Modular GPS Software Radio Architecture

Neil Gerein, Alison Brown, *NAVSYS Corporation*

BIOGRAPHY

Alison Brown is the President and CEO of NAVSYS Corporation. She has a PhD in Mechanics, Aerospace, and Nuclear Engineering from UCLA, an MS in Aeronautics and Astronautics from MIT, and an MA in Engineering from Cambridge University. In 1986 she founded NAVSYS Corporation. Currently she is a member of the GPS-III Independent Review Team and Scientific Advisory Board for the USAF and serves on the GPS World editorial advisory board.

Neil Gerein is a Product Manager for NAVSYS Corporation's Receivers Group and is responsible for the management and development of NAVSYS' next generation of GPS receivers. He is currently completing his M.Sc. in Electrical Engineering and holds a BSEE in Electrical Engineering from the University of Saskatchewan.

ABSTRACT

Software radio technology provides advanced signal processing and multi-band multi-mode software-based RF receiver capabilities. When applied to GPS, this software radio technology provides the ability to add in new GPS waveform applications through only software and/or firmware modifications to the GPS receiver. A major benefit of this approach is that GPS users will no longer be forced to purchase new GPS hardware if their needs change or to take advantage of next generation GPS capabilities. Software applications to fit changing requirements can instead be used to upgrade the GPS software radio. GPS receivers will also not require unique GPS hardware but can be implemented as one of multiple waveforms being processed on a single software radio.

NAVSYS has developed a prototype modular GPS software radio that runs on a PC-based platform using a software and firmware reprogrammable architecture. In this paper this architecture is described and test results are presented with discussion of the GPS software radio performance in implementing the next generation of GPS waveforms and algorithms.

INTRODUCTION

The next generation GPS satellites (Block IIR and IIF) will include new GPS waveforms to improve the GPS performance for both military and civil users. A new M-code signal is planned to be added to the L1 and L2 frequencies that will improve the robustness of GPS to jamming.[1] For civil users, it is planned to add either a new civil PRN code (Lc) or the C/A code to the GPS L2 frequency. Two PRN ranging codes are planned to be transmitted on L5: the in-phase code (denoted as the I5-code); and the quadraphase code (denoted as the Q5-code)[2]. Current generation military and civil GPS User Equipment (UE) are not compatible with these new GPS waveforms and will need to be replaced.

The term "software radio" was coined in 1991 to signal the shift from the hardware-intensive digital radios of the 1980's to the multi-band, multi-mode, software-based radios planned for the year 2000 and beyond. Software programmable radio technology offers numerous advantages over the inflexible implementation of previous radio designs as the software reprogrammable environment allows for improvements or enhancements without altering the radio hardware. Updates to a software reprogrammable radio no longer require hardware changes but instead can be delivered on a CD downloaded through a network. The software radio concept also allows users to acquire relatively generic hardware and tailor its capabilities to their individual needs by choosing software that fits their specific application; this is analogous to the flexibility inherent in today's personal computers.

The DoD is migrating to a software radio approach to support next generation mobile communications with the acquisition of a Joint Tactical Radio System (JTRS). The JTRS will be a secure, multi-band, multi-mode, multi-channel software radio that supports a broad range of C4I requirements. Some of the waveforms planned to be included are illustrated in Figure 1.

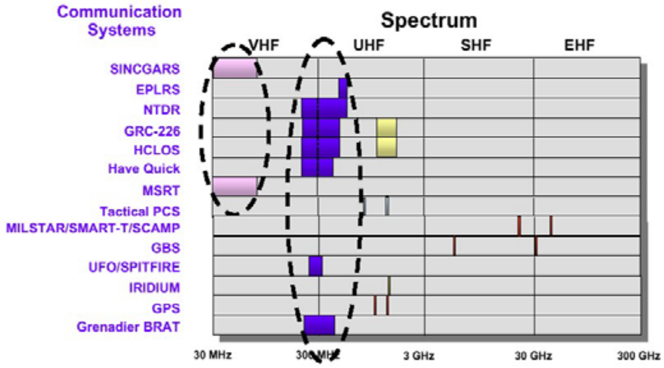

Figure 1 Joint Tactical Radio System (JTRS) Planned Waveforms

The development of civil and military software radio standards can be leveraged to embed next generation GPS waveform capability to support the GPS Modernization activities. In this paper, the design of a GPS software radio is described with test results to demonstrate the flexibility and high accuracy that can be achieved using this approach.

GPS SOFTWARE RADIO ARCHITECTURE

The GPS software radio application is designed to run on NAVSYS' digital GPS receiver hardware. This includes the components illustrated in Figure 2 and described below. The GPS software radio architecture shown in Figure 2 is designed to support a modular approach, in terms of hardware, firmware and software.

The receiver hardware can be configured with multiple component cards as needed depending on the number of antenna inputs and frequencies to be tracked, and the number of receiver satellite channels to be implemented.

The firmware is configurable to support different digital signal processing functions, as needed, including spatial processing (e.g. beam-forming), code generation (e.g., C/A or P(Y)) and code correlation and carrier mixing.

The modular software architecture allows the user to activate individual software modules, depending on the user's application. The GPS software radio hardware, firmware and software design is described further in the following sections of this paper.

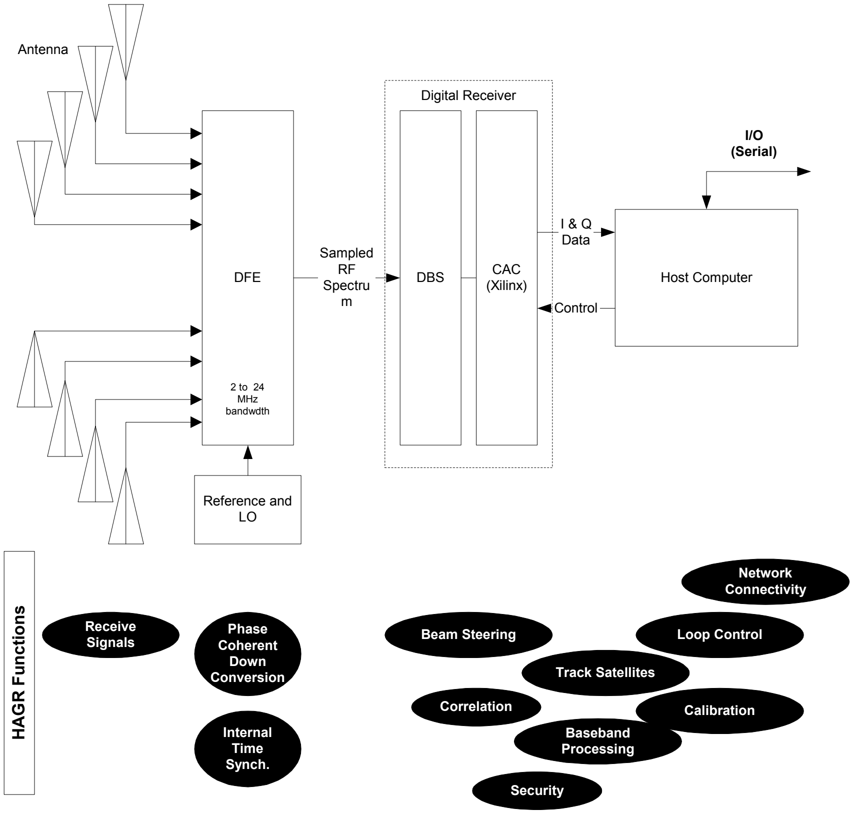

Figure 2 NAVSYS Software GPS Receiver Architecture

DIGITAL GPS RECEIVER HARDWARE

Digital Front-End Board

The digital front-end (DFE) board performs the function of phase-coherent down conversion and digitizing the received satellite RF signals. The software receiver architecture allows inputs to be provided from multiple antenna elements, which could be configured as an antenna array, such as in our High-Gain Advanced GPS Receiver (HAGR) product[3], or to receive signals at different frequencies, such as L1, L2 and L5. Each DFE board includes four separate RF channels, as shown in Figure 3. The input frequency of each individual channel is selected through the front-end filters. The input RF signals are mixed to a 70 MHz IF where they are sampled using a 12 bit analog-to-digital converter. The IF filter bandwidth can be selected from 2 to 24 MHz and the sample clock can be adjusted up to 56 MHz. The digitized signal is converted to a low voltage differential signal for transmission to the Correlator Accelerator Card.

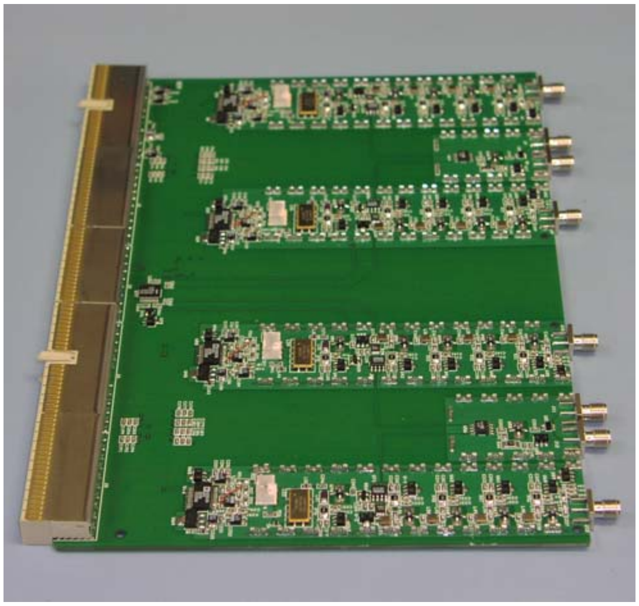

Figure 3 Digital Front End (DFE) Board

When using an analog-to-digital converter for direct IF sampling, the sampling frequency, f_s must be greater than, or equal to, twice the signal bandwidth Δf to avoid signal aliasing. This is know as the Nyquist criteria:

Equation 1

$$f_s \geq 2\Delta f$$

For direct IF sampling, the following equation ensures that the center of the IF, f_c, is placed in the center of a Nyquist zone (NZ). Each Nyquist Zone covers a frequency bandwidth of $0.5*f_s$. The conventional GPS signals have a 20 MHz bandwidth. Equation 1 says that our sampling frequency has to be more than 40 MHz. The M-code and L5 signals occupy a bandwidth of 24 MHz and so require a higher sample rate. The DFE board can handle sample rates up to 65 MHz. The sample clock is generated phase locked to the input 5 or 10 MHz clock, and with the L1 and L2 Local Oscillators (LOs).

Correlator Accelerator Card

The correlator accelerator card (CAC) includes the following functions: Code generation, code correlation, carrier mixing and I/Q accumulation. The CAC card can also be programmed to perform digital beam-steering when operating with multiple antenna inputs from an array. This functionality is provided for each of six channels, and is repeated for each code and frequency for every satellite channel tracked. The CAC card logic is implemented using Xilinx Field Programmable Gate Arrays (FPGAs) and can be reprogrammed through firmware downloads from the Host Computer. The CAC board layout is shown in Figure 4. The current generation CAC card firmware includes C/A and P(Y) (L1 and L2) code generation. As described later in this paper, a test version of the GPS M-code has also been coded for use in the CAC firmware.

Host Computer

A standard PC is used as the host computer. Our baseline configuration is to use an 850 MHz Pentium III CPU with 1 Gigabytes of DRAM and a 40 Gigabyte EIDE hard drive. This can be configured for desk-top, rack-mounted or portable operation. The DFE and CAC cards are installed on the PCI bus of the host computer.

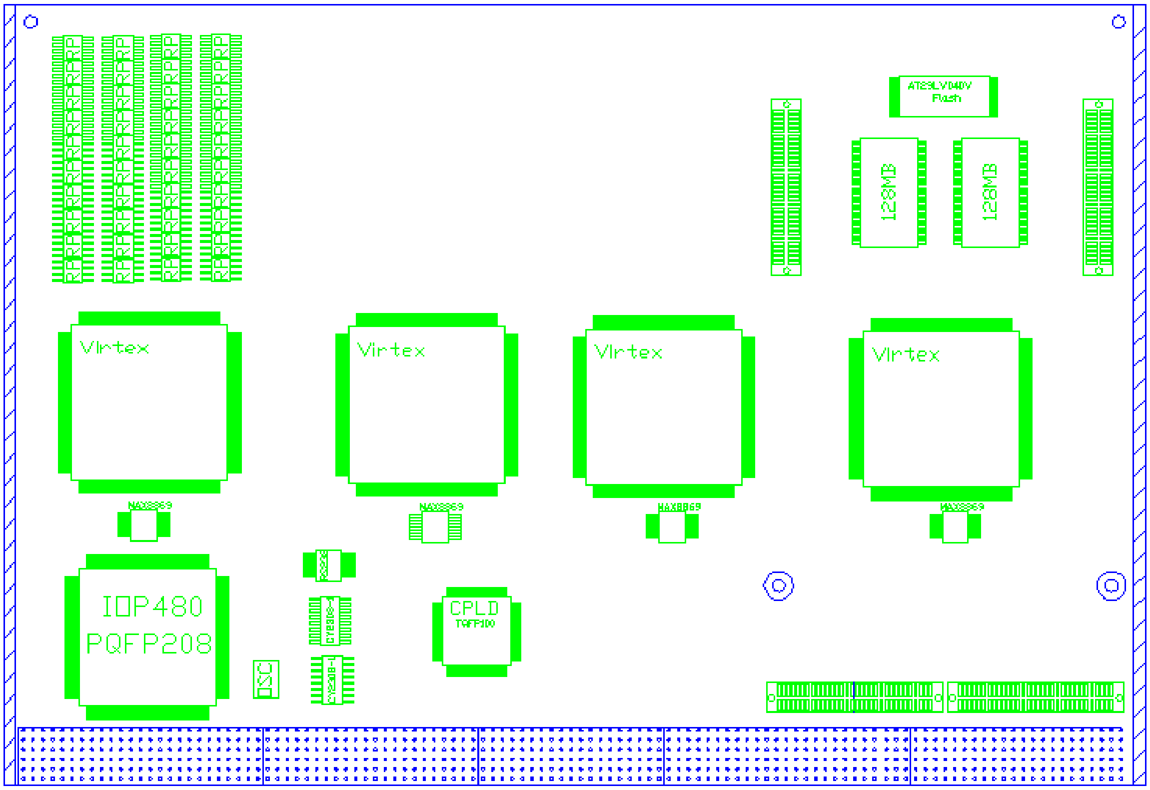

Figure 4 CAC Board Physical Layout

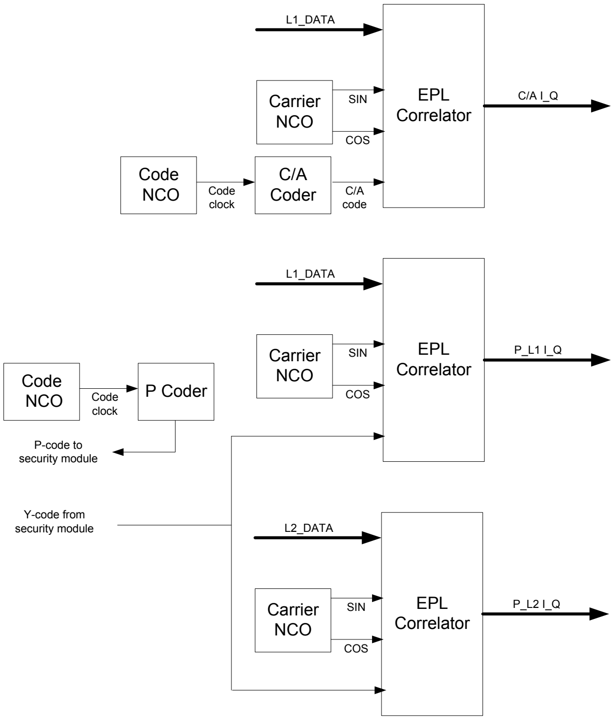

Figure 5 CAC Channel Firmware Blocks

MODULAR FIRMWARE DESIGN

The CAC Xilinx FPGA firmware used in our GPS software radio uses a set of standard firmware blocks. These blocks include accumulators for correlation, code and carrier NCOs, local bus interfaces, tap registers, and shift registers and counters for PRN generation. Modules are added to the device in a "drag and drop" manner. A binary bit file is generated by the FPGA design tools and save to disk. This bit file is loaded across the PCI bus under AGR software control to program the devices.

Figure 5 shows the firmware modules used to build a single C/A, P(Y) L1, and P(Y) L2 SV channel. The L1_DATA and L2_DATA busses are the sampled data from the DFE. These busses are generic in the sense that any sampled data may be input into the device, regardless of carrier frequency.

All modules contain control registers that are memory mapped so that they may be controlled by a host computer over a PCI bus. The main blocks in a SV channel are:

- Code NCO – Converts the software downloaded code phase and code frequency values, and produces the code clock for the PRN coders.
- Carrier NCO – Maps the software downloaded carrier phase and carrier frequency values to sine and cosine values to be used for carrier removal.
- C/A Coder – Contains the shift registers and logic to generate the C/A code.
- P Coder - Contains the shift registers and logic to generate the P code. The P code is passed off the FPGA to the security module and returns as Y code.
- EPL Correlator – This block contains the complex multipliers for code and carrier removal. Also included are Early/Prompt/Late shift registers, and integrate and dump registers.

To implement a new GPS code a new coder block can be created and dropped into the FPGA design. The new bit file for the FPGA can be loaded across the PCI bus and the software can be modified to provide the necessary control signals. No physical hardware on the CAC board needs to be changed.

MODULAR SOFTWARE DESIGN

The software application used in our GPS software radio is called the Advanced GPS Receiver (AGR). The AGR software is designed to provide both flexibility and high accuracy performance to allow a common software application to be adapted to meet a broad range of current and future GPS receiver requirements. The AGR software performs the following main functions:

1. Tracks GPS satellites
2. Computes Navigation solutions (standalone and differentially corrected)
3. Network interface with other computers
4. Outputs many different types of data to log files
5. Operate in Post process with logged data files

The interfaces between the component "modules" of the AGR software are shown in Figure 5. The ellipses represent software modules, the double-bordered rectangles represent hardware available with the PC, and single-bordered rectangles represent special hardware required by the AGR (i.e., the CAC card). The numbered items, 1 – 7 in Table 1, correspond directly to the AGR modules. Table 1 gives a basic functional description of the software modules in the current AGR software application. These are continually being expanded to add new capabilities as described later in this paper.

Table 1 Current AGR Modules and Their Functionality

Item	Module	Function
1	Track	Selects satellites, tracks them, and provides measurements to the rest of the system.
2	Remote	Takes the most current track measurements and differential corrections (if available), and computes the best possible navigation solution.
3	Ref	Takes the most current track measurements and the reference position from the configuration, and computes differential corrections.
4	API	Responds to commands from the client and gets the data requested by the client, formats it, and supplies it to the client.
5	Post	Reads data in from input post-process files, time align its, and injects into the system to mimic real-time system.
6	Exec	Orchestrates the flow of the data throughout the system.
7	Navigation	Provides navigation services to both the Remote and Reference.

Each software module can be configured by the user through a set of keywords. A sample set of keywords from the Track module is given in Table 2. A full list of all of the keywords that can be programmed are included in reference[4]. These allow definition of the RF hardware class being used (e.g. frequency plan, sample rate, filter bandwidths), the signals to be processed (e.g. PRN codes, frequencies) and the types of tracking loops and navigation algorithms to be employed.

Table 2 Example Track Configuration Keywords

Keyword	Description
Sample Clock Freq (kHz)	Frequency sampling rate of the master data clock going into the CAC board, in kHz.
Channel signal class	Type of signal to be tracked. Jam_detect and Jam_track are optional modules.
Code Type	Type of GPS code to be tracked.
Nav Frame Format	Type of broadcast Navigation message to decode. GPS is conventional GPS nav message. WAAS is the FAA Wide Area Augmentation Signal format.
Nav Data Period	Period of navigation data bit (msec). Use 20mSec for GPS and 2mSec for WAAS.
IQ Data Select	Type of IQ data input.
Signal Sqr. Minimum	Minimum scale factor for tracking loops (0 dD-Hz SNR)
Signal Sqr. Maximum	Maximum scale factor for tracking loops (theoretical maximum 60 dBHz)
SNR Accum Period (ms)	Period over which the SNR is accumulated
IF Freq (Hz)	Intermediate frequency of input GPS signal in Hz
Doppler Bin Size (Hz)	Doppler frequency bin size in Hz
P-Code Search Range (chips)	P_code search range (chips)
Code State Mode	End code state Stop at: 0 – Inactive; 1 - Wide code loop; 2 - Medium code loop; 3 - Narrow code loop
Carrier State Mode	End carrier state. Stop at: 0 - Not active; 1 - Wide FLL; 2 - Narrow FLL, Wide PLL; 3 - Narrow PLL (no nav data); 4 - Narrow PLL (with nav data)
Codeless State Mode	End codeless state Stop at: 0 – Wide search; 1 – Wide FLL; 2 - Narrow FLL, Wide PLL; 3 - Narrow PLL (no nav data)
Thresholds: Dismiss	10 msec IQ_sqr threshold (Pfa = 5e-4 = pmd at 33 dB-Hz)
Thresholds: Declare	100 msec IQ_sqr threshold (PMD < 1e-4 at 33 dB-Hz)
Thresholds: Code	100 msec IQ_sqr threshold while carrier tracking falls back to search
Thresholds: Signal	Sig_sqr threshold for -> state 1 (32 dB-Hz for carrier tracking)
Thresholds: AFC	AFC locksum for <100 Hz state 1 ->2 transition (32 dB-Hz)
Thresholds: Lock sum	Threshold for PLL pull-in and state 2 -> 3 transition (32 dB-Hz)
Thresholds: Phase error	Threshold for 1-msec carrier in state 2->3 transition (32 dB-Hz)
Thresholds: Hysteresis	Back to wide FLL (2 -> 1) at (t_afc - t_hysteresis) (28 dB-Hz)
Thresholds: PLL	Back to wide PLL.
Filter Consts: AFC Alpha 1; AFC Alpha 2	AFC locksum (over track.acq_integ_period mSecs) filter time constant s
Filter Consts: IQ Alpha 2; IQ Alpha 3; IQ Alpha 4	IQ locksum filter time constant
Filter Consts: Alpha Codeless	Codeless locksum filter time constant
Filter Consts: Alpha Noise	Signal power (100 msec) filter time constant (1 second)
Filter Consts: Alpha Carrier MSE; Alpha Code MSE	Carrier mse (1 msec) filter time constant (100 ms)
Freq Lock Loop 1st BW (Hz); 2nd BW (Hz)	State 1 frequency lock loop bandwidth (Hz)
Phased Lock Loop BW (Hz)	Phase lock loop bandwidth (Hz)
IQ Lock Loop BW (Hz)	IQ loop bandwidth (Hz)
Delay Lock Loop BW (Hz)	Delay lock loop bandwidth (Hz)
Acq. Integ. Per. (mS)	Acquisition integration period (msec)
Track Integ. Per. (mS)	Tracking integration period (msec)
Phase Lock Loop Mode	Phase lock loop mode
Mid E-L Spacing (chips)	Code chip spacing of Code state 2
Narrow E-L Spacing (chips)	Code chip spacing of Code state 3

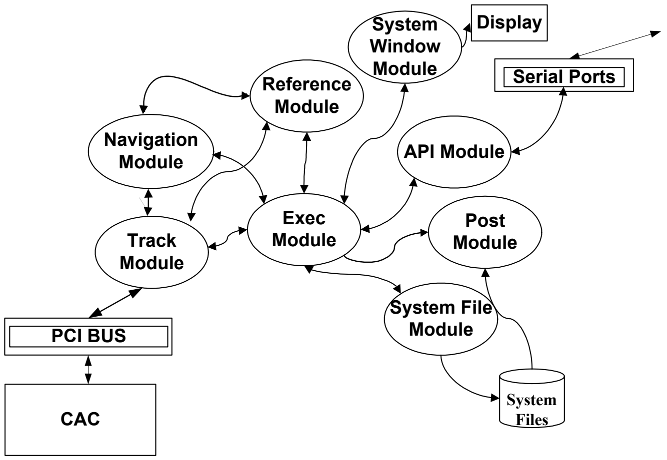

Figure 6 AGR Top-Level Modules

TRACK MODULE

The Track module is one of the core modules of the system. Its major tasks are defined in Table 3.

Table 3 Track Module Tasks

1. Continuously determine the best set of satellites to track.
2. Obtain raw tracking information from each channel from the CAC.
3. Run the tracking loops with the raw tracking data.
4. Provide the track measurements to other system modules at the rate they have requested.
5. Output track measurement and Raw_nav data to the Track Measurement System File.
6. Update the Track, Channel states, and Raw Nav watch data sets and deliver to the Navigation module for processing

Figure 7 shows the Track module functions. These functions are used by each Track sub-module in an object oriented manner. The functions are driven by CAC card interrupts, which come at 1 kHz.

The CAC card receives the digitized GPS signal, modulates a different GPS C/A code PRN sequence in each of its channels, and provides the correlation values to the PC at a 1 kHz rate via a hardware-interrupt. At each interrupt, the CAC function reads the raw tracking correlation information from each channel. This module then runs the tracking loop algorithms on each channel. The Track function then sends new channel update data back to the CAC module, which re-programs the CAC with the new information. All this must take place before the next interrupt comes one millisecond later.

For each channel, the Track sub-module runs three separate tracking loops:

1. Search/Acquisition Loop
2. Code Tracking Loop
3. Carrier Tracking Loop

Figure 8 shows the three loops the Track sub-module runs. The state value reported by Track is a decimal number in which the 100’s place is the state of the carrier loop, the 10’s place is the state of the code loop, and the 1>s place is the state of the search loop. Any state above 400 indicates that the Track module is tracking the SV in that channel and is demodulating the data on the signal correctly.

Figure 8 also shows how the Track configuration parameters are used. It should be noted that each channel has its own independent set of Track configuration parameters that allow the type of tracking loop and the thresholds for state transition to be defined for each code tracked.

The Track module can be set to output data to the Track Measurements and Raw Nav data files. The track measurement file can be output at different rates, from 1 Hz to 1 kHz. These files allow low level access to the tracking states and also the raw I/Q data from the CAC cards.

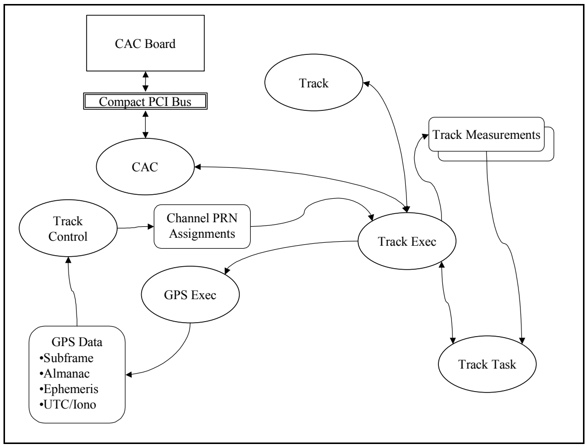

Figure 7 Track Module Sub-Systems and Data Flow

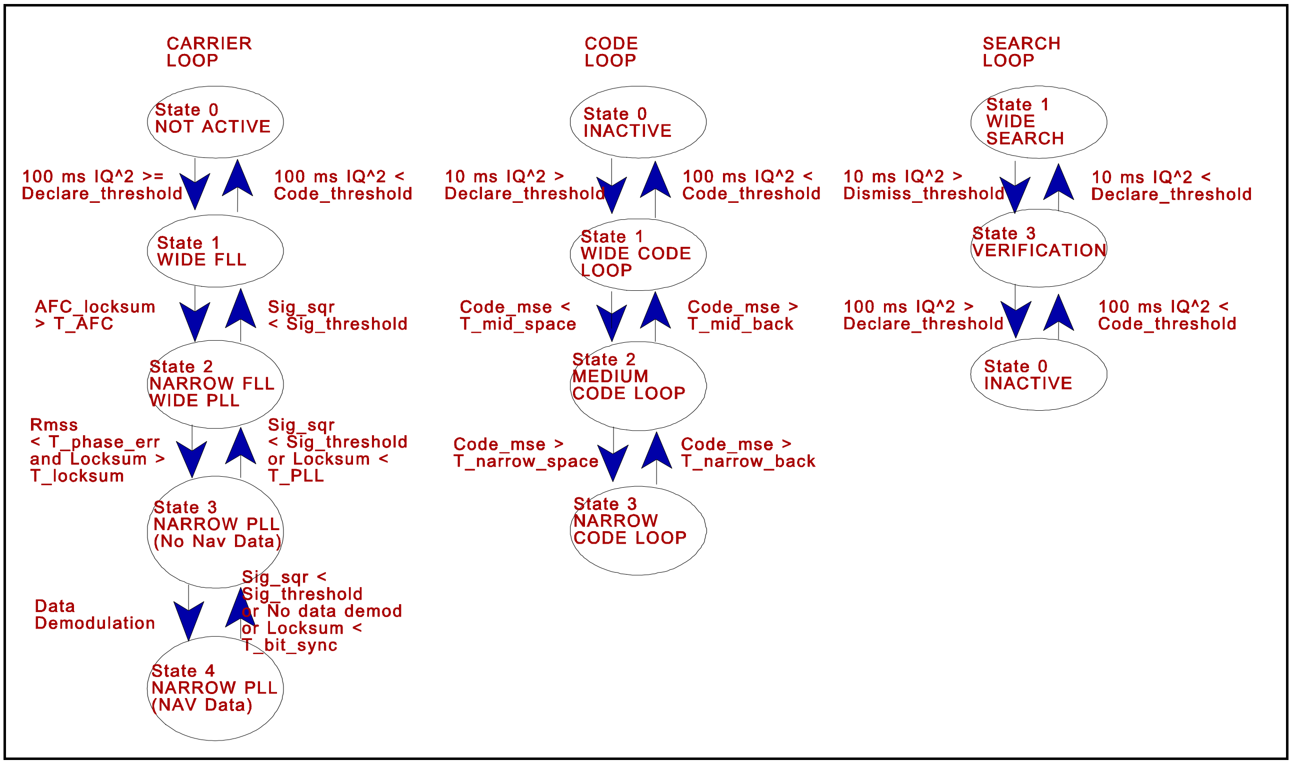

Figure 8 Tracking States Flow Diagram

GPS SOFTWARE RADIO EXAMPLE TEST RESULTS

The following tests were run to demonstrate the inherent precision, flexibility, and adaptability of the GPS software radio approach.

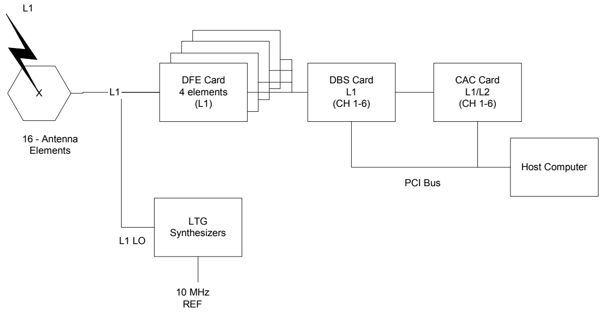

Figure 9 Sixteen Element L1 Array High Accuracy Test Setup

High Accuracy Tracking

The all-digital nature of the modular software radio allows extremely high accuracy satellite observations to be made. To demonstrate the high accuracy possible using this architecture, C/A and P(Y) code observations were made using a 16-element L1 antenna array[5]. The modular configuration for this testing is illustrated in Figure 9. To sample the data from all antenna elements 4 DFE cards were used in the system. A PCI based NAVSYS Digital Beamsteering (DBS) card module was installed in the host computer to provide the digital weights and phase shifts for the antenna array. The output from the DBS was connected to the input of the CAC. To provide the DBS control the standard AGR software was compiled with the beamsteering control module.

The RMS white noise on the pseudo-range observations was computed by differencing the PR+CPH measurement shown in Figure 10. The short-term noise on this observation is due to the receiver white noise. (The medium term cyclic error on the PR+CPH is due to multipath and the longer term drift due to ionosphere). The observed PR measurement noise is shown in Figure 11 and Figure 12 for all of the satellites tracked for both the C/A and P(Y) code observations. The observed PR RMS noise is shown in the bottom figure. The predicted PR noise, based on the observed C/N0 for each satellite, is also plotted (top figure). The observed PR noise shows good correspondence with the predicted values. In Figure 13, the PR RMS errors for the C/A code and P(Y) code HAGR observations are plotted as a function of the C/N0 observed. For C/N0 values above 52 dB-Hz, the P(Y) code HAGR provided pseudo-range accuracies of 5 cm (1-sigma) while for C/N0 values above 55 dB-Hz the C/A code observations were accurate to 15 cm. These values are for 1-Hz observations without any carrier smoothing applied. The mean observed RMS accuracies are summarized below in Table 4. These results show that the software receiver is capable of observing the GPS measurements to high resolution.

Table 4 Measured RMS PR Noise (m)

SVID	C/A HAGR RMS PR	P(Y) HAGR RMS PR
1	0.239	0.054
3	0.284	0.056
8	0.200	0.045
11	0.278	0.059
13	0.252	0.059
14	0.214	0.049
20	0.222	0.050
21	0.252	0.058
22	0.248	0.047
25	0.202	0.044
27	0.183	0.044
28	0.236	0.055
29	0.225	0.050
30	0.477	0.089
31	0.325	0.055

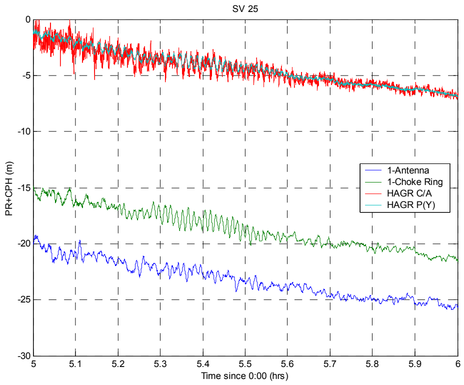

Figure 10 PR+CPH (m) - SV 25

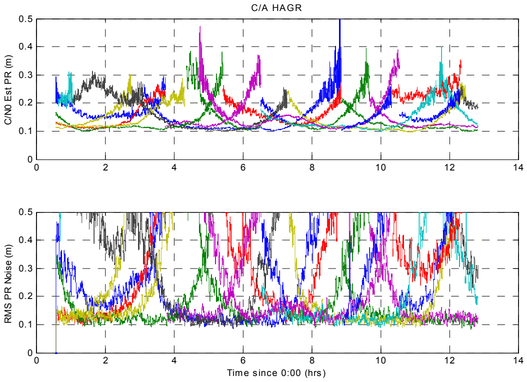

Figure 11 C/A HAGR RMS Pseudo-Range Noise (m)

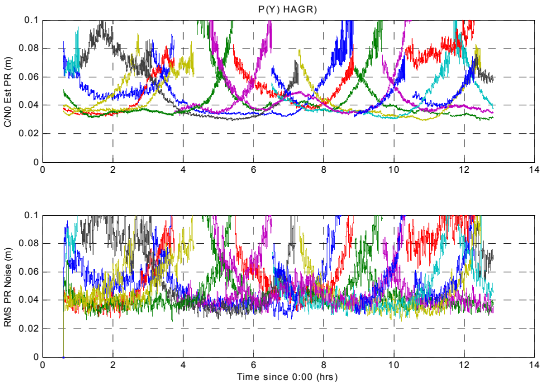

Figure 12 P(Y) HAGR RMS Pseudo-Range Noise (m)

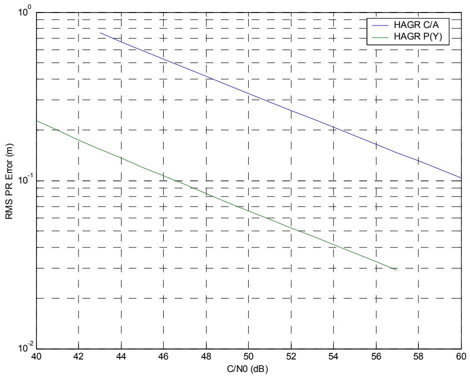

Figure 13 C/A & P(Y) HAGR RMS PR error vs C/N0

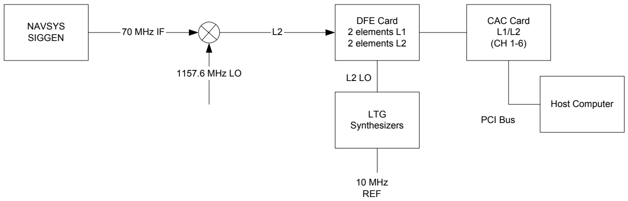

Figure 14 L2 C/A Code Test Setup

<u>Multi-Frequency Operation</u>

A major benefit of the GPS software radio architecture is the ability to include measurements from different frequencies. The DFE cards can be used to receive different RF frequencies simply by changing the RF front-end filter and the frequency of the LO input used to mix the signals to the 70 MHz IF. The AGR software is able to track signals at different frequencies simply by changing the Track Module key words. To illustrate this capability, the test set-up shown in Figure 14 was used to insert an L2 signal modulated with C/A code into an L1/L2 DFE card. The GPS software was simply told to look for the C/A code signal at L2 instead of L1. The tracking results are shown in Figure 15.

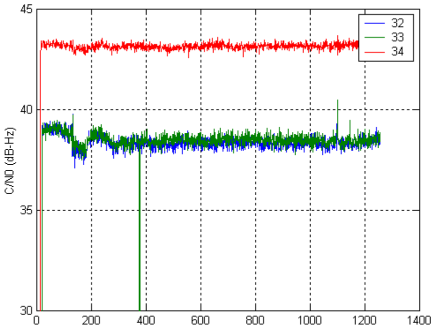

Figure 15 L2 C/A code Tracking

Adding New Codes

Under GPS modernization, the next generation GPS satellites will have the capability to broadcast new military and civil codes. The reprogrammability inherent in the GPS software radio will allow these new PRN codes to be tracked by making software and firmware changes to the software radio. To illustrate this capability, the code generation module was modified to generate a test M-code signal with the characteristic BOC(10,5) modulation shown in Figure 16. The C/A, P, and M-code signal generated by the CAC using the code generation firmware is shown in Figure 17 to Figure 19.

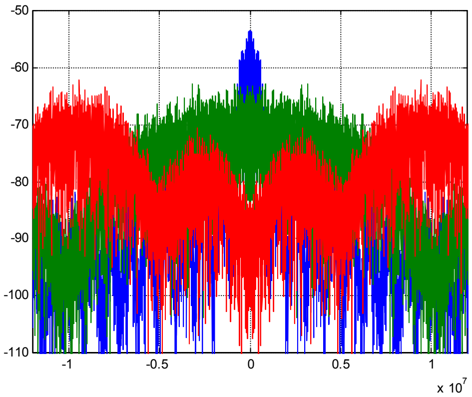

Figure 16 C/A, P and M-code spectral characteristics

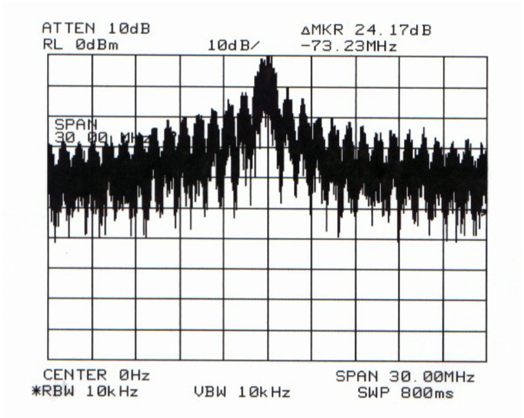

Figure 17 C/A Code CAC output

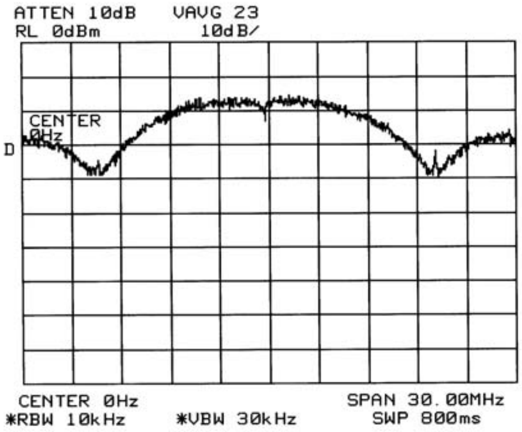

Figure 18 P Code CAC Output

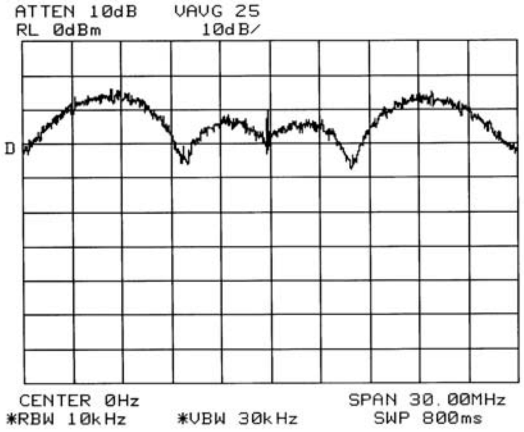

Figure 19 M-code modulation from CAC output

CONCLUSION

The NAVSYS modular GPS software radio design has been described in this paper and test results were included to demonstrate the following capabilities and benefits.

- **High Performance Operation**. The digital signal processing inherent in the software radio approach allows the GPS observations to be derived to high levels of accuracy. The low level access to the GPS signal structure also allows optimized signal processing techniques to be applied to further improve signal processing, such as multipath minimization techniques[6], digital beam-steering and null-steering algorithms and space-time-adaptive-processing (STAP) or space-frequency-adaptive-processing (SFAP) methods.
- **Multi-Frequency, Multi-Mode Operation**. The nature of the software radio simplifies the introduction of additional frequency channels and the tracking of new codes. New frequencies are added through simple changes to the RF-to-digital front-end filters and selection of a new LO. New codes are added through firmware and software modifications.

- **Flexibility and Upgradeability.** The reprogrammable nature of the GPS software radio allows it to be upgraded through firmware and software modifications to accommodate next generation GPS signals and user applications.

REFERENCES

[1] Navstar GPS Military-Unique Space Segment/ User Segment Interfaces, ICD-GPS-700, 21 June, 2001.

[2] Navstar GPS Space Segment/User Segment L5 Interfaces, ICD-GPS-705, 16 April, 2001

[3] A. Brown and N. Gerein, "Test Results from a Digital P(Y) Code Beamsteering Receiver for Multipath Minimization," Proceedings of ION 57th Annual Meeting, Albuquerque, NM, June, 2001.

[4] "AGR User's Guide," NAVSYS Document AGR 00-14.

[5] A. Brown and N. Gerein, "Test Results from a Digital P(Y) Code Beamsteering Receiver for Multipath Minimization", Proceedings of ION 57th Annual Meeting, Albuquerque, NM, June,2001

[6] A. Brown, N. Gerein, L. Savage, "Multipath Characterization using Digital Phased Arrays," ION 57th Annual Meeting, Albuquerque, NM, June, 2001.