# DYNAMICALLY RECONFIGURABLE SOFTWARE DEFINED RADIO FOR GNSS APPLICATIONS

Alison K. Brown (NAVSYS Corporation, Colorado Springs, Colorado, USA, abrown@navsys.com)

## ABSTRACT

Historically, the military has used special purpose Global Positioning System (GPS) radios for radio navigation. This has the disadvantage of locking users into fixed technology solutions designed to meet a fixed set of requirements. Software Defined Radios (SDR) have the advantage of being able to easily adapt to provide new capabilities using current generation technology. Continued improvements in SDR technology are enabling their use for Global Navigation Satellite Systems (GNSS) applications that require small form factor, low-power designs. This has the added benefit of allowing the signal processing algorithms for future GPS signals to be included in the GNSS SDR design without changing or modifying the hardware of the GPS receiver. This paper describes a GNSS SDR reconfigurable architecture that leverages the flexibility of a SDR to re-use resources for processing the different GPS signals. The paper also discusses the benefits of this GNSS SDR approach for military and civil users compared with conventional special purpose GNSS receiver solutions.

## 1. Introduction

Recent advances in the GPS constellation have resulted in additional signals being made available for both military and civil applications. Figure 1 shows the modernized GPS signal structure which includes signals broadcast on three frequencies (L1, L2 and L5) and including both the legacy GPS modulation codes (C/A and P(Y)), additional civilian codes on L2 and L5 (the L2c and I5/Q5 codes) and also additional military codes on L1 and L2 (M-code)[1, 2, 3]. In the next few years, GNSS users will also be able to access new signals from other satellite constellations, including the European Galileo, the Russian Global Navigation Satellite System (GLONASS), the Japanese Quasi-Zenith Satellite System (QZSS), the Indian GPS Aided Geo Augmented Navigation (GAGAN) and the Chinese Compass Navigation Satellite System.

Historically, GNSS receivers have been designed with dedicated channels each capable of tracking only a single satellite code. When operating using only the C/A code signals from the GPS satellites, this has been manageable and GPS chip sets routinely allow tracking of twelve or more satellites at the same time. As the number of codes and frequencies increase, the demands on a conventional GPS receiver get higher. With the GPS satellites alone, a next generation receiver could be required to operate on three frequencies each using a different code which would require triple the number of channels when using a conventional receiver design. Moreover, to track the different signals broadcast by other GNSS satellite systems, even more channels would be required to be set up to track their new codes. The increase in the number of channels using a conventional receiver design would also result in increased device size and correspondingly higher power operation.

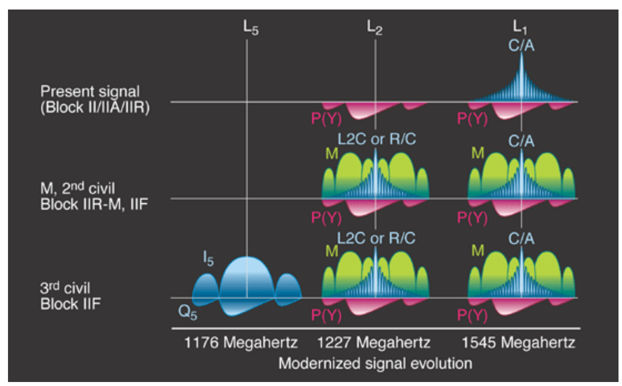


**Figure 1 Modernized GPS Signal**

In order to obtain a good navigation solution, it is only necessary to operate enough tracking channels in a GNSS receiver to obtain sufficient satellites in view to achieve good geometric dilution of precision. That generally only requires between 4-6 satellite signals. In this paper, we describe a GNSS Software Define Radio (SDR) architecture that allows for dynamic reconfiguration of the SDR channel resources to track different GNSS satellite codes and frequencies. With this approach, the flexibility of the SDR can be leveraged to implement a full-function GNSS receiver capable of leveraging any of the GNSS signals without requiring massive numbers of parallel channels to operate.

## 2. GNSS SDR Design

NAVSYS Corporation is currently leveraging our prior GPS SDR development efforts[4, 5, 6] to design a miniaturized SDR architecture with low-power design features and dynamic reconfiguration of the receiver channels to allow different GNSS frequency bands and signal codes to be processed by each channel. The SDR design being developed is flexible enough to cover the GNSS frequency bands for L1 and L2 operation and the new civil L5 frequencies using either the military or civil codes. The design is flexible enough to handle in the future the signals from other GNSS satellite systems such as Galileo or GLONASS.

As illustrated in Figure 2 and Figure 3, the GNSS SDR uses three RF channels to receive the L1 (1575.42 MHz), L2 (1227.6 MHz) and L5 (1176.45 MHz) signals. The digital outputs from each of these RF channels, termed Digital Antenna Elements (DAEs), are then provided to the SDR baseband processor. If any RF channel is not being used it will be disabled to save power.

The entire receiver baseband processing is being implemented on a single Xilinx Virtex-6 FPGA (Field Programmable Gate Array). The FPGA is initially loaded from external flash memory with a base configuration to enable communication to a host input device. The user would then be able to select any combination of GNSS codes to be tracked by the six receiver channels. Utilizing Xilinx dynamic partial reconfiguration, the receiver channels are loaded from external flash memory and configured by the host to acquire and track specific satellite signals.

If the user decides to track different GNSS codes, that change can be made at any time. Once the system receives the command to reconfigure from the host, the FPGA will read the required partial reconfiguration bit files from external flash RAM and use that to configure the selected channel. RF channels and encryption key management modules will be enabled or disabled as needed. Tracking operations of the other five channels will not be affected by the reconfiguration.

At initialization, the new receiver channel will be handed accurate knowledge of time that will allow it to begin tracking satellites extremely quickly. This will be done without powering down the system that would result in a "cold start" state of satellite tracking.

In Figure 2, a conventional C/A code operation is shown where all of the channels are set to perform C/A code correlation using only an L1 RF Channel (termed Digital Antenna Element). In Figure 3, the SDR configuration is shown where the resources are shared between tracking satellites on the L1 C/A code and also on the L2C code.

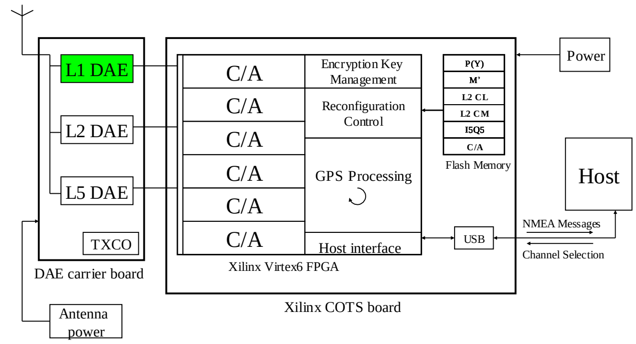


**Figure 2 Default configuration using L1 C/A code**

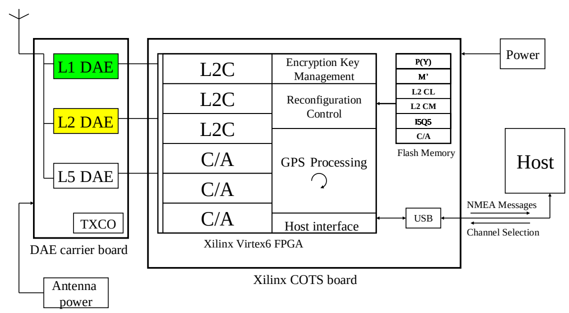


**Figure 3 Change to L1 C/A and L2C tracking**

## 3. Xilinx Baseband Processor Design

The Xilinx Baseband Processor design is illustrated in Figure 4.

A Partial Reconfiguration region is established for each GPS processing channel. Each processing channel follows a common design consisting of signal selection, signal and code NCOs, GPS code generators, correlation channels, and a register interface. The signal selection code at the channel front end determines which RF front end (L1/L2/L5) is processed and which GPS crypto variable is used for the military code generation. The number of correlators used in each channel depends on the complexity of the signal being processed by that channel.

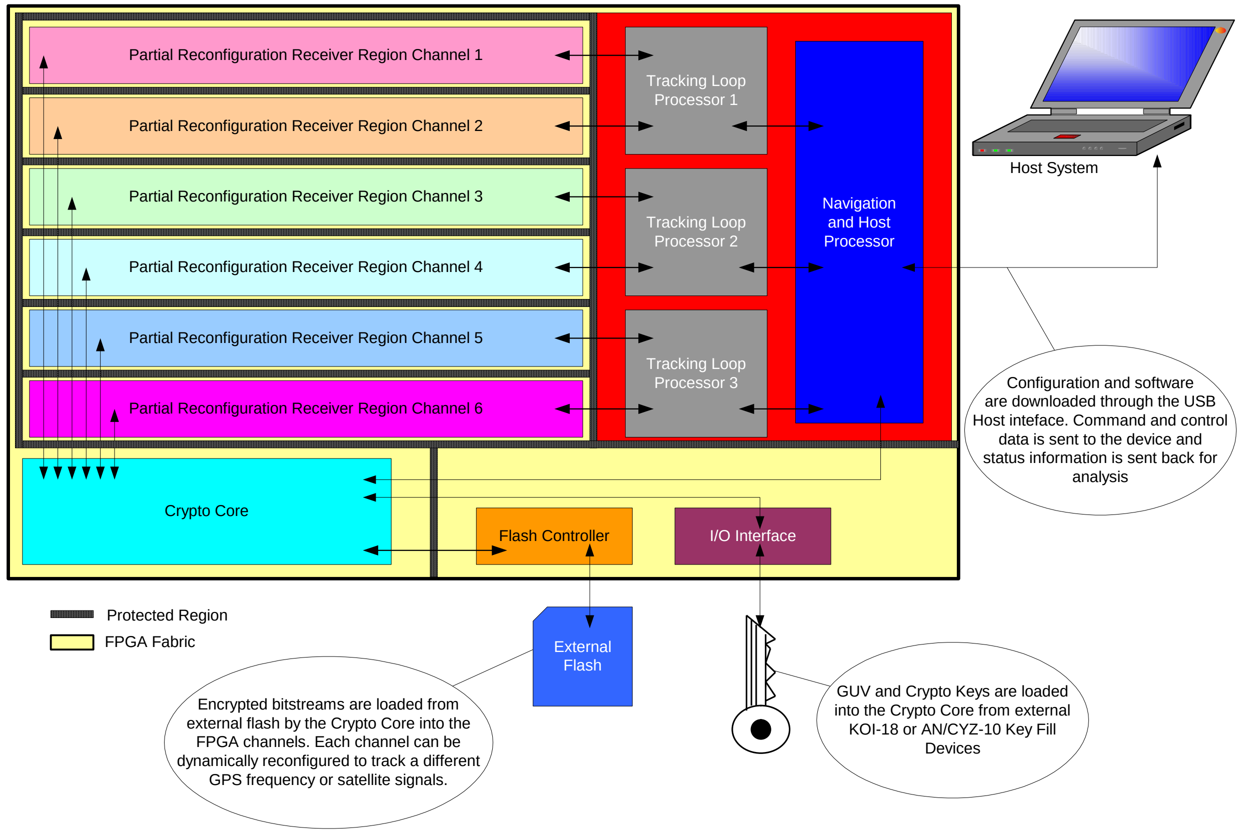


**Figure 4: Xilinx Baseband Processor Design**

A register interface between the Partial Reconfiguration channel and the Tracking Loop processor provides the means for the tracking loop processor to set up NCOs, code generators and signal selection. It is also used to transfer correlator results to the tracking processors. The register interface for each channel is mapped into the memory of one of the tracking loop processors in the central section. The tracking loop processors are Xilinx MicroBlaze instantiations. Currently, two tracking channels are handled by each of three MicroBlaze processors. Using a memory-mapped register interface provides a simple and efficient data path between the channel hardware and the tracking processor. Each tracking loop processor handles two tracking channels and communicates with the navigation processor and crypto core using serial interfaces. A fenced area on the Xilinx chip is used to store GPS keys and perform the GPS crypto variable generation.

An additional MicroBlaze processor in the central section implements the GPS navigation computation and also provides the host interface. Navigation can use GPS navigation message data extracted from satellite signals or provided via the host interface if the host has a network connection. The host interface is USB with a simple NMEA-format command set that is used to configure/control the receiver and to obtain the PVT solution from it. If the host is an SCA-based Joint Tactical Radio System (JTRS) radio then a small SCA component running on the host's red side provides an adapter between the USB interface to the GNSS device and the SCA applications on the JTRS radio.

The internal fencing is built from unused configuration logic blocks (CLB). In the fence, no routing or logic can exist. This provided at least three physical failures before the separation boundaries are breached[7].

## 4. Dynamic Reconfiguration

Using Xilinx's dynamic reconfiguration capability[8], each of the FPGA channels in the GNSS SDR can be dynamically reconfigured to track a different GPS frequency or satellite signal. The common base configuration is programmed into the FPGA when it first boots from the flash memory. The GNSS SDR user can then decide what processing will be required in each of the six reconfigurable channels. The appropriate FPGA bitstream is then loaded into each of the dynamically reconfigurable areas from the flash memory. The size of each of the reconfigurable regions is fixed and determined by the complexity of the largest processing algorithm estimates of the required FPGA resources for each of the supported GPS signal types. Figure 4 illustrates how the FPGA bitstreams are loaded into the dynamically reconfigurable areas. The partial bitstream loading is done through the Xilinx Internal Configuration Access Port (ICAP) [9]. This internal port allows for the reconfiguration image to be decrypted and verified before being applied without ever leaving the FPGA where it could be tampered with. In the current family of Xilinx FPGAs, the ICAP interface is a 32 bit bus operating at a maximum frequency of 100MHz. This yields a maximum throughput of 3.2Gb/second. This will allow for any channel to be reconfigured in approximately 1.5 milliseconds.

## 5. Software and Firmware Interaction

All of the processors within the GNSS device are Xilinx MicroBlaze processors running code written in C/C++. Using the MicroBlaze approach allows a good tradeoff between dedicated hardware (the correlator channels) and software to implement each of the functions in the device. The tracking loops are implemented partially in hardware and partially in software. The software uses no operating system. It is implemented as a simple state machine which makes it easy to program and debug and ensures very well defined performance at runtime. The navigation processor is also a MicroBlaze running several tasks in a single state machine. The single navigation processor communicates with the tracking loop processors to set up and monitor the six tracking channels. The navigation processor also implements the interface to the host. Ephemeris data for the position, velocity and time (PVT) solution calculation can come from the navigation message extracted from the GPS signal channels (where available) or it can be provided by the host.

Figure 5 shows how the reconfigurable FPGA firmware and software components are used to track a set of GPS satellite signals. In this example, one channel is being used for C/A and two more are shown processing either P(Y) or the L5 I5 & Q5 signals. When at least one channel is used for C/A signal processing that channel can extract the GPS navigation data message sub-frames and generate the satellite almanac and ephemeris data. Each signal processing channel has an FPGA configuration that implements the correlation processing. The remainder of the tracking loop is implemented in a software module. The software modules are also reconfigured based on the tracked signal type for each channel. The GNSS SDR design allows for any combination of the supported GPS signals to be implemented on each of the processing channels. The output of a single processing channel is a pseudorange measurement which is fed to the navigation solution computation component together with time and satellite ephemeris data. The computed position is available as an output from the GNSS SDR device to the host.

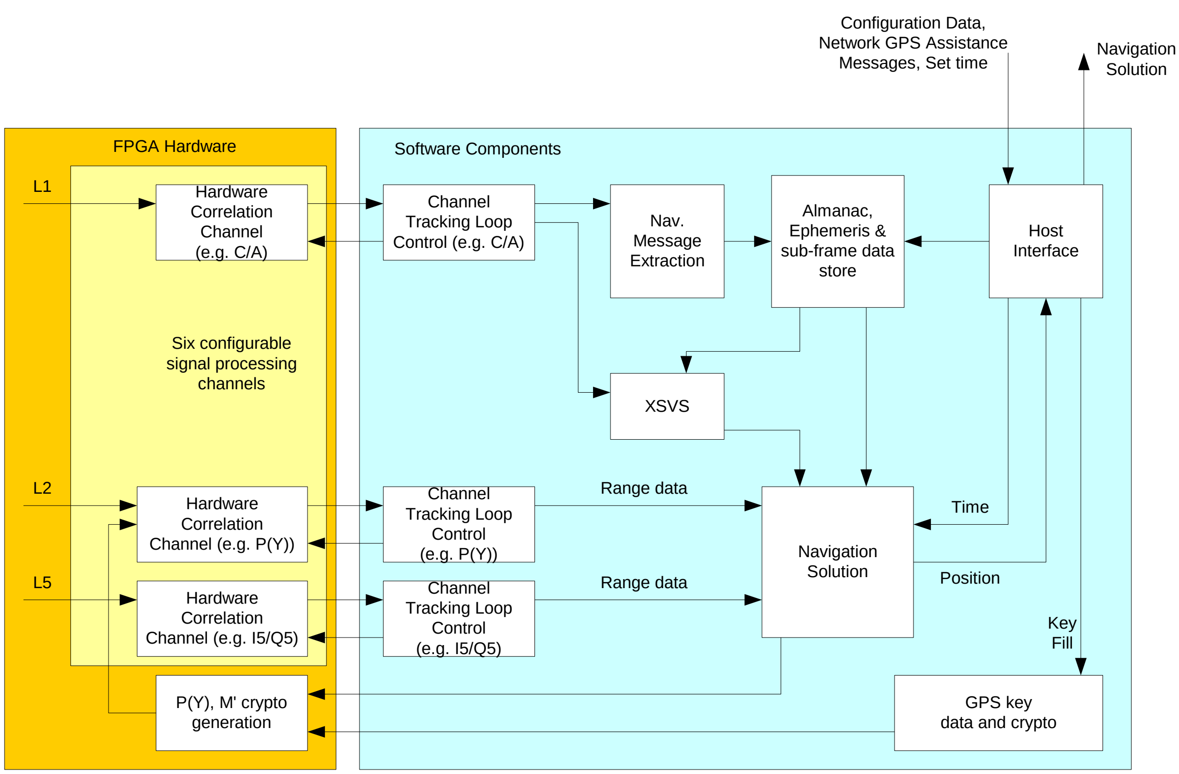


**Figure 5: Data Flow for Navigation Solution Computation**

## 6. Conclusion

The dynamic reconfiguration of the GNSS SDR channels allows re-use of the existing SDR resources without requiring dedicated channels to be implemented for any GNSS code/frequency pair. Only the receiver channels required are loaded and running on the FPGA at any given time. This allows for smaller devices to be used, consuming less power and having a smaller footprint.

With the flexibility provided by a GNSS SDR, another advantage is that as new GNSS satellite signals or codes become available the SDR can be upgraded to handle these signals without deploying new hardware. Only the FPGA configuration file stored in flash memory would need to be changed. Upgrades could also be added to improve the GNSS receiver's performance including innovative algorithms to enhance the tracking performance in GPS-degraded environments, reduce multipath, or adapt to the presence of detected interference.

This work is sponsored under the AFRL/RYDR contract FA8650-10-C-1733.

## 7. References

[1] Navstar GPS Space Segment/Navigation User interfaces, IS-GPS-200E, 8 June 2010

[2] Navstar GPS Military-Unique Space Segment/User Segment Interfaces, ICD-GPS-700A, 11 March 2004

[3} Navstar GPS Space Segment/user Segment L5 Interfaces, IS-GPS-705A, 8 June 2010

[1] Low-Power, Software Defined Global Positioning System (GPS) Receiver (GNSSuSDR), Contract No.: FA8650-10-C-1733.

[2] Low cost, lightweight, low power Precise Positioning System (PPS) GPS Solution for Software Defined Radios Contract No. N00039-09-C-0132

[3] Embeddable Software Programmable Radio, Contract No. N00014-09-C-0312

[4] Mark McLean and Jason More, "FPGA-based Single Chip Cryptographic Solutions," Military Embedded Systems, March 2007. http://www.mil-embedded.com/pdfs/NSA.Mar07.pdf

[5] "Partial Reconfiguration User Guide," Xilinx, UG702 (v 12.1) May 3, 2010 http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/ug702.pdf

[6] A. Brown, B. Johnson, Y. Lu, and P. Brown, "Low Power SDR Networked Architecture for Digital Camera Image Tagging," SDR '09 Technical Conference and Product Exposition, Washington, DC, December 2009

[7] A. Brown and B. Mathews, "Indoor Navigation using a Software Defined Radio," Proceedings of SDR '08 Technical Conference, Washington, DC, Oct. 2008