



AUDITOR – GA 687367

Advanced Multi-Constellation EGNSS Augmentation and Monitoring Network and its Application in Precision Agriculture

D2.1 Version 1.0

Architecture definition

Contractual Date of Delivery:	M5
Actual Date of Delivery:	31/05/2016
Editor:	<i>Jacobo Dominguez (ACORDE)</i>
Author(s):	<i>Esther López, Jacobo Domínguez, David Abia (ACORDE); Carles Fernandez-Prades, Marc Majoral, Javier Arribas (CTTC); Alberto García Rigo, Manuel Hernández-Pajares (UPC);</i>
Work package:	<i>WP2 - Receiver architecture definition</i>
Security:	CO
Nature:	R
Version:	1.0
Total number of pages:	28

Abstract:

This deliverable establishes the architecture that will be the foundation of the AUDITOR project, the internal details of the subsystems presented here will be further specified in D2.2. The core hardware elements of the proposed architecture for a flexible GNSS receiver are a configurable GNSS RF front-end and a mixed FPGA/microcontroller flexible processing unit. The software modules that will implement the GNSS augmentation algorithms and provide the precise positioning services are also introduced in this document. These low level software modules will be embedded in the GNSS receiver to allow an efficient real time data processing of the GNSSs data stream and extended by a cloud services to obtain correction data and provide additional features via user-oriented applications.



Document Control

Version	Details of Change	Author	Approved	Date
0.1	Document Created	JD		
1.0	First complete version of the document	°	EL	31/05/2016

Executive Summary

GNSS receivers integrate more and more features becoming complex processing units that offer high precision services using multiple GNSS systems and hardware channels. By analysing WP1 deliverables, which established the state of the art, requirements and common unit tests for modern commercial GNSS receivers, this deliverable presents the overall architecture that will be the base for the AUDITOR GNSS receiver.

The proposed architecture integrates a dual system GNSS front-end (Galileo/GPS) that features one fixed and one configurable channel. This front-end is tightly integrated with a fully customizable processing unit based on a powerful ARM/FPGA processor where customized GNSS augmentation services will be implemented.

The AUDITOR GNSS receiver architecture follows an open approach where open source libraries, some maintained by AUDITOR partners as the GNSS-SDR, are integrated while custom algorithms will be developed and integrated to support real time precise positioning services.

Moreover, the proposed GNSS receiver architecture includes integration with cloud services to obtain GNSS reference correction data from external sources and to provide user services via web applications.

Authors

Partner	Name	e-mail
ACORDE	Esther López	e-mail: esther.lopez@acorde.com
	Jacobo Domínguez	e-mail: jacobo.dominguez@acorde.com
	David Abia	e-mail: david.abia@acorde.com
CTTC	Carles Fernández-Prades	e-mail: carles.fernandez@cttc.es
	Marc Majoral	e-mail: marc.majoral@cttc.es
	Javier Arribas	e-mail: javier.arribas@cttc.es
UPC	Alberto García Rigo	e-mail: alberto.garcia.rigo@upc.edu
	Manuel Hernández-Pajares	e-mail: manuel.hernandez@upc.edu

Table of Contents

Document Control	2
Executive Summary	3
Authors	4
Table of Contents	5
List of tables	5
List of Figures.....	5
List of Acronyms and Abbreviations.....	7
1. Overall System Architecture	9
2. GNSS Requirements review	12
3. GNSS Front-End Module	14
4. Digital Processing Platform	16
4.1 High Accuracy Software Module.....	20
4.2 Network Software Module	21
4.2.1 Required messages and information	21
4.2.2 WARKT CPF real time implementation	22
4.2.3 Client PVT Solver	24
4.3 User Interfaces.....	24
4.3.1 Internal Interfaces.....	24
4.3.2 External interfaces	25
5. Power Module	26
6. Conclusion.....	27
7. References	28

List of tables

Table 2.1: GNSS Requirements.....	12
Table 4.1: Features of some models of Zynq devices	18
Table 4.2: Coarse estimation of the number of required DSP slices. N is the number of correlators per satellite signal.....	20
Table 4.3: Coarse estimation of the number of required DSP slices for different GNSS receiver configurations.....	20

List of Figures

Figure 1.1: System Architecture	9
Figure 3.1: Front-End module architecture.....	14
Figure 4.1: Zynq-7000 All Programmable SoC Overview (from [1]).....	17
Figure 4.2: GNSS signal acquisition block in the Programmable Logic.....	19

Figure 4.3: GNSS signal tracking block in the Programmable Logic19

Figure 4.4: WARKT Algorithm layout for the roving user.....22

Figure 4.5: WARTK-RT software data flow diagram23

List of Acronyms and Abbreviations

Term	Description
ACP	Accelerator Coherency Port
ADC	Analog-to-Digital Conversion
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
APU	Application Processor Unit
ASSP	Application Specific Standard Product
AXI	Advanced eXtensible Interface
CAN	Controller Area Network
CLB	Configurable Logic Block
CPU	Central Processing Unit
DAP	Debug Access Port
DDR	Double Data Rate
DevC	Device Configuration interface
DMA	Direct Memory Access
DMIPS	Dhrystone Million Instructions Per Second
DSP	Digital Signal Processor
ECC	Error Correction Checking
EHCI	Enhanced Host Controller Interface
EMIO	Extendable Multiplexed Input / Output
FE	Front End
FPGA	Field Programmable Gate Array
GIC	General interrupt controller
GMII	Gigabit Media-Independent Interface
GNSS	Global Navigation Satellite System
IOP	Input / Output Peripherals
IP	Intellectual Property
IRQ	Interrupt ReQuest
LPDDR	Low Power Double Data Rate

Term	Description
LUT	Look-Up Table
MAC	Media Access Control
MIO	Multiuse Input / Output
MMU	Memory Management Unit
OCM	On-Chip Memory
OTG	On-The-Go
PCAP	Processor Configuration Access Port
PL	Programmable Logic
PS	Processing System
RAM	Random Access Memory
RGMII	Reduced Gigabit Media-Independent Interface
ROM	Read Only Memory
SGMII	Serial Gigabit Media-Independent Interface
SoC	System-on-Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SWDT	System Watch Dog Timer
TTC	Triple Timers / Counters
UART	Universal Asynchronous Receiver-Transmitter
ULPI	UTMI+ Low Pin Interface
UTMI	USB 2.0 Transceiver Macrocell Interface
VFPU	Vector Floating Point Unit
WDT	Watch Dog Timers

1. Overall System Architecture

In this section an overview of the system architecture is presented which includes all internal and external subsystems devoted to implement the AUDITOR systems. The architecture designed satisfies the main GNSS requirements identified in **WP1** that are revisited in section 2. The specific details of each subsystem will be further extended in **D2.2 Subsystem specification**.

Figure 1.1 depicts the overall architecture of the AUDITOR systems and its related subsystems; this figure also represents the relations of the different WPs with the main hardware elements, software modules and logical subsystems.

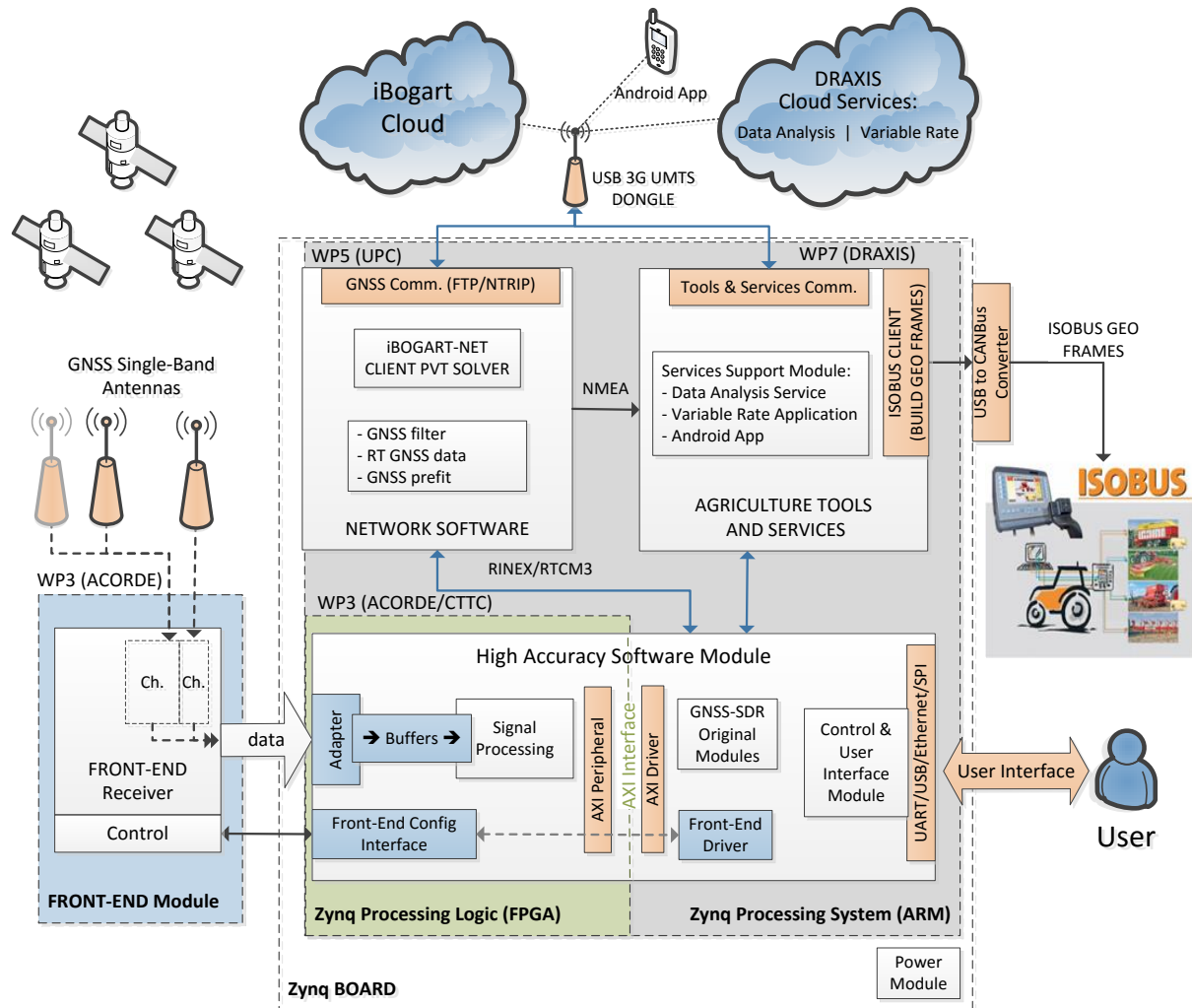


Figure 1.1: System Architecture

There are two main hardware elements that embed different software processes and algorithms:

- Front-end Module.
- Zynq Board.

Front-End (FE) Module (related to WP2): This hardware/software module implements a configurable RF dual-band multi-constellation receiver which acquires the GNSS signals.

The FE embeds all analogue and digital hardware required to convert the RF signal to digital samples. It provides a data output interface with I/Q samples, clock and a management interface (UART/SPI) to configure its main parameters.

Zynq BOARD: The Zynq Board is the main embedded processing element of the AUDITOR system. The Zynq platform has been selected as the base for this subsystem due to its adequate performance, flexibility and convenient FPGA/ARM mixed features. The Zynq Board is shown in Figure 1.1 divided into its two main interconnected hardware subsystems the **Zynq Processing Logic (PL or FPGA)** and the **Zynq Processing System (PS or ARM)**. In the AUDITOR GNSS receiver these two hardware elements support three logical subsystems:

- High Accuracy Software Module
- Network Software Module
- Agriculture Tools and Services

The **High Accuracy Software Module** (related to WP3) contains the lower level processing algorithms which are implemented partially into the FPGA and the ARM Zynq hardware.

The Zynq Processing Logic contains customized intellectual property (IP) FPGA cores. These cores implement high performance real-time processes that perform the signal acquisition and conversion of the GNSS signals. These cores are customized for the AUDITOR systems using Zynq tools in VHDL language. The main features of the IP cores that will be integrated in this module are:

- GNSS signal sampling and data buffering,
- low level data pre-processing adapting the data stream throughput,
- configuration interface for the front-end parameters,
- GNSS data interface used by the ARM processing system.

The FE module will be implemented as a standalone module that is connected to the Zynq Processing Logic which provides the adequate computational power to perform customized post-processing of the I/Q samples in real-time conditions. To achieve this performance while maintaining a high configurable GNSS receiver the FPGA will implement a data-adapter/low-level-buffer configurable by ARM drivers that will perform power efficient pre-processing to the FE data throughput.

The Zynq Processing System implements higher level algorithms, functions and services of the AUDITOR system. These modules are integrated into an embedded Linux OS with third-party libraries and drivers. In this subsystem software drivers to configure the FE and the IP cores through the AXI interface will be provided. The GNSS-SDR [1] library that is supported by some AUDITOR partners will also be integrated and customized into this subsystem. Additional user interface are planned to be integrated into this module to ease the debugging and testing of the High Accuracy Software module through the wired external interfaces (SPI/UART/USB/Ethernet) and allow the future integration with external subsystems.

The **Network Software** (related to WP5) subsystem implements the local algorithm and remote communication to implement the ionospheric Boosted GNSS Applications in Real-Time (iBOGART) based on the iBOGART-net-MODEL prototype (WP4). The main features of the Network Software are:

- retrieve the real-time RT-IGS data providing a serial pipe of data streams,
- perform pre-processing of the measurements and run the hybrid ionospheric-geodetic model,
- implement a Kalman filter estimation based on the previous model,
- support the precise GNSS navigation by merging the GNSS receiver data and the precise corrections distributed to users

The **Agriculture Tools and Services** (related to WP7) subsystem provide the basis to implement higher level services for agriculture end-users. The main services provided by this subsystem are:

- precise navigation of autonomous agriculture machinery,
- data analysis service of the positioning data,
- a variable rate application based on the data analysis services.

These services will be provided via web/android applications that will be implemented as cloud services. These cloud services communicate with the embedded Agriculture Tools and Services processes depicted in Figure 1.1 to retrieve positioning data information, provide remote data and implement the ISOBUS interface to communicate with the local agriculture machinery.

In the next sections a summary of the previous embedded subsystems will be presented except for the Agriculture Tools and Services. This service is fully based on cloud applications and is expected that a minimal implementation is needed to be performed in the GNSS receiver.

2. GNSS Requirements review

This section reviews the GNSS Requirements defined in [2] Table 3.2 completing these requirements with comments related to the proposed architecture and the hardware/software limitations.

Table 2.1: GNSS Requirements

CATEGORY	DESCRIPTION	COMMENTS
Usage cost	<ol style="list-style-type: none"> 1. Low cost 2. Save time and money and reduce fatigue 3. Increase profit margins 	Based on standard Zynq platform that will ensure low cost hardware and open source libraries. Low costs also related to the setup and exploitation of the GNSS receiver based on embedded system and cloud scalable services.
Connectivity and data sharing	<ol style="list-style-type: none"> 1. Compatibility and connectivity with: <ol style="list-style-type: none"> a. soil mapping sensors b. yield mapping sensors c. yield monitor displays d. NDVI ground sensors e. LiDAR sensors f. light bar driver assistance systems g. auto-steer systems h. VRA machineries i. UAV's j. UGV's 	<p>Connectivity with external sensors and subsystems will be provided by the ISOBUS protocol and the support of the NMEA 0183 standard.</p> <p>ISOBUS is based on ISO 11783 - "Tractors and machinery for agriculture and forestry – Serial control and communications data network" which ensures a wide compatibility.</p> <p>ISOBUS together with NMEA 0183 cover most of the common systems, sensors and displays.</p> <p>Future integrations could be extended using USB/RS232 interfaces with custom protocols.</p>
	<ol style="list-style-type: none"> 2. Real time data transferring in frequency of 20Hz 	The GNSS receiver will support data rates of 1Hz. The feasibility to support 2-5-10-20Hz will be evaluated during testing phases as big transfer rates are quite important in case of UGV's and auto-steering systems.
	<ol style="list-style-type: none"> 3. Real time data transferring to ground control stations 	Data transferring to user ground control station possible via 3G radio link. However AUDITOR system follows a cloud service approach.
	<ol style="list-style-type: none"> 4. Real time correction to compensate for the combining delay ("positioning offset") 	We will assess the need for transmitting temporal derivatives of the parameters' coefficients to allow for a simple method of compensation.
	<ol style="list-style-type: none"> 5. ISOBUS standard ISO 11783 support 	Supported by a USB to CAN converter.
	<ol style="list-style-type: none"> 6. NMEA 0183 support 	Supported by a USB to RS422 converter.
	<ol style="list-style-type: none"> 7. RS232 and USB connectors 	Multiple interfaces available in the selected Zynq platform
	<ol style="list-style-type: none"> 8. Raw data exporting capabilities 	Provided initially for debugging and testing in the High Accuracy Software Module the possibility to export it locally or to cloud services will be assessed.
	<ol style="list-style-type: none"> 9. Digital Elevation Model (DEM) creation 	We will assess the need of this application that could be provided remotely by cloud services.
	<ol style="list-style-type: none"> 10. Data extraction at the most common GIS protocols (KML, SHP, etc.) 	The basic set of common GIS protocols will be evaluated and then integrated to provide positioning information via standard formats.

CATEGORY	DESCRIPTION	COMMENTS
	11. Capability of shifting raw data files time to compensate for the combining delay ("positioning offset")	We will assess the need for transmitting temporal derivatives of the parameters' coefficients to allow for a simple method of compensation.
Reliability	1. High precision <ul style="list-style-type: none"> a. Low Vertical error (less than 2cm) b. Low Horizontal error (less than 1cm) c. Accurate travel speed calculation ($< \pm 3\%$) 	Vertical error of up to few decimetres and Horizontal error of several centimetres to one decimetre ¹ . We could assess deriving an accurate travel speed in case the receiver makes the Doppler observables available.
	2. Widely available	A high enough Bandwidth to allocate the message being transmitted is envisaged. Multiple constellations (Galileo/GPS), multiple channels ensure higher availability.
	3. Positioning recording and projection	We will assess the need of this application that could be provided remotely by cloud services.
	4. Data recording in frequency up to 20Hz	The GNSS receiver will support data recording rates of 1Hz. The feasibility to support 2-5-10-20Hz will be evaluated during testing phases.
	5. IMU support	We will assess the use of commercial external IMUs. The communicate interface with commercial IMUs could be achievable using the available RS232/USB. However the specific protocol to consume its data depends heavily on the IMU manufacturer.
	6. Dead Reckoning capabilities	Kalman filters estimations implemented in WP5 are based on GNSS data and initially not oriented to consume IMU measures to implement dead reckoning capabilities. Depending on the success of requirement "5. IMU support" the feasibility to integrate its measures to support dead reckoning conditions will be evaluated.
	7. IP67 protection	IP65
	8. Ability to work through low visibility field conditions	Multiple constellations (Galileo/GPS), multiple channels and positioning of antennas ensure higher availability even in low visibility conditions
	9. Easily connectable and flexible	Based on open architecture and common well-known interfaces and operating systems.

¹ Note that in a real scenario like in AUDITOR, the accuracy level may not reach the obtained values in laboratory conditions.

3. GNSS Front-End Module

The Front-End module (FE) is composed of a RF core that embeds a dual channel GNSS receiver and additional digital control logic to Monitor&Control (M&C) the main receiver parameters.

An overview of the FE module GNSS is show in Figure 3.1.

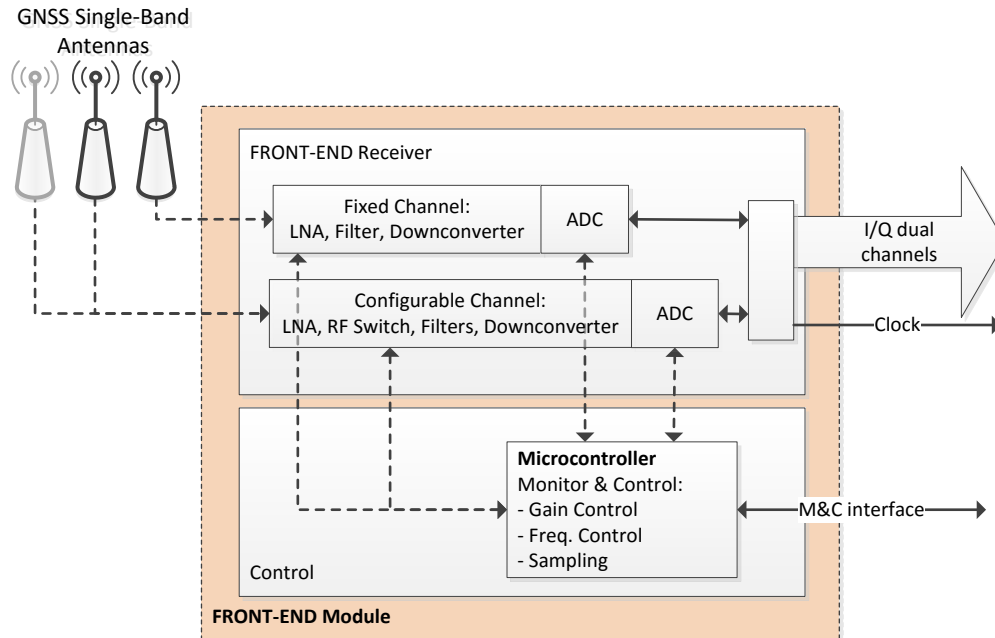


Figure 3.1: Front-End module architecture

The proposed FE GNSS architecture includes a dual-band receiver divided into a fixed channel and a configurable one:

- Fixed: Galileo (E1, 1559-1591MHz)/GPS (L1, 1563-1587 MHz) → ~32 MHz total bandwidth
- Configurable between:
 - GPS (L2, 1215-1237 MHz) → ~24 MHz bandwidth
 - Galileo(E5a 1164-1191 MHz)/GPS (L5, 1164-1191 MHz) → ~32MHz bandwidth

The GNSS signals are captured via single band antennas. The use of single-band antennas versus dual or triple band antennas adds more flexibility to the system and could support advanced algorithms based on phase difference by positioning the antennas in different locations if needed. This allows selecting and replacing these antennas independently which widens the selection based on market available models. Market available single band antennas are more and better cost-optimized than for dual or triple band antennas models, usually more expensive and with more proprietary features. Implementing independent channels simplifies the FE complexity and reduces inter-bands interferences.

The fixed channel implements an E1/L1 optimized downconverter and hardware sampler to offer a continuous stream of I/Q samples.

The configurable channel provides extra adaptability to the GNSS receiver allowing selecting between the L2 band and the more convenient dual constellation E5a/L5. This feature is implemented using RF switches and dynamic reconfiguration of the internal mixer frequencies.

Proper RF shield of the FE is needed to avoid interference between the RF channels, the internal downconverters and the I/Q data sampling data stream.

Additionally to the previous RF core elements a support microcontroller will be integrated to perform the associated M&C tasks for the FE module. This microcontroller is externally controlled via a standard serial interface that is connected to the Zynq BOARD and will also allow performing standalone tests to assess the FE functionalities. The main M&C tasks are:

- Reporting and adjusting automatic gain control parameters for each band.
- Configuration of working parameters and LO frequencies used in the downconverters.
- Selection of different ADC parameters (sampling frequency, output format) to adapt the I/Q data throughput to the GNSS data post processing.

In order to provide the maximum flexibility, the ADCs will utilize all bits in their scale, but a custom core in the Zynq Board will pre-process them and will pass only the desired bits (configurable) to the High Accuracy Processing Module.

The FE provides the I/Q data for both GNSS channels simultaneously via a general purpose wide bus connector and also exposes its main reference clock through this interface.

4. Digital Processing Platform

AUDITOR's digital processing platform is in charge of executing:

1. The GNSS software receiver
2. The high accuracy software module
3. Network Software module
4. Agriculture Tools and Services (Cloud services)

GNSS baseband processing requires a high computational load. Even when executed in modern, powerful computers, the number of satellite channels that the receiver is able to process in real-time is limited, especially for BOC-based signals. The solution proposed in AUDITOR is based on a System-on-Chip architecture that combines power-efficient, all-purpose **ARM®** processors and powerful **FPGAs** in a single chip. This allows executing the GNSS software receiver and other software modules in the ARM® processors, while offloading the most demanding computations to the FPGA.

The **Zynq®-7000 family** [6] is based on the Xilinx All Programmable SoC architecture. These products integrate a feature-rich dual-core ARM® Cortex™-A9 based processing system (PS) and 28 nm Xilinx programmable logic (PL) in a single device. The ARM Cortex-A9 CPUs are the heart of the PS and also include on-chip memory, external memory interfaces, and a rich set of peripheral connectivity interfaces. This innovative architecture allows for the integration of the software programmability of an ARM®-based processor with the hardware programmability of an FPGA, enabling hardware acceleration while integrating CPU, DSP, ASSP, and mixed signal functionality on a single device.

The Zynq-7000 architecture enables implementation of **custom logic in the PL and custom software in the PS**. It allows for the realization of unique and differentiated system functions. The integration of the PS with the PL allows levels of performance that two-chip solutions (e.g., an ASSP with an FPGA) cannot match due to their limited I/O bandwidth, latency, and power budgets.

Xilinx offers a large number of soft IP for the Zynq-7000 family. Stand-alone and Linux device drivers are available for the peripherals in the PS and the PL. The Vivado® Design Suite development environment enables a rapid product development for software, hardware, and systems engineers. Adoption of the ARM-based PS also brings a broad range of third-party tools and IP providers in combination with Xilinx's existing PL ecosystem. The inclusion of an application processor enables **high-level operating system support**, e.g., Linux. Other standard operating systems used with the Cortex-A9 processor are also available for the Zynq-7000 family.

The PS and the PL are on separate power domains, enabling the user of these devices to power down the PL for power management if required. The processors in the PS always boot first, allowing a software centric approach for PL configuration. PL configuration is managed by software running on the CPU, so it boots similar to an ASSP. Figure 4.1 shows the block diagram of a Zynq-7000 device.

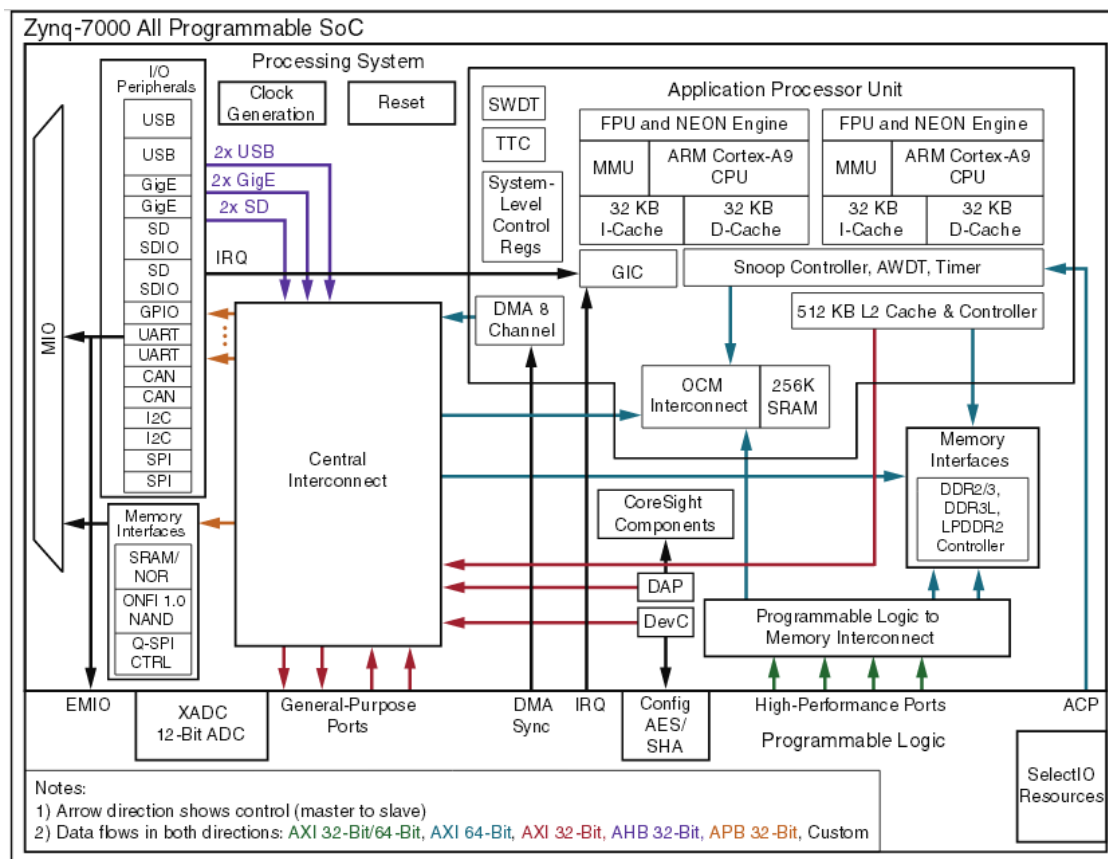


Figure 4.1: Zynq-7000 All Programmable SoC Overview (from [1])

Processing System

As shown in Figure 4.1, the Processing System comprises four major blocks: *i*) Application Processor Unit (APU), *ii*) memory interfaces, *iii*) I/O peripherals (IOP), and *iv*) interconnect. The following list summarizes its main features which will be further detailed in **Deliverable 2.2**:

- Dual-core ARM® Cortex™-A9 Based Application Processor Unit (APU),
- Multiple Level Caches
- On-Chip Memory boot ROM and RAM
- External Memory Interfaces 16/32bits
- 8-Channel DMA Controller
- Interconnect of PS and PL (AXI based)

Programmable Logic

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable". FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. The most common FPGA architecture consists of an array of logic blocks (called Configurable Logic Block, CLB), I/O pads, and routing channels.

In the specific case of the Zynq-7000 All Programmable SoCs, Lookup Tables (LUTs) can be configured as either one 6-input LUT (64-bit ROMs) with one output, or as two 5-input LUTs (32-bit ROMs) with separate outputs but common addresses or logic inputs. Each LUT output can optionally be registered in a flip-flop. Four such LUTs and their eight flip-flops as well as multiplexers and arithmetic carry logic form a slice, and two slices form a configurable logic block (CLB). Four of the eight flip-flops per slice (one flip-flop per LUT) can optionally be configured as latches.

The following list summarizes the main features of Zynq-7000's Programmable Logic which will be further detailed in **Deliverable 2.2**:

- Configurable Logic Blocks (CLB)
- 36 Kb Block RAM
- DSP Blocks
- Programmable I/O Blocks
- JTAG Boundary-Scan
- PCI Express Block
- Serial Transceivers
- Two 12-Bit Analog-to-Digital Converters

Table 4.1 summarizes the main features of some models of Zynq devices that have an associated design boards and Evaluation Kit including all the basic components to enable a complete embedded processing platform.

Feature	Model: Z-7020 (Ev. kit: Zedboard)	Model: Z-7045 (Ev. kit: ZC706)	Model: Z-7100 (Ev. kit: MINI ITX)
Logic Cells (K)	85	350	444
Block RAM (Mb)	4.9	19.1	26.5
DSP Slices	220	900	2020
Maximum I/O pins	128	212	250

Table 4.1: Features of some models of Zynq devices

AUDITOR Design

Details on the signal Acquisition and Tracking blocks, the most computationally demanding operations in a GNSS receiver, are provided in Figure 4.2 and Figure 4.3.

SIGNAL ACQUISITION HW ACCELERATOR IP

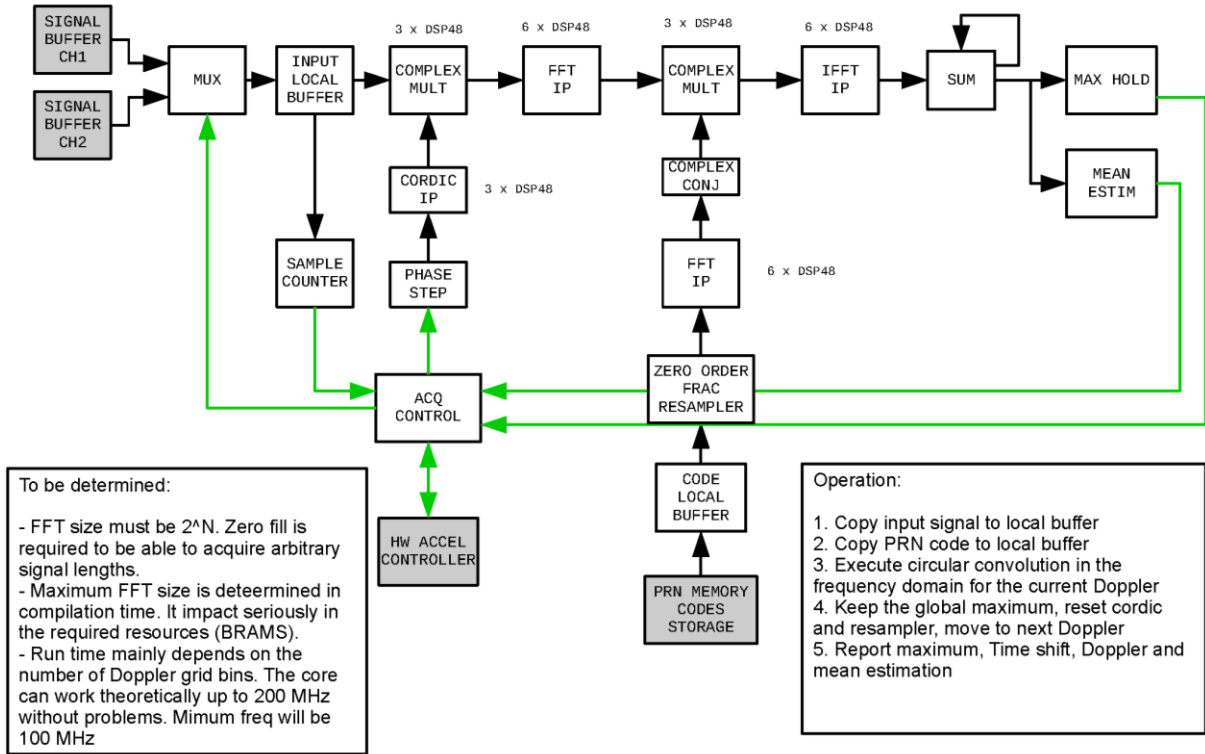


Figure 4.2: GNSS signal acquisition block in the Programmable Logic

SIGNAL TRACKING HW ACCELERATOR IP

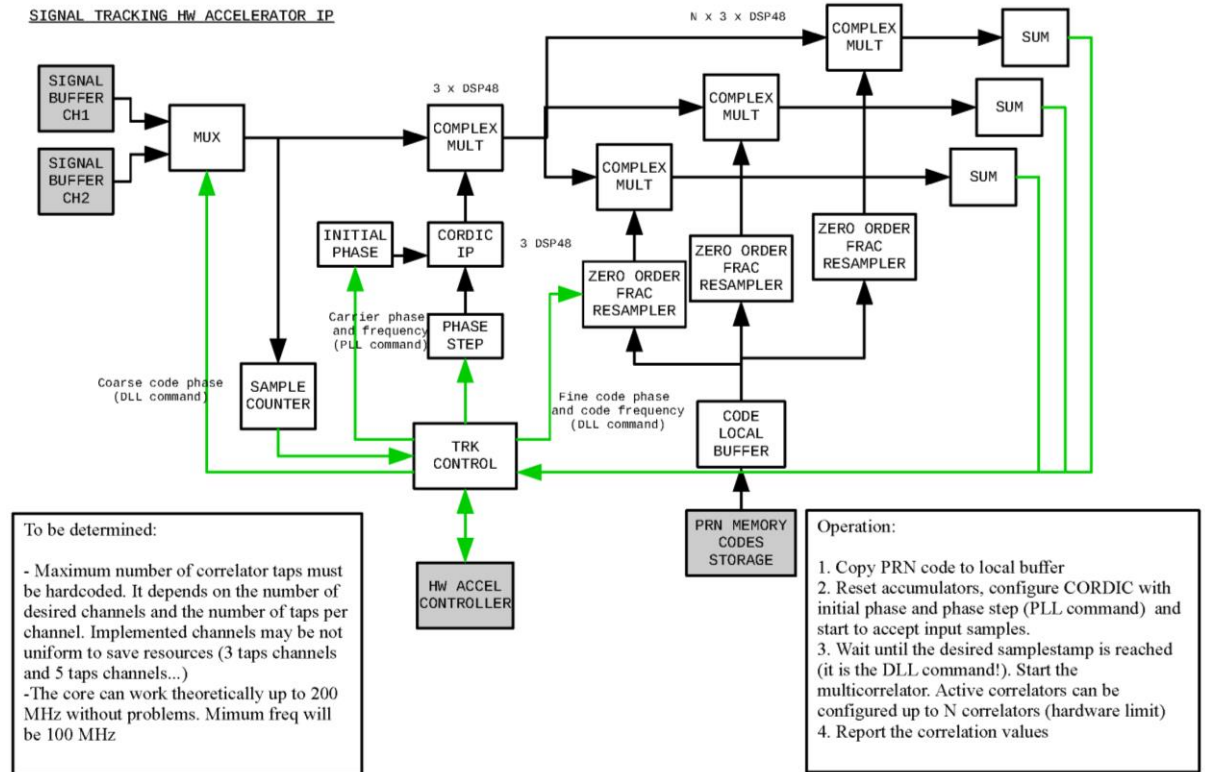


Figure 4.3: GNSS signal tracking block in the Programmable Logic

From Figure 4.2 and Figure 4.3 it can be obtained a first estimation on the number of required DSP slices. Results are summarized in Table 4.2.

	Acquisition	Tracking	Total
DSP slices (per satellite/signal)	27	$6 + 3 \cdot N$	$33 + 3 \cdot N$

Table 4.2: Coarse estimation of the number of required DSP slices. N is the number of correlators per satellite signal

Table 4.2 allows a preliminary estimation of the total number of required DSP slices for different GNSS receiver configurations. Results are shown in Table 4.3.

GNSS receiver configuration	Required number of DSP slices (approx.)
12 GPS L1 channels (N=3 correlators per channel)	504
12 Galileo E1 channels (N=5 correlators per Galileo channel)	576
12 GPS L1 channels and 12 GPS L2C channels (N=3 correlators per channel)	1008
12 GPS L1 and 12 Galileo E1 channels (N=3 correlators per GPS channel and N=5 per Galileo channel)	1080
12 Galileo E1 channels and 12 Galileo E5a/b channels (N=5 correlators per channel)	1152

Table 4.3: Coarse estimation of the number of required DSP slices for different GNSS receiver configurations

The final selection of the Evaluation Board will be performed in Deliverable D2.2.

4.1 High Accuracy Software Module

GNSS signal processing takes place in the High Accuracy Software Module. Depending on the computational effort and flexibility required, some elements will run on the **Zynq Processing Logic** (as VHDL cores) while others will do on the **Zynq Processing System** (as Linux software).

Thus, the PL contains the following main cores:

- Signal acquisition IP
- Signal tracking IP
- Additionally, some logic required to interface with the FE are required, such as:
 - adapter (ADC/VHDL driver): to provide the adequate bit/sample relation to the signal processing module,
 - Intermediate buffer: to support real-time continuous exchange of data,

- FE config interface: based on a UART port to M&C the FE.

Whereas the software of the PS will consist on:

- Drivers for interfacing with the PL (DMA, configuration registers)
- HW accelerators for acquisition and tracking
- GNSS-SDR software
- RTK/PPP/DGPS engine
- User interfaces
- Driver for FE management: to control the FE through the FE config interface and the adapter (ADC/VHDL driver) embedded into the PL.

4.2 Network Software Module

This section describes the high-level architecture of the Network Software Module. In brief, the WARTK user algorithm is based on the fast ionospheric-free ambiguity constraint thanks to the availability of precise ionospheric corrections from the **iBOGART cloud (Central Processing Facility** known as the CPF, running the hybrid ionospheric-geodetic model) and the smoothed double-differenced Melbourne-Wübbenna combination.

For the description of the iBOGART Central Processing Facility at the **iBOGART Cloud**, please refer to Section 4.2.2.

4.2.1 Required messages and information

A WARTK rover must gather messages and information from two different sources (which will feed the corresponding navigation filter and ambiguity fixing, see Figure 4.4) in order to compute a precise positioning using WARTK approach:

- **Roving Receiver Observables**

The observables shall be acquired from the rover dual-frequency Software Receiver. This consists of the dual-frequency pseudoranges and carrier-phases for each satellite in view and for each observation epoch. In AUDITOR, data shall be made available in real time at 1 Hz and the suitability to use higher sampling rates will be assessed during the implementation phase. As it is shown in Figure 1.1, the observables are to be made available by the High Accuracy Software Module in RINEX format or RTCM format.

In addition, SRMTID index reflecting the intensity of MSTID activity will be computed in real time by the iBOGART-user-auto-SRMTID module and it will be taken into account in the user filter in order to autonomously down-weight the more affected satellites, if any. In order to keep track on ionospheric disturbances, the index shall be made available each minute.

- **Network Data and Corrections**
 - **Reference Station Observables**, same measurements as in the rover but from a reference station receiver. These measurements are necessary in order to compute the satellite clocks and to constrain the values of double differenced ambiguities (and eventually to fix them to an integer number of cycles).

- **Precise Satellite Orbits (and eventually Clocks)** are used in the CPF computations, so the same set of orbits (and eventually clocks) should be used in the user segment. Currently WARTK uses IGS Ultra-rapid predicted orbits and clocks, which are published in real-time through the Internet and offer an accuracy of 5 cm and 3 ns (for the predicted half) respectively, which is quite better than the 100 cm and 5 ns offered by the GPS broadcast signal.
- **STECs in the reference stations** (or an equivalent adjustment) that can be used by the rover to get an interpolated value to its position. This value is broadcasted jointly with a parameter of quality, in such a way that the user is able to downweight the worst values.

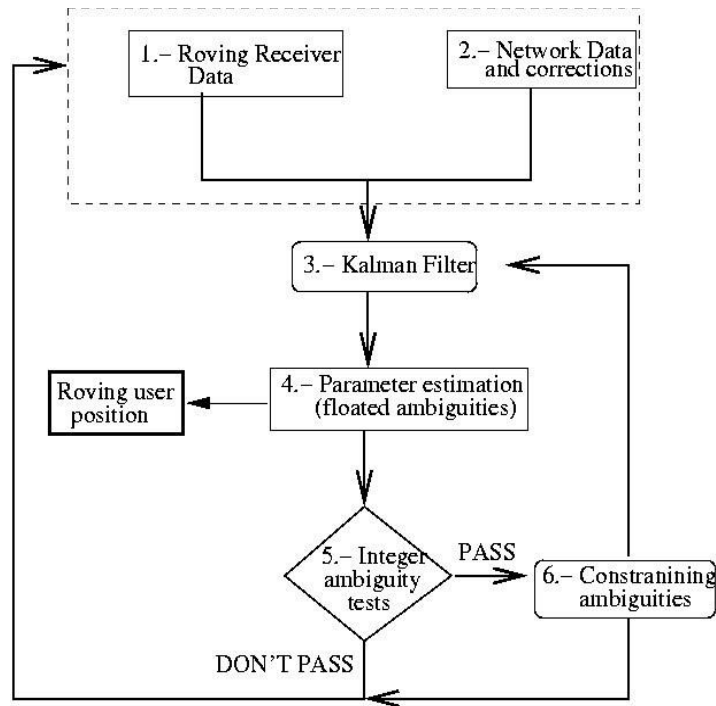


Figure 4.4: WARTK Algorithm layout for the roving user.

In Figure 4.4, Roving Receiver Data corresponds to the AUDITOR rover station observations, directly provided by the High Accuracy Software Module. And Network Data and Corrections have information from four different sources (observations, orbits and clocks, ionospheric corrections and SRMTID index). Therefore, three different messages should be transmitted to the rover from the iBOGART Central Processing Facility (in the iBOGART Cloud, as depicted in Figure 1.1). The first one is the “OB1” message (observations), which comes from the reference station, the second one is the “S3C” message (orbits and clocks), which is gathered from IGS servers, and “DSM”, the third one, contains ionospheric corrections and should be continuously generated and broadcasted in real-time by the WARTK CPF. A brief introduction of the implementation of the WARTK CPF is presented next.

4.2.2 WARTK CPF real time implementation

The so-called Wide Area Real-Time Kinematic Central Processing Facility (WARTK-RT CPF) is able to run in real-time conditions thanks to the retrieval of NTRIP GNSS datastreams (these can be directly obtained using, for instance, BKG’s BNC) and the usage of Linux pipes and WARTK associated programs supporting real time processing.

From the GNSS datastreams, the **CPF can obtain a continuous flow of GPS observables and thus generate WARTK's OB1 messages in real-time.** Apart from that, the CPF downloads updated RINEX and SINEX files from a set of world-wide (or European) receivers, it updates precise predicted orbit sets every 6 hours and continuously generates the WARTK output message, which includes real-time plots, among other features. In Figure 4.5, a diagram on the WARTK software structure is shown, which details are to be provided in deliverable D2.2.

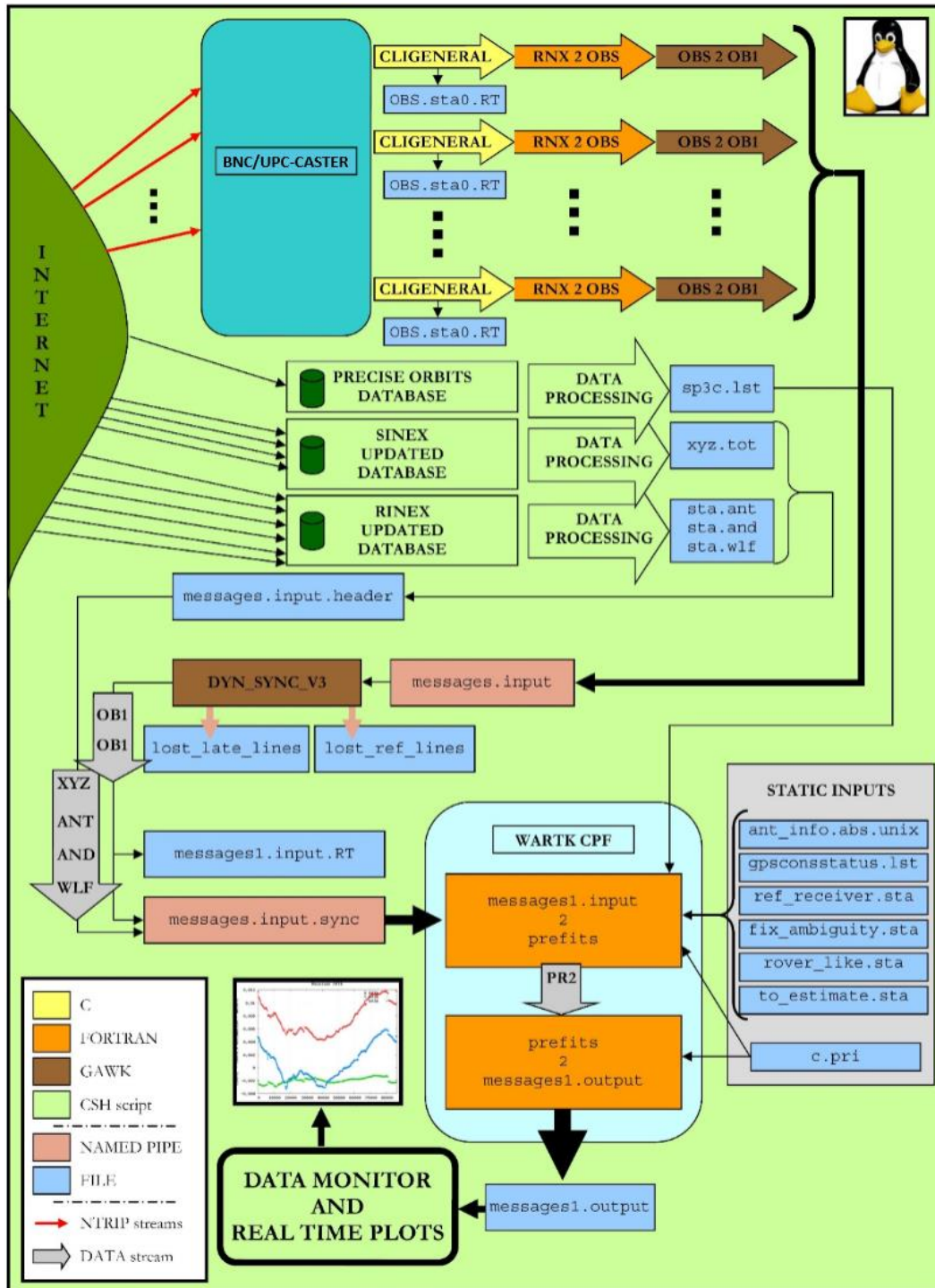


Figure 4.5: WARTK-RT software data flow diagram

More in detail, the whole CPF SW package consists of six main CSH scripts, some secondary C and FORTRAN programs (including the WARTK core) and many AWK scripts. It should be installed in a Linux compatible machine, and requires some extra software to be installed. Specifically it needs the

following OpenSource applications: wget, gawk, gnuplot, GMT, GraphicsMagick command-line utilities, and Randomize Lines utility. In addition, it has been designed to be 100% portable within different Linux machines.

More details about the CPF real time implementation and its messages format (“OB1”, “S3C”, “DSM”) will be included in deliverable D2.2.

4.2.3 Client PVT Solver

The preferred baseline to derive the positioning solution is **RTKLIB**. This will be the preferred solution, once its flexibility to support WARTK is double-checked, since the receiver already supports RTKLIB execution. In this way, it will be analysed that all WARTK corrections can be taken into account in the RTKLIB processing flow. For such a purpose, it is expected to modify the source code of RTKLIB to support correction messages beyond the ones already implemented in the current specification.

In case this is finally not possible, the UPC navigation filter will be used instead as back-up option. Specially in such a case, it will be necessary that the embedded Linux supports the execution of CSH/FORTRAN/AWK scripts.

4.3 User Interfaces

In this section the main internal and external interfaces for the AUDITOR system are summarized.

4.3.1 Internal Interfaces

RF front-end – digital processing platform interface

The digital samples at the output of the ADCs will be transferred to the digital processing platform through a Pmod interface. Pmod (or Peripheral Module) interface is an open standard defined by Digilent Inc in the Digilent Pmod™ Interface Specification [2] for peripherals used with FPGAs or microcontrollers. There are six-pin and twelve-pin versions of the interface defined. The six-pin version provides four digital I/O signal pins, one power pin and one ground pin. The twelve-pin version provides eight I/O signal pins, two power pins and two ground pins. In general, Pmod peripheral modules can plug directly into connectors on the host controller board, or be connected to the controller board via six-pin or twelve-pin cables. Pmod peripheral modules are powered by the host via the interface’s power and ground pins.

The Pmod interface is not intended for high frequency operation over long distances, however, using RJ45 connectors and twisted pair Ethernet cable, signals have been sent reliably at 24 MHz and distances of up to 4 meters [2].

Programmable Logic – Processing System interface

Xilinx® adopted the Advanced eXtensible Interface (AXI) protocol for Intellectual Property (IP) cores for Zynq devices. AXI is part of ARM AMBA, a family of micro controller buses first introduced in 1996. AMBA is an open standard for the connection and management of functional blocks in a System-on-Chip. The first version of AXI was first included in AMBA 3.0, released in 2003. AMBA 4.0, released in 2010, includes the second version of AXI, AXI4 [3].

The **AXI specifications** describe an interface between a single AXI master and a single AXI slave, representing IP cores that exchange information with each other. Memory mapped AXI masters and

slaves can be connected together using a structure called an Interconnect block. The Xilinx AXI Interconnect IP contains AXI-compliant master and slave interfaces, and can be used to route transactions between one or more AXI masters and slaves.

There are three types of ports which can be used to transfer data between the FPGA (Programmable Logic) and the CPU (Processing System):

1. **General Purpose AXI ports:** 2x Master (from CPU to FPGA) and 2x Slave port (from FPGA to CPU). These ports are connected to the central interconnect of the processing system and can be used to transfer data to/from DDR memory or on-chip memory (OCM).
2. **High Performance AXI ports:** 4x Slave port (from FPGA to CPU) provide high-bandwidth access to DDR or OCM
3. **ACP (Accelerator Coherency Port):** Slave port (from FPGA to CPU) high-throughput port connected directly to the snoop control unit (SCU). The SCU maintains cache coherency (omits the need for cache flush/invalidates).

4.3.2 External interfaces

Devices belonging to the Zynq family offer a wide range of interfaces (see Figure 4.1) that will be used in AUDITOR project to connect to the GNSS front-end, to external ISOBUS machinery and access TCP/IP networks either via a wired connection or a Wi-Fi/3G via USB dongle:

- Two 10/100/1000 tri-speed Ethernet MAC peripherals with IEEE Std 802.3 and IEEE Std 1588 revision 2.0 support
- Two USB 2.0 OTG peripherals, each supporting up to 12 Endpoints
- Two full CAN 2.0B compliant CAN bus interfaces
- CAN 2.0-A and CAN 2.0-B and ISO 118981-1 standard compliant
- Two full-duplex SPI ports with three peripheral chip selects
- Two high-speed UARTs (up to 1 Mb/s)
- Two master and slave I2C interfaces
- GPIO with four 32-bit banks, of which up to 54 bits can be used with the PS I/O (one bank of 32b and one bank of 22b) and up to 64 bits (up to two banks of 32b) connected to the Programmable Logic
- Up to 54 flexible multiplexed I/O (MIO) for peripheral pin assignments

However, this list only refers to the electrical specifications at the device's output pins, but not all of those interfaces are then exposed by the evaluation boards, where the physical connectors are located. For instance, the Zynq Mini-ITX Development Kit utilizes Ethernet, USB 2.0 (Host mode only) and USB UART physical layer transceivers for communication purposes. Network access is provided by a single 10/100/1000 Mb/s Ethernet PHY device, which is connected to the Zynq-7000 AP SoC via a standard RGMII interface. The PHY device connects to the outside world with a standard RJ45 connector.

5. Power Module

The Zynq platform is widely available for development in the ZedBoard hardware development kit. The power of this module is performed using a standard +5V DC (2A) power supplier.

The Zynq platform is also provided in a more compact production-oriented form factor, the MicroZed Board. This compact board is powered using either a DC power supply or a dual USB connector for low power applications.

In both scenarios the GNSS FE will be powered externally using a standard +5V DC (0.5A) power supply during development phases. Optionally the power to the Zynq platform could be provided through the GNSS FE that in this case should be connected to a +5V DC (2A) power supply.

DC voltages commonly used in machinery (+12V, +48V) could be easily supported by adding a compatible power supply.

6. Conclusion

In this deliverable the architecture for the AUDITOR GNSS system has been presented in section 1. This architecture is introduced as a combination of multiple hardware/software subsystems that have been related to AUDITOR WPs and partners contributions. The hierarchy of the architecture has been summarized as:

- **GNSS front-end module:** as the main RF receiver module which offers configurable options and a flexible GNSS data stream.
- **Zynq Board:** as the main digital processing platform that embeds two processing units:
 - **Processing Logic (FPGA based):** to implement an efficient preprocessing and adaption of raw samples as part of the High Accuracy Software Module for the Processing System.
 - **Processing System (ARM based):** operating system with custom and third-party libraries that implement the core AUDITOR augmentation algorithms and higher level processes. These processes have been divided into three logic modules:
 - **High Accuracy Software Module:** partially implemented in the Processing Logic consumes the raw GNSS data and is based on customized version of the GNSS-SDR library.
 - **Network Software Module:** implements the iBOGART cloud integration to provide ionospheric corrections.
 - **Agriculture Tools and Services:** that is expected to have a minimal footprint in the embedded GNSS receiver. It is based in cloud services to provide high-value applications for the users.

In section 2 the GNSS D1.2 requirements have been revisited and evaluated versus the proposed architecture. The core requirements than the proposed AUDITOR system could fully achieve, partially cover or asses and possible fulfil during the project span have been discussed and justify.

Finally for all the proposed subsystems an initial subsystem specification has been included in sections 3, 4 and 5. These specifications already include some implementation details and also describe the Zynq platform that will be the core embedded hardware to run the main AUDITOR processes and algorithms. The subsystem specifications will be extended and fully defined in Deliverable 2.2 which is expected to be updated during the project life span as implementation progresses.

7. References

- [1] AUDITOR-D1.1 State of the Art
- [2] AUDITOR-D1.2 Requirement definition
- [3] AUDITOR-D1.3 Test definition
- [4] AUDITOR-D2.2 Subsystem specification
- [5] GNSS-SDR project: <http://gnss-sdr.org/>
- [6] Xilinx, «Zynq-7000 All Programmable SoC Overview - Product Specification,» San José, CA, 2016.