsicAccum, IntrinsicAccum);
}
    
```

소프트웨어 GPS 수신기의 실시간 성능을 확보하기 위해서는 알고리즘 및 소프트웨어 처리에 대한 효율적인 설계가 필요하다. 본 논문에서는 소프트웨어로 구현된 각 모듈의 계산 부하 및 처리속도를 분석하여 처리효율이 낮은 모듈에 대한 코드 개선을 수행하였다. 다음은 1초 분량의 중간주파수 신호 처리과정에서 측정된 각 소프트웨어모듈별 소요시간을 백분율로 비교한 그림이다. 그림에서와 같이, 중간주파수 신호와 국부신호를 처리하는 혼합기-적산기 모듈과 중간주파수 신호데이터 제어모듈에서 대부분의 처리시간이 소비됨을 볼 수 있다. 따라서 중간주파수 신호처리에 해당하는 소프트웨어모듈의 연산 효율성을 높이는 것이 실시간 동작을 결정하는 매우 중요한 기술적 요인으로 작용한다. 이와 같은 처리소요시간에 대한 분석결과를 바탕으로, 연산효율성을 높이기 위하여 최적화된 sin/cos 함수와 SSE2, 그리고 OpenMP 기법을 활용하였다[1][2]. 최적화된 삼각함수연산을 위하여 고정소수점기반인 연산 알고리즘의 적용과 어셈블러(assembly) 코드를 활용하여 설계하였다. SSE2 기법은 프로세서의 레지스터(register) 변수를 직접 이용함과 동시에 128-bit의 레지스터 변수를 4분할하여 재배치된 각각의 32-bit 데이터를 한번에 계산하는 기법으로서, 32-bit 데이터의 경우에는 4배의 연산속도를, 8-bit 데이터의 경우에는

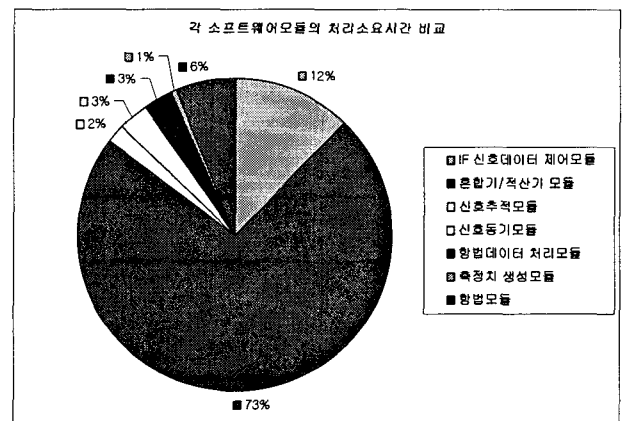


그림 5. 각 소프트웨어모듈별 처리소요시간 비교

Fig. 5. Comparison of The Processing Time with Each Software Modules

약 16배의 연산속도를 제공한다. 이러한 SSE2 기법을 혼합기-적산기 모듈에 적용하여 각 채널에서 중간주파수와 국부 신호간의 승산 및 적산을 효과적으로 처리하도록 설계하였다. SSE2를 적용할 때 크게 4가지의 소프트웨어 설계기법이 존재하며, 본 논문에서는 어셈블(assembly) 코드와 유사한 성능을 제공하는 Intrinsics를 사용하여 처리모듈을 설계하였다. 표 2는 혼합기-적산기 모듈내부에 실제 Intrinsics 함수로 작성된 동작코드의 예를 나타낸다[2].

중간주파수 신호데이터 제어모듈에서는 크게 4가지 기능이 수행되며, 데이터획득장치 제어부와 중간주파수 신호데이터의 버퍼링 및 큐잉(Queuing)부, 기본 처리단위인 3ms 데이터블록 생성부, 그리고 신호처리를 위한 2ms 데이터블록 생성부로 분리할 수 있다. 그림 6은 중간주파수 신호데이터의 버퍼링 및 큐잉부와 3ms 데이터블록 생성부를 나타낸다.

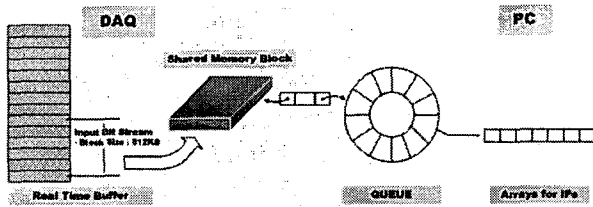


그림 6. 중간주파수 신호데이터 버퍼링 및 큐잉부와 3ms 데이터블록 생성부

Fig. 6. IF Data Buffering/Queuing and Generation of 3ms IF Data Blocks

그림에서 보듯이, 데이터획득장치로부터 입력된 중간주파수 신호데이터는 실시간 버퍼(buffer)에 임시 저장된 후, 512KB 데이터단위로 발생하는 이벤트에 의해 PC의 공유메모리버퍼로 전송된다. 이때 공유메모리버퍼는 이중버퍼의 큐로 설계되었으며, 기본 적산시간에 해당하는 1ms 데이터단위로 3ms 데이터블록 생성부에 전송한다. 전송된 1ms 데이터는 3ms 데이터블록내부에 순차적으로 입력되며, 모든 채널은 3ms 데이터블록을 공유데이터로 인식, 처리하게 된다. 각 채널은 공유된 3ms 데이터블록내에서 신호처리단위인 2ms 데이터블록을 재설정하고, 다시 코드지연만큼 이동한 지점으로부터 1ms 데이터를 추출하여 신호추적모듈로 입력한다. 이러한 구조는 각 채널마다 해당 위성에 대한 국부코드의 재생성 없이, 중간주파수 신호데이터의 이동으로 신호처리가 수행된다는 점과 각 채널 간 최대 1ms의 데이터블록의 차가 발생할 수 있다는 점에 기인한다[9]. 이러한 데이터 흐름제어모듈의 구동시, 2ms 데이터블록과 최종신호처리단위인 1ms 데이터블록을 재생성한다면, 불필요한 데이터이동과 메모리의 비효율적 활용 등이 발생되므로, 2ms과 1ms 데이터블록은 3ms 데이터블록내에서 상대적 위치를 나타내는 포인터를 이용하여 처리되도록 설계하였다. 이를 통해 불필요한 데이터 생성과정을 제거함으로써, 상당부분 개선된 처리속도와 계산부하의 감소를 얻을 수 있다. 또한 공유데이터인 3ms 데이터블록의 재설정만으로 적산시간의 유동적인 변경이 가능할 것으로 판단된다. 그림 7은 구현된 3ms 공유 데이터블록과 각 채널마다 정의된 2ms과 1ms 데이터블록에 대한 포인터의 관계를 나타낸다.

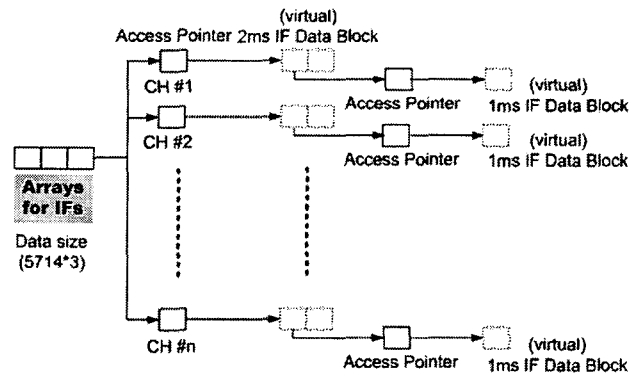


그림 7. 3ms 공유 데이터블록과 2ms/1ms 데이터블록 포인터의 관계

Fig. 7. The Relationship between 3m Shared Data Block and 2ms/1ms Data Block Pointer

이러한 구조로 설계된 소프트웨어 GPS 수신기는 WindowXP를 기반으로 동작하며, 개발언어로 Visual C++를 사용하였다. 일반적으로 Window 운영체제는 실시간운영체제(Real-Time Operating System; RTOS)와 달리 정확한 시각에서의 동작을 보장할 수 없으므로, 높은 우선순위의 확보와 이벤트의 고속처리를 통해서 실시간성을 확보해야 한다. 따라서 각 처리모듈의 독립적 구동과 우선순위개념을 적용하고자 대부분의 모듈을 쓰레드(thread)로 설계하였으며, 각 쓰레드간의 동기과 데이터통신에 따른 지연시간을 고려, 최적화 설계를 수행하였다. 이와 같은 설계기법을 활용함으로써, 실시간운영체제와 유사한 동작성능을 확보하였으며, 실시간 동작특성이 보장될 수 있도록 구성하였다. 다음 표는 본 논문에서 구현한 소프트웨어 GPS 수신기의 시스템 환경을 나타낸다.

표 3. 구현시스템 환경

Table 3. Environment of The Implemented System

항 목	구현시스템 환경
프로세서	Pentium4 3.0GHz with HT
운영체제 (Operating System)	WindowXP(32-bit version)
RAM	512MB
개발언어	Visual C++

### 3.7 성능실험 결과

본 논문을 통해 구현된 실시간 소프트웨어 GPS 수신기의 성능분석을 위하여 후처리 방식과 실시간 성능평가를 수행하였다. 신호처리 성능확인을 위해서는 각 채널당 처리된 많은 량의 데이터를 저장해야 하므로, 후처리방식을 통해 성능분석을 수행하였다. 약 10분동안 중간주파수 신호데이터를 저장하여 처리하였으며, 저장완료 후, 연속실험을 통해 실시간 동작성능을 확인하였다. 그림 8과 9는 구현된 소프

트웨어 GPS수신기를 활용한 11번 위성과 17번 위성의 신호 획득성능 결과를 나타낸다.

그림에서 보듯이, 도플러주파수와 코드지연치에 대한 2차원 검색공간내에서 각 위성신호의 검출결과를 확인할 수 있다. 11번 위성의 경우에는 3000Hz 도플러구간(doppler bin)에서, 17번 위성의 경우에는 -1000Hz 도플러구간에서 신호 검출이 이루어졌다. 초기 1ms 데이터만으로 모든 가시위성의 신호검출이 가능함을 확인하였으며, 실시간동작을 보장하기 위하여 검출된 가시위성만을 추적하도록 구성하였다. 그림 10과 12는 11번과 17번 위성에 대한 신호추적성능을 나타낸다. 그림의 위 부분으로부터 순차적으로 I-Q 신호전력, 코드전력, CLI, 코드지연치, 그리고 I-Q 위상자 다이어그램(phasor diagram)과 주파수 추적결과를 나타낸다. I-Q 신호전력 결과에서 보듯이, 약 1.5초 이후 신호추적이 완료되어 I 신호가 항법비트로 표현됨을 확인할 수 있다. CLI가 음수인 지점과 I 신호출력과의 비교를 통해 정확한 비트동기 및 검출이 성공적으로 수행되었음을 확인할 수 있다. 코드전력 결과로부터 Prompt(P) 코드의 신호전력이 Early(E)와 Late(L) 코드 신호전력보다 약 2배정도를 나타내므로, 성공적인 코드추적이 이루어졌음을 볼 수 있다. 또한 GPS 신호는 Binary Phase Shift Keying(BPSK) 변조방식을 이용함에 따라, I/Q plot에서는 신호추적 완료시점부터 I 신호축에 모든 신호정보가 집중됨을 볼 수 있다.

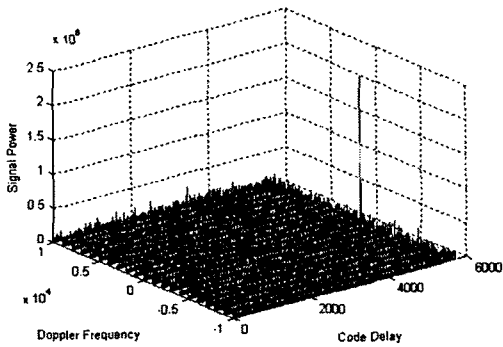


그림 8. 11번 위성에 대한 신호획득결과  
Fig. 8. Acquisition Result with SV 11

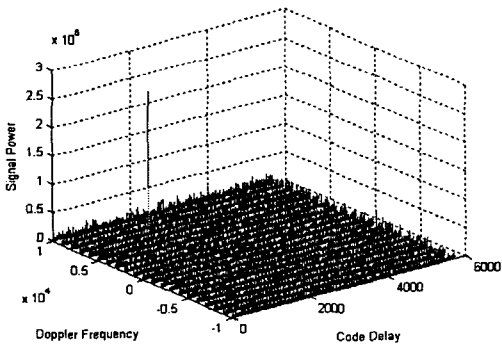


그림 9. 17번 위성에 대한 신호획득결과  
Fig. 9. Acquisition Result with SV 17

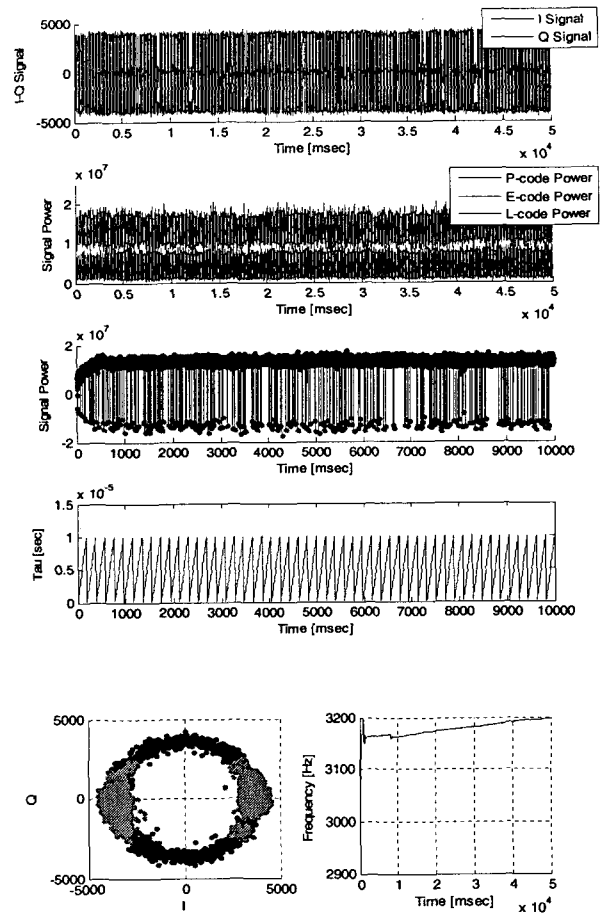


그림 10. 11번 위성에 대한 신호추적결과  
Fig. 10. Tracking Result with SV. 11

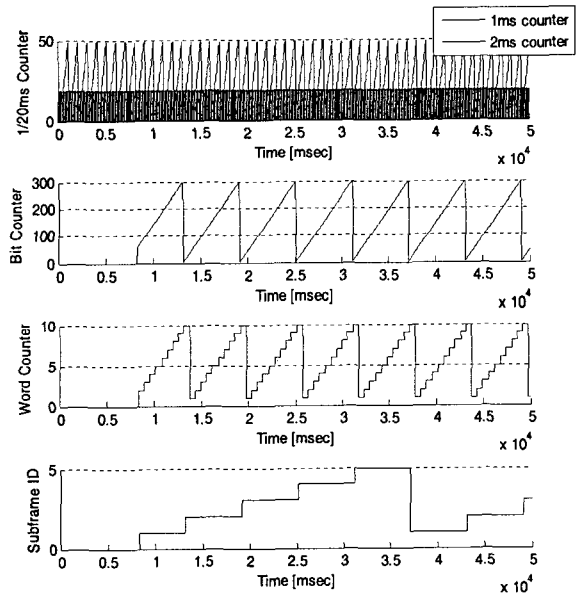


그림 11. 11번 위성에 대한 신호동기결과  
Fig. 11. Synchronization with SV. 11

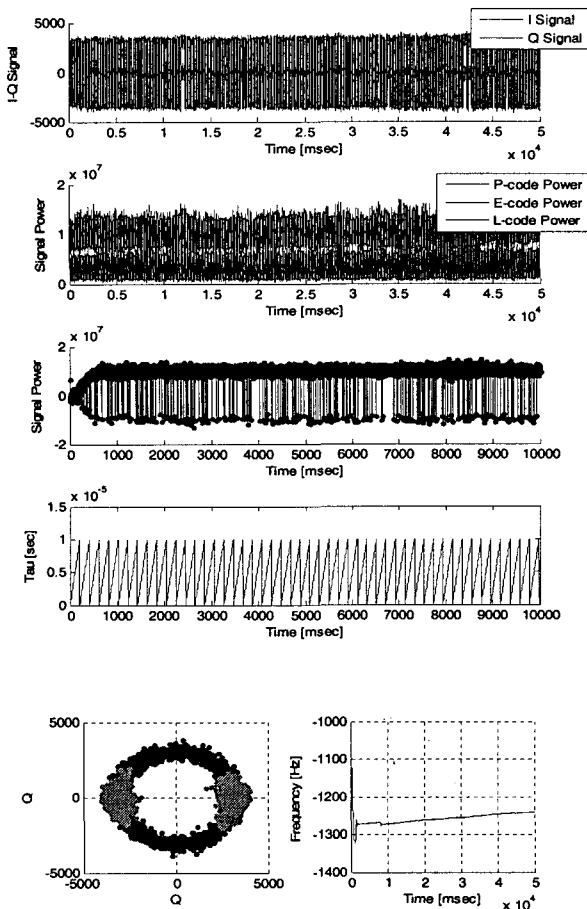


그림 12. 17번 위성에 대한 신호추적결과  
Fig. 12. Tracking Result with SV. 17

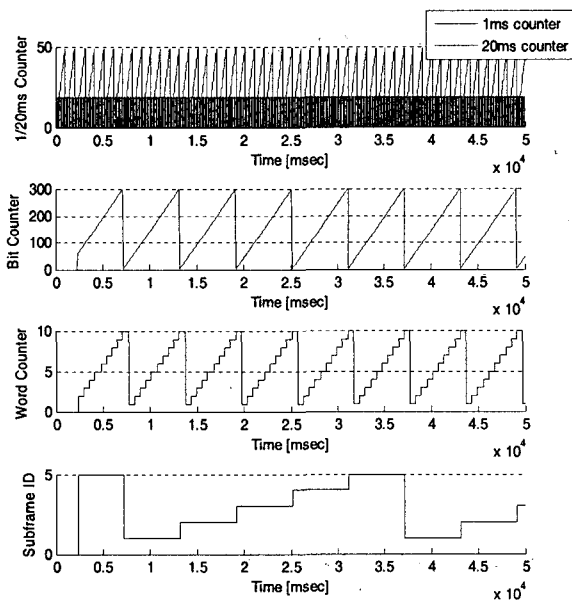


그림 13. 17번 위성에 대한 신호동기결과  
Fig. 13. Synchronization with SV. 17

그림 11과 13은 각 위성의 신호동기결과를 나타낸다. 11번 위성의 경우에는 약 2초 후, 17번 위성의 경우에는 약 8초 후부터 신호동기가 완료됨을 비트 계수기와 워드계수기를 통해 확인할 수 있으며, 동기된 시점부터 각 서브프레임의 구별을 통해 항법데이터의 추출과 GPS 시각추정이 진행되고 있음을 서브프레임결과로 확인할 수 있다. 1ms와 20ms 계수기 결과는 비트 및 워드 계수기와 동일한 결과를 보이므로, 각 위성신호와 일치된 시각 및 항법데이터 추출이 이루어지고 있음을 나타낸다.

이와 같이 성능확인된 신호처리 소프트웨어모듈과 항법모듈, 그리고 데이터 제어모듈 등을 통합한 소프트웨어 GPS 수신기의 측위성능을 확인하였다. 실험환경은 아주대학교

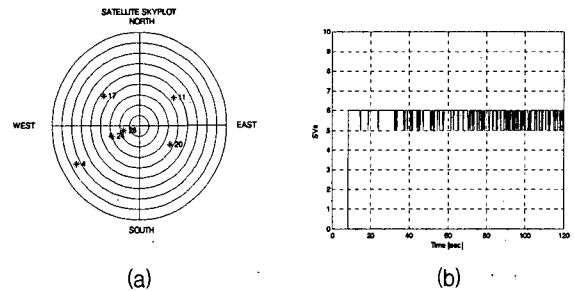


그림 14. 실험환경 : (a) 가시위성 배치도, (b) 추적 위성수  
Fig. 14. Experiment Environment :  
(a) Sky Plot, (b) The Number of Tracking Satellites

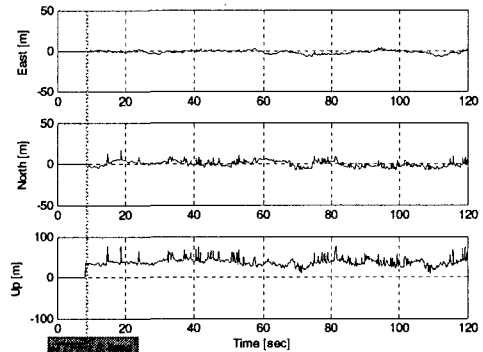


그림 15. ENU 좌표계의 각축에 대한 측위오차  
Fig. 15. Position Errors of Each Axis in ENU coordinates

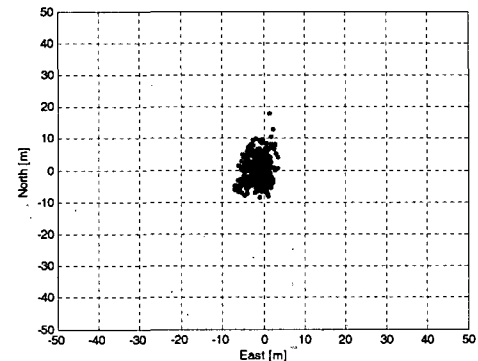


그림 16. East-North 평면 측위오차  
Fig. 16. Position Errors in East-North planes



공대건물 옥상에 설치된 GPS 안테나를 활용하여 GPS 신호를 수신, 처리하였으며, 그림 14에서와 같이 총 6개의 가시 위성을 추적, 이용하였다. 그림 15와 16은 실시간 측위결과를 나타낸다. 그림에서와 같이, 약 10m(2dRMS) 정도의 측위정밀도를 보이고 있으며, 초기위치획득시간 성능은 약 8.4초를 제공한다. 여기서, 8.4초의 초기위치획득시간 성능은 FFT-IFFT 신호획득모듈과 본 연구실에서 개발된 위성 궤도추산력 데이터도움기법(ephemeris data aiding technique)을 활용함에 기인한다. 측위결과에서 보듯이, 일반적인 독립(stand-alone) GPS 수신기의 측위정밀도와 동일한 성능을 나타내고 있다. ENU 좌표계에서의 수직축(Up axis)으로 다소 큰 오차를 보이고 있으나, 하드웨어기반 상용 GPS 수신기와 유사한 측위성능으로 판단되며, DGPS 적용시 효과적인 제거가 가능할 것으로 사료된다.

그림 17은 구현 수신기의 데이터처리 수행시간의 분석 결과이며, 1초 분량의 중간중과수 신호데이터를 처리하기 위해 요구되는 소프트웨어 GPS 수신기의 전체 동작시간을 의미한다. 그림에서 확인할 수 있듯이, 1초를 초과하는 처리시간은 존재하지 않으며, 평균적인 처리시간은 0.90085초가 되므로, 실시간으로 처리됨을 알 수 있다. 그림 18은 각 코드 최적화기법에 따른 처리시간을 비교한 그림이다. 그림에서 보듯이, 최적화를 적용하지 않은 경우, 약 16초의 처리시간을 보이는 반면, 공유메모리 데이터블록을 적용한 경우에는 약 6초를, OpenMP를 적용한 경우에는 4초 이내로 감소하며, 최종 SSE까지 적용한 경우에는 0.9초정도로 감소됨을 볼 수 있다. 이를 통하여 각 최적화기법의 적용에 따른 처리시간의 감소효과를 확인하였으며, 최종 실시간이 가능함을 확인하였다. 그림 19는 본 논문에서 구현한 소프트웨어 GPS 수신기의 동작화면을 나타내며, 약 80%의 CPU 사용정도에서 약 0.9초의 처리시간을 보이고 있음을 확인할 수 있다.

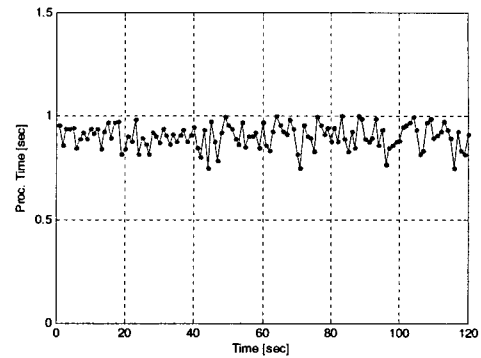


그림 17. 1초 데이터에 대한 처리시간 결과  
Fig. 17. Processing Time with 1 sec Data

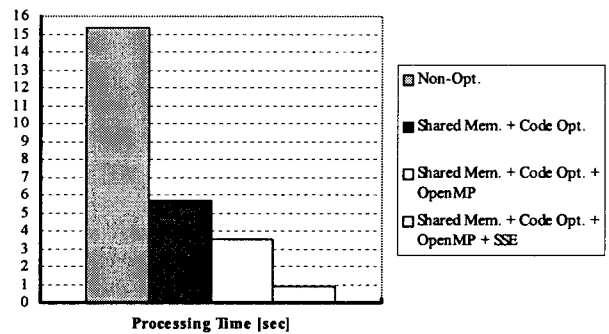


그림 18. 코드최적화에 따른 처리시간 비교  
Fig. 18. Comparison of The Processing Time by Code Optimization

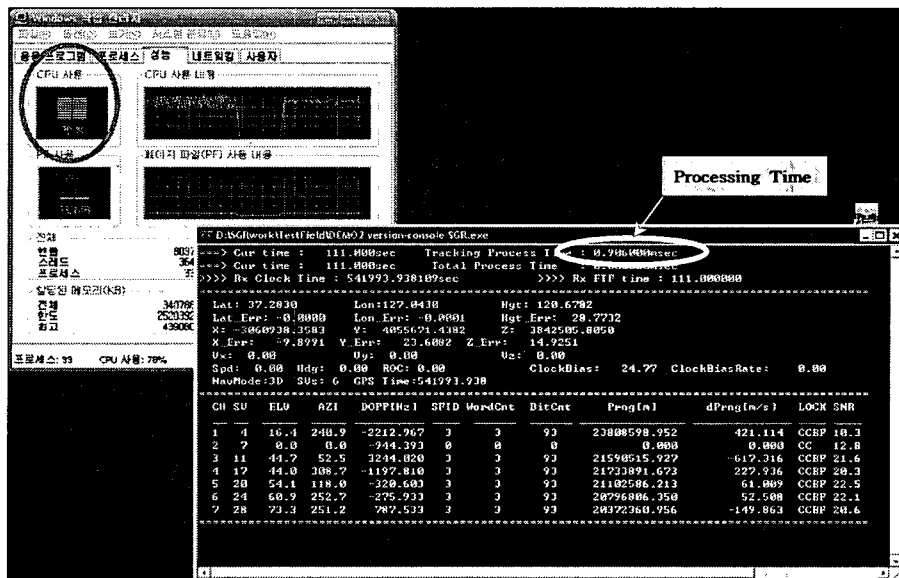


그림 19. 구현된 소프트웨어 GPS 수신기의 동작화면  
Fig. 19. Operation of The Implemented Software GPS Receiver

#### 4. 결 론

본 논문에서는 PC기반 실시간 소프트웨어 GPS 수신기를 구현하였다. 구현된 소프트웨어 GPS 수신기는 범용 프로세서를 사용한 PC에서 동작되도록 설계하였으며, 바이트형태의 중간주파수 신호데이터를 이용, 처리하도록 구현하였다. 이러한 바이트형태의 중간주파수 신호의 이용은 다양한 범용프로세서로의 손쉬운 적용이 가능한 점, RF부의 A/D 정밀도와 상관없이 데이터처리가 가능한 점, 그리고 신호감도를 개선할 수 있는 구조적 이점을 제공한다. 중간주파수 신호데이터 입력을 위한 데이터획득장치와 제어모듈, 큐구조의 공유메모리버퍼를 활용한 중간주파수 데이터 흐름제어모듈, 그리고 혼합기-적산기 모듈의 최적화를 통해 중간주파수 신호데이터의 처리가 실시간으로 가능하도록 설계, 구현하였다. 또한 SSE2와 OpenMP의 소프트웨어 설계기법을 적용하여 대부분의 처리시간을 차지하는 혼합기-적산기 모듈 및 중간주파수 신호데이터 제어모듈의 실시간성을 확보하였다. FFT-IFFT기반 신호획득모듈과 FLL-PLL 결합 신호추적모듈의 구현을 통해 고속의 신호획득성능 및 광대역의 신호추적성능을 제공함으로써, 향상된 이동체의 기동특성(medium dynamic condition)과 고속 초기위치획득시간 성능을 포함한다. 실제 신호환경에서의 동작실험을 통해 신호처리성능 및 실시간 동작성능을 확인하였으며, 일반적인 독립 GPS 수신기의 측위성능과 동일함을 확인하였다. 본 논문에서 구현된 실시간 소프트웨어 GPS 수신기는 현재 최대 8채널의 실시간성능을 확보한 상태이므로, 좀더 세분화된 최적화와 소프트웨어 설계변경을 통해 12채널정도의 동작성의 확보가 요구된다. 반면, 기존 소프트웨어 GPS 수신기와 비교해볼 때, 유연한 데이터 연동구조와 신호처리특성을 포함하고 있으므로 실시간성 보완과 관련된 추후연구를 통해 상용화 소프트웨어 GPS 수신기의 기반 설계기술로 활용이 가능할 것으로 판단된다.

#### 참 고 문 헌

- [1] Intel Corp., OpenMP Application Program Interface, Intel Corp., 17-112, May, 2005.
- [2] Intel Corp., Getting Started with SSE/SSE2 for the Intel Pentium4 Processor, Intel Corp., 2002.
- [3] Global Positioning Product Handbook, GEC Plessey Semiconductors, 1997.
- [4] B. W. Parkinson and J. J. Spilker Jr., Global Positioning System : Theory and Applications Vol. 1, AIAA, 352-355, 1996.
- [5] D. J. R. Van Nee and R. M. Coenen, "New Fast GPS Code-acquisition Technique Using FFT," Electronics Letters, Vol. 27, No. 2, Jan. 1991.
- [6] 고선준, 원종훈, 이자성, "소프트웨어 GPS 수신기용 신호처리 알고리즘 개발," GPS Workshop 2002, 2002.
- [7] R. Jaffe and E. Reichtin, "Design and Performance of Phase Lock Circuits Capable of Near-Optimum Performance Over a Wide Range of Input Signal and Noise Levels," IRE Trans. on Information Theory,

Vol. IT-1, March 1965.

- [8] E.D. Kaplan, Understanding GPS: Principles and Applications, Artech House, 1996
- [9] J. B. Y. Tsui, Fundamentals of Global Position System Receivers A Software Approach, John Wiley & Sons Inc., 2000.

### 저 자 소 개



#### 고 선 준(高先俊)

1972년 10월 29일생. 1997년 아주대학교 제어계측공학과 졸업. 1999년 동대학원 제어계측공학과 졸업(석사). 현재 동대학원 전자공학과 박사과정.

Tel : 031-219-2487

Fax : 031-212-9531

E-mail : sjko@ajou.ac.kr



#### 원 종 훈(元鍾焄)

1972년 3월 15일 생. 1995년 아주대 제어계측공학과 졸업. 1997년 아주대 대학원 제어계측공학과 졸업(석사). 2005년 동 대학원 전자공학과 졸업(박사). 2005년~현재 독일 뮌헨 소재 독일연방 국방대학(University Federal Armed Forces, Munich) 측지항법연구소(Institute of Geodesy and Navigation), Research Associate.

Tel : +49-89-6004-3553

Fax : +49-89-6004-3019

E-mail : JongHoon.Won@IfEN.BauV.UniBw-Muenchen.de



#### 이 자 성(李資城)

1949년 1월 27일생. 1975년 한양대학교 전기공학과 졸업. 1977년 University of Southern California 전기공학과 졸업(석사). 1981년 동 대학원 전기공학과 졸업(박사). 1981~1992년 Rockwell International Corp. 선임 연구원. 1992년~현재 아주대학교 전자공학과 교수.

Tel : 031-219-2482

Fax : 031-212-9531

E-mail : jsung@ajou.ac.kr