A Space-Borne GNSS Receiver for Evaluation of the LEO Navigation Based on Real-Time Platform

Hung-Yuan Chang, Wen-Lung Chiang and Kuo-Liang Wu

A space-borne GNSS receiver is indispensable for comprehensive validation of performance of the proposed navigation function on a test bench mainly comprising an Attitude Control and Dynamics/Environments Software Simulator (hence ACDESS) running under a RTOS. In this paper, an ACDESS platform built on the PXI system manufactured by National Instruments is proposed. The platform consisting of necessary interfaces for communication between two PXI sub-platforms works with a GNSS RF signal generator which feeds the motion data to RF signal controller in real-time. The proposed NSPO ESGNSSR is aiming for closing the control loop in this test bench. The end-to-end tests of NSPO ESGNSSR in LEO satellite scenarios achieving a positioning accuracy better than $8\,\mathrm{m}$ (1σ, 3D) suggests its promising perspectives for NSPO's future space missions. All responses in the loop meet the mission requirements as well.

1 Introduction

Today, the global positioning system (hence GPS) receiver plays a more significant role in the autonomous competence enhancement of navigation and the facilitation of operation for a variety of spacecrafts and rockets [1, 2]. Thus, many key players in the global space arena pour numerous resources into the research and development of

H.-Y. Chang (✉) · W.-L. Chiang · K.-L. Wu
Flight Control Division, NSPO, HsinChu, Taiwan
e-mail: yuan@nspo.narl.org.tw

W.-L. Chiang
e-mail: peterchiang@nspo.narl.org.tw

K.-L. Wu
e-mail: alexwu@nspo.narl.org.tw

Table 1 Comparision of ESGNSSR and other space-borne GNSS receivers

Model	SEGNSSR	Topstar 3000	SGR-20	Mosaic
Manufacturer	NSPO	Thales Alenia	SURRY	EADS-Astrium
Position accuracy	8 m, 3D, 95%	10 m, 3D, 1s*	10 m, 3D, 95%	10 m, 3D**
Velocity accuracy	5 cm/s, 3D, 95%	1 cm/s, 3D, 1s*	15 cm/s, 3D, 95%	1 cm/s, 3D**
Time transfer with OCXO	80 ns, 95%	200 ns, 3s*	500 ns, 95%	100 ns**
Dimension	150*138*38.5 mm^3	276*117*170 mm^3	160*160*50 mm^3	242*262*25 mm^3
Power consumption	3.5 W	16 W	5.5 W	10 W
Weight (w/o ant.)	0.6 kg	3 kg	0.95 kg	0.7 kg
Operation temperature	$-25 \sim 60$ °C	$-15 \sim 50$ °C (operating) $-25 \sim 60$ °C (start-up)	$-20 \sim 50$ °C	$-25 \sim 60$ °C
Vibration	15 g	13 g	15 g	–
Mission involved		FORMOSAT-2, KOMSAT-2, AMS02, PROBA-2...	UoSAT1 UoSAT12, PROBA-1, BILSAT-1, 14 units flown (2009)	TerraSAR-X, SARLupe, Aeolus, TanDEM-X...

the autonomous space-borne GNSS receivers, including NSPO of Taiwan [3]. A brief comparison of several current space-borne GNSS receives including the ESGNSSR is summarized in Table 1 [4–6]. The ESGNSSR may be competitive among them. The flight model of this receiver manufactured in the early 2015 passed a series of comprehensive functional tests and environmental acceptance tests, etc. which were completed by the end of 2015. One mission considered using ESGNSSR during its development phase is the FORMOSAT-7/COSMIC-2 [7]. A real-time platform equipped with ESGNSSR presented in this paper is used to verify the navigation requirements of this LEO satellite. Moreover, the environmental stress screening process of ESGNSSR will be concisely introduced in this paper as well.

In response to the booming global navigation satellite systems, NSPO is gradually enhancing the capability of L1-GPS receiver and turning it to be a high-precise navigation unit, a multi-mode and multi-band receiver, and even a science payload, the reflectometry receiver of a global navigation satellite system. The fundamental purpose of this extension study is to port some software algorithms, such as signal acquisition and correlation, routinely repetitive computing, and time-consuming functions to the FPGA whose CPU, a digital signal processor (DSP), is solely dedicated to parts of the tracking algorithm, whole navigation algorithm, total orbit propagation algorithm, and so on for the sake of reducing the CPU workload. Due to the fast-paced development and evolution of the FPGA nowadays, the new system architecture upgradable via an FPGA should be able to achieve the goal of being a high-precision navigation receiver, and even a scientific receiver. Last but not least, the test results show that the new system architecture not only retains the original overall performance, but also sets aside more resources available for possibility of future extension.
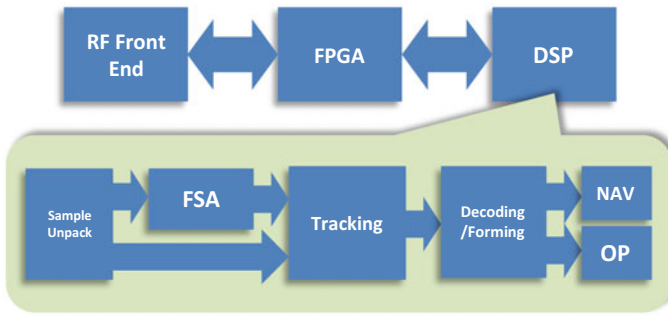
Fig. 1 The key parts of the entire SGR and the calculation process of software algorithm

1.1 Software-Defined GPS Receiver

The predecessor of this ESGNSSR is a software-defined GPS receiver (SGR) in which the L1-GPS baseband signal processing is performed in DSP resulting in 50% of its throughput consumed [8]. The key parts of the entire SGR, the relationships among them, and the calculation process of software algorithm in DSP are shown in Fig. 1.

The Sample Unpack module is mainly for trimming RF raw data and uses the fixed-point DSP instruction set to further achieve the goal of data arrangement. The FSA (Fast Search Algorithm) module is mainly responsible for the signal Fourier transformation, and screening out preliminary satellite signals. The Tracking module takes care of the signal correlation and signal phase-locked loop. Decoding/Forming module is used to decode the used satellite data. Both NAV (Navigation) and OP (Orbit Propagator) modules are in charge of calculating decoded data for navigation.

The entire software architecture is designed to schedule various on-demands computing for real-time tasks according to the weight of each function. As shown in Fig. 2, an external interrupt source is required for driving EDMA to move the RF data from FPGA to DSP. The real-time task in entire software is triggered by EDMA. The execution of the Tracking task in this software system must be completed within 1 ms; otherwise the post-processing results such as NAV and OP will lose reliability. The calculations of NAV and OP are allocated to use the remaining time after the Tracking task is complete in each millisecond. The lowest-weight function, FSA, is only responsible for searching new visible satellites and generating extra information in data pool waiting to be used when system is in idle state

In this SGR, the FPGA in the whole system only plays one role in the data communication and coordination between the DSP and other peripherals. There is not any logic implementation of the software algorithm relevant to the L1-GPS receiver implemented in FPGA. All the logic functions implemented in the FPGA are shown in Fig. 3. The SPI (Serial Programing Interface) module acts as a user interface for setting the operating parameters of RFFE. The CDET (Command Detector) module provides a user interface to execute special instructions. The Wait CE (Chip Enable) module can be used to adjust bus timing.
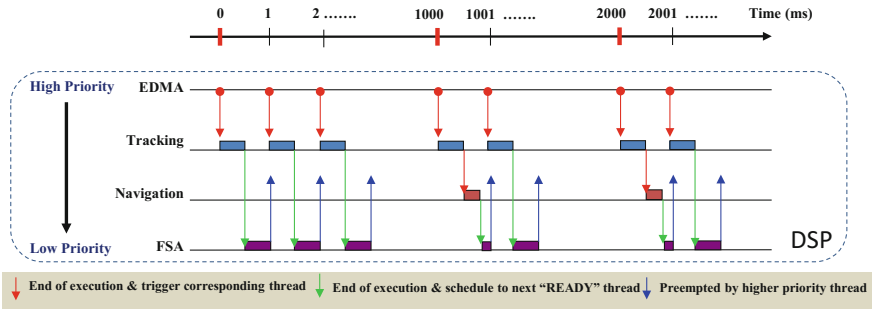
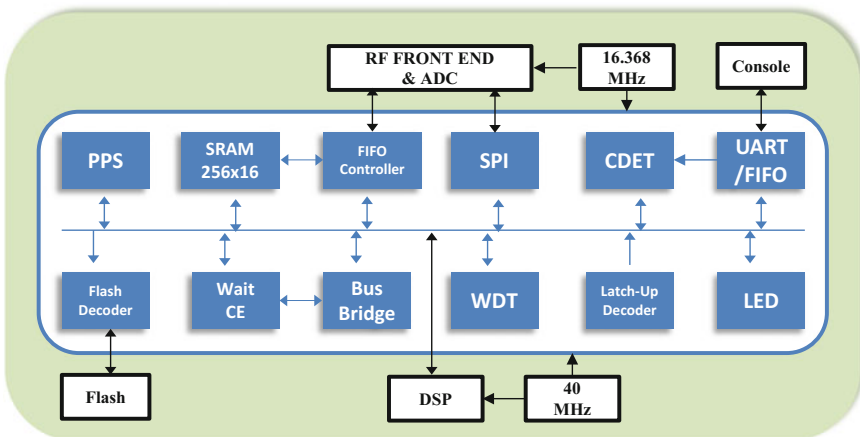**Fig. 2** The real-time scheduling of tasks

Fig. 3 The block diagram of original logical function

It may be a good entry point for understanding the purpose of this paper by considering the current usage of system resources. The actual utilization of DSP bus reached 46.9%, and the real-time tasks also occupied up to 20% of CPU utilization as shown in Fig. 4.

By comparison with the FPGA, the usage of logic gates is currently about 3.78% while memory usage is about 2.68%, as listed in Table 2. This table is derived from the report of system resource usage after the logic circuit has been compiled. Therefore, the idle resources in FPGA have to be used properly in order to reduce the usage of DSP and achieve the goal of being a multi-mode, multi-band, high-precision navigation receiver or even other scientific objectives.

In response to the flourishing global navigation satellite systems, NSPO has planned to gradually extend this SGR to become a multi-mode, multi-band, and high-precision navigation receiver and even be suitable for scientific applications. Starting from 2013, NSPO initiated the preliminary study of extension feasibility in order to achieve this goal. The main purpose is to port the highly repetitive and time-consuming functions from DSP to FPGA in the early research phase of this
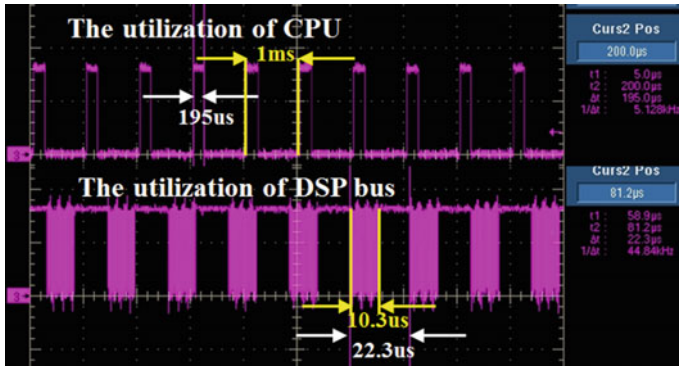
Fig. 4 The system resource usage of SGR

Table 2 FPGA compile report of SGR

CORE	Used:	2844	Total:	75264	3.78%
IO_(W/_clocks)	Used	78	Total:	147	53.06%
Differential_IO	Used:	0	Total:	65	0.00%
GLOBAL_(Chip+Quadrant)	Used:	5	Total:	18	27.78%
PLL	Used:	0	Total:	2	00.00%
RAM/FIFO	Used:	3	Total:	112	2.68%
Low_Static_ICC	Used:	0	Total:	1	0.00%
FlashROM	Used:	0	Total:	1	0.00%
User_JTAG	Used:	0	Total:	1	0.00%

extension work, such as the functions of signal acquisition and signal correlation. The processor is primarily responsible for navigation solution, orbit propagation and other tasks with non-real-time requirements. This paper will present the work and achievements of all the pre-extension research in detail after introducing the mission application and environmental certification of this SGR in Sect. 1.2.

## 1.2 Space Mission and Certification Test

FORMOSAT-7/COSMIC-2 Program is a major collaboration space program between Taiwan and the U.S. In this program, the designated representative of Taiwan is NSPO and its U.S. counterpart is NOAA (National Oceanic and Atmospheric Administration). This program includes 12 radio occultation (RO) science mission satellites plus one augmented NSPO-Built satellite. The 12 satellites are planned to be launched and deployed in two separate clusters of 6 satellites into the designated low and high inclination angle orbits in 2017 and 2019, respectively. The basic information about FORMOSAT-7/COSMIC-2 program is summarized in Table 3.

Table 3 The specification of FORMOSAT-7/COSMIC-2

Specifications	Magnitude
Number of satellite:	13 satellites as a goal
Mass:	<300 kg /Satellite (wet)
Mission orbit:	1st set: low-inclination angle (24 deg), 520 ~550 km circular orbit
	2nd set: high-inclination angle (72 deg), 720 ~750 km circular orbit
Constellation:	One satellite per orbital plane, 6 orbital planes per cluster of satellites
Mission life:	5 years

The NSPO-Built satellite scheduled to be launched in the 2nd cluster will further enhance the radio occultation (RO) data sampling density of the COSMIC-2 constellation and also serve as a qualification platform for NSPO's indigenous key components development. For the NSPO-Built satellite, four in-house components were considered including Fiber Optical Gyro, GPS Receiver, On-Board Computer, and Power Control and Distribution Unit.

The harsh environment, such as total ionizing dosage, single event effect, extreme temperature and launch vibration always cast certain threaten to GPS receivers intended for space applications. Usually the thermal cycling, thermal vacuum, vibration, EMI/EMC, and ionizing radiation are all required testing items for GPSR's environmental stress screening process to meet corresponding mission requirements. Around Nov. 2012, the engineering model of SGR passes part of the entire qualification test campaign, such as the thermal cycle test, the vibration test, and the radiation test. NSPO also completed all EQM qualification tests according to the preset schedule in 2015. Figure 5 shows the on-site snapshots of all environmental
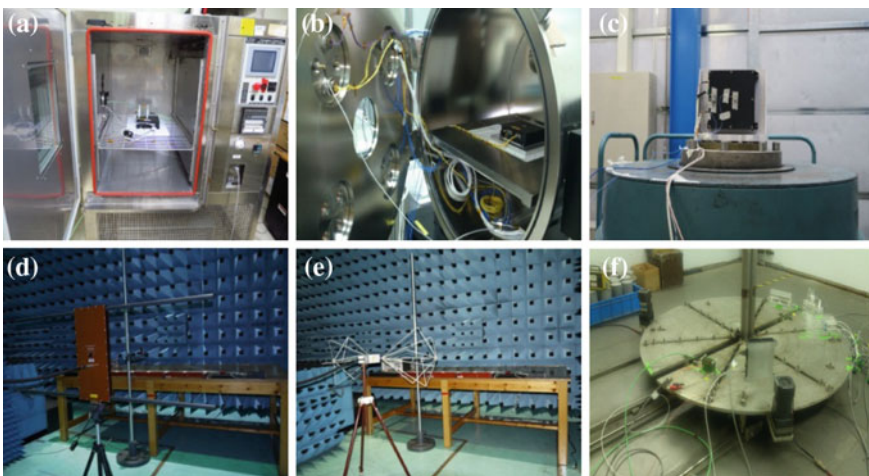
Fig. 5 On-site photos of all environmental tests

tests, including thermal cycling (a), thermal vacuum (b), vibration (c), EMI (d), EMC (e), and radiation (f) tests.

The temperature trending data shown in Fig. 6 were obtained from the thermal cycling test completed in April 2015. The SGR functioned normally while the operation temperature cycled between −40 and +80 °C in one-week testing period.

The temperature trending data shown in Fig. 7 were obtained from the thermal vacuum cycling test completed in May 2015. The SGR also functioned normally while the operation temperature cycled between −40 and +80 °C in one-week testing period.

The vibration test, the EMI/EMC test, and the radiation test, were all completed one by one before the end of 2015. This SGR passed the vibration test in which a
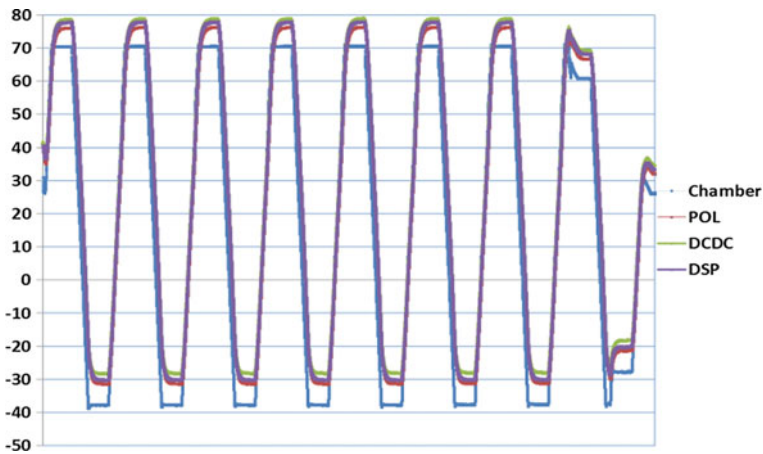

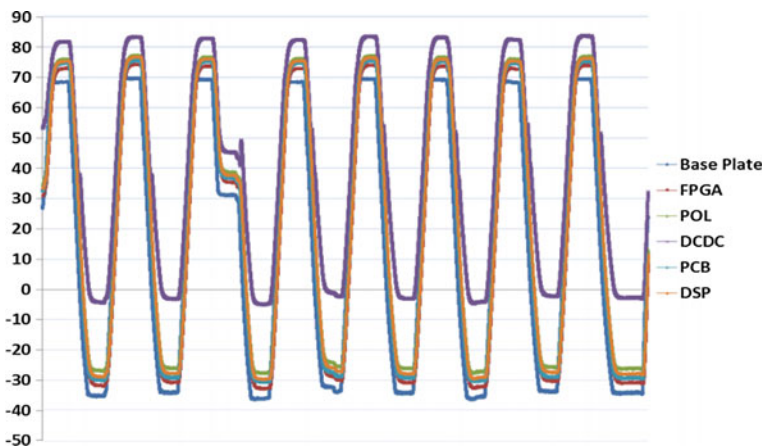**Fig. 6** Thermal cycling test of SGR

Fig. 7 Thermal vacuum cycling test of SGR

testing value of 20 g applied and remained healthy throughout the test. It then passed the EMI/EMC test when a frequency interference between 1G and 2GHz applied during the test, and never exceeded the threshold set by the MIL-STD-461F RE102 while the testing frequency ranged from 1G to 18GHz. It also had normal power consumption before having an exposure dose of 34 Krads.

2 Extendable Space-Borne GNSS Receiver

In order to serve as a multi-mode, multi-band, and high-precision navigation receiver, the prototype of purposed ESGNSSR has been completely designed in 2015. The primary difference between current and previous functions of FPGA is that some extra logics implementing L1-GPS receiver algorithms are added. The new logic block diagram of FPGA is shown in Fig. 8.

The block diagram of extended logic circuits is shown in Fig. 9 in which the following three logics, Sample Unpack module, FSA module, and Correlation module, were ported from DSP to FPGA.

The DMA/Data Switch and Router module is mainly responsible for transferring the RF raw data to the Sample Unpack module and storing the data generated by the FSA and Correlation modules in FIFO buffer for later use. The System Control module is responsible for performing the judgment and management of all tasks in extended logic and it also needs to handle data communication and handshaking between the FSA and the Correlation modules in order to ensure data concurrency.

Trying to utilize remaining resources available in FPGA to extend new functions in DSP for the SGR is our goal. The preliminary study of functional extension feasibility mainly focused on using the FPGA to implement the highly repetitive and time-
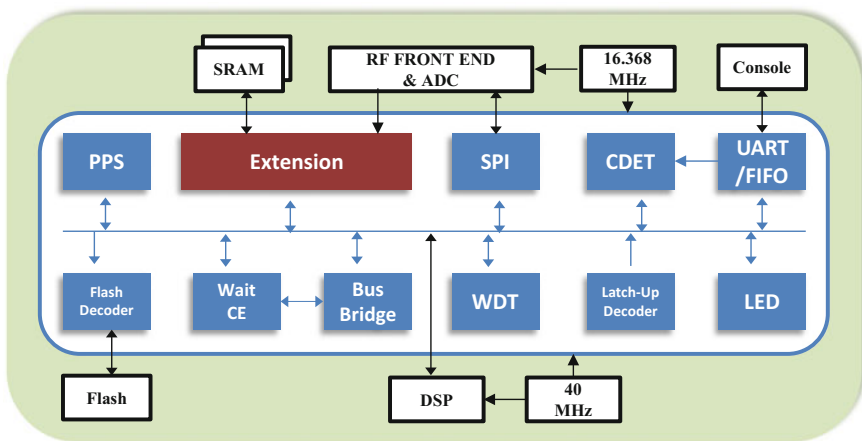

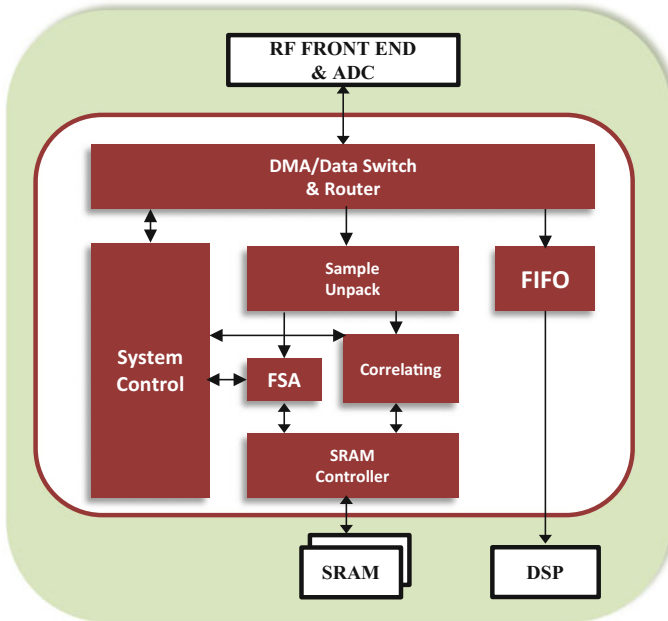**Fig. 8** New logic block diagram of FPGA

Fig. 9 The block diagram of extended logic

consuming functions, such as Sample Unpack, FSA and Correlation in order to reduce the workload of the DSP. However, the signal phase-locked loop, Decoding/Forming, and navigation solution, etc., are still handled by the DSP. Among all GPS algorithms, these three functions occupy most memory of this SGR. Note that FPGA has only a small memory size of 512K Bits compared to 1M Bytes memory built in the DSP. The limitation of available memory could be the bottleneck if these three algorithms are all implemented in the FPGA. Therefore, two pieces of external SRAM's (1M Words each) were added to the FPGA in order to overcome this problem. One chunk of SRAM is dedicated to storing the signal searching results from the FSA, and the other is set to store results of signal correlation from the Correlator.

2.1 Logic Architecture

The main purpose of the Sample Unpack module is to rearrange the signal data generated from the RF front-end, including data decimation and moving average filtering. In Eq. (1), the incoming signal models after data sampling are concisely described.

$$y(t_n) = [R_e(t_n) + jI_m(t_n)] \cdot e^{j2\pi f_s t_n}$$
$$= I_y(t_n) + Q_y(t_n) \tag{1}$$

where f_s represents the sampling frequency. The logic processing steps of Sample Unpack module are introduced as follows. The real and imaginary parts of the raw signals are separated first. Next, four adjacent data points are selected for arithmetical computation. After data combination, the raw data collected at a sampling rate of 16.368 MHz and two bits in each I and Q parts will form a new data structure which has a size of four bits and is sampled at a rate of 4.092 MHz. The architecture of Sample Unpack shown in Fig. 10 in which the plus or minus sign can be appropriately known from the expansion result of Eq. (1).

The FSA module primarily handles signal search in ESGNSSR. Its acquisition algorithm is described as follows. The first step is to perform FFT with incoming signals and can be expressed in Eq. (2).

$$Y_i(k) = \sum_{n=0}^{N-1} \{I_y(t_n) + Q_y(t_n)\} \cdot e^{-j2\pi kn/N} \tag{2}$$

where Y represents incoming signals, k is in range of 0–$N-1$, N is number of samples per millisecond, and i is in range of 0 to 7. Next, an FFT is performed with its replica as described in Eq. (3).

$$L(k) = \sum_{n=0}^{N-1} C_{PRN}(t_n) \cdot e^{-j2\pi kn/N} \tag{3}$$

where L is the FFT result of replica. Third, an inverse FFT is performed with the multiplication result of $Y_i(k)$ and $L(k)$, and it can be formulated as in Eq. (4).

$$M_i(k) = \sum_{n=0}^{N-1} \{Y_i(k) \cdot L(k)\} \cdot e^{j2\pi kn/N} \tag{4}$$

Fig. 10 The architecture of sample unpack
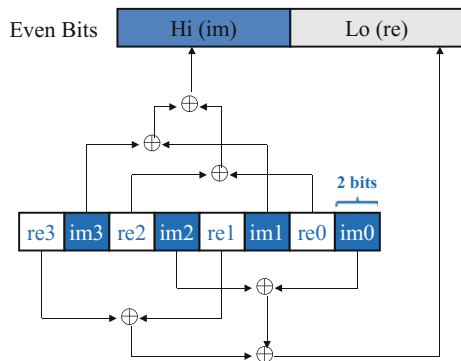
where M is the result of inverse FFT. Then it sequentially performs 8-point FFT by using the results from inverse FFT, and output the desired information, such as Doppler shift and code shift. The Eq. (5) summarizes this process.

$$E\,(k) = \sum_{n=0}^{N'-1} M_i\,(k) \cdot e^{-j2\pi kn/N'} \tag{5}$$

where N' is 8 and k is in range of 0 to $N' - 1$. The FSA is the most complex RTL module in this study. The implemented architecture of the FSA module is illustrated in Fig. 11 where the 4096-point FFT and the 4096-point IFFT share the same FFT IP, and a time-division multiplexing mechanism is used. Beware that the replicated L1-GPS PRN code is stored in the 1023-PRN-code ROM.

At first, the consecutive data in first eight milliseconds are stored in the external SRAM. The FSA module will let the replica be the first to enter the FFT IP and store the results generated from FFT IP in PRN code FFT buffer. Then the module will retrieve one millisecond data from the external SRAM for the FFT IP to do the FFT operation and provide the results generated from the FFT IP for multiplier to perform the multiplication with the data stored in PRN code FFT buffer. Moreover, the data in PRN code FFT buffer used for multiplication should be applied a frequency bin shift between $+130$ to -130 in advance. A complex multiplier is then used to perform data multiplication. After multiplication operation finished, the result will enter the FFT IP to perform the IFFT. The final results generated from the IFFT will constitute a correlation buffer and be stored within the external SRAM. Such a process needs to be executed eight times iteratively.
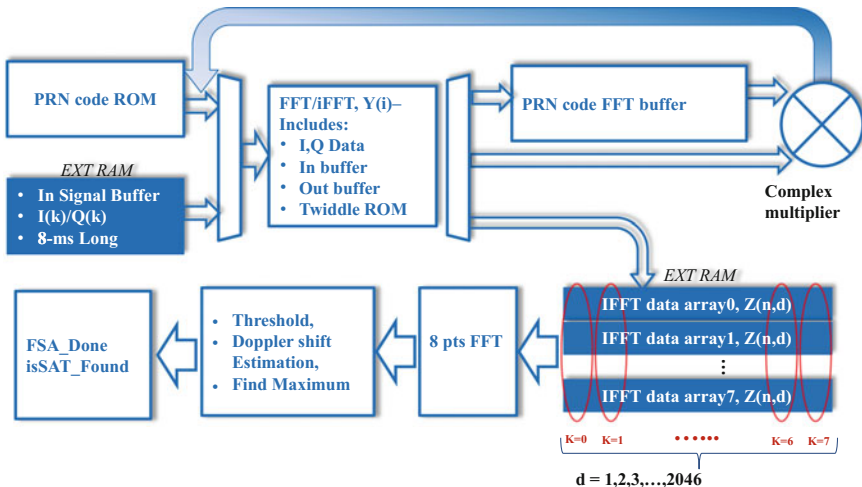

Fig. 11 The architecture of FSA

After completion of IFFT computation by taking 8 ms consecutive data as input, the results are stored temporarily in the correlation buffer. Next, the FFT8, an 8-point FFT IP, will longitudinally retrieve data from this correlation buffer one by one to perform the FFT8. After that, a logic comparator will work on previous results to retain the largest parameters such as sample position, frequency bin, and relevant time stamp in millisecond for later use. Simultaneously, in light of the analysis result, we will select an appropriate threshold which is used to determine whether a L1-GPS signal is found based on the comparison between the largest parameter and this threshold.

The Correlation module is mainly responsible for correlating incoming signals with replica according to parts of the tracking results generated by DSP. Here we can describe the correlation equations in Eqs. (6)–(8).

$$G_P = \sum_{k=0}^{N-1} y\left(t_k\right) \cdot C_{PRN}\left(t_k\right) \cdot e^{-j2\pi f_D t_k} \tag{6}$$

$$G_E = \sum_{k=0}^{N-1} y\left(t_k + \Delta t\right) \cdot C_{PRN}\left(t_k\right) \cdot e^{-j2\pi f_D t_k} \tag{7}$$

$$G_L = \sum_{k=0}^{N-1} y\left(t_k - \Delta t\right) \cdot C_{PRN}\left(t_k\right) \cdot e^{-j2\pi f_D t_k} \tag{8}$$

where f_D is Doppler frequency; Δt is samples, G_P, G_E and G_L are the correlation results in prompt, early, and late phase respectively.

The implemented architecture of the Correlation module is shown in Fig. 12. The Correlation module continually receives data from the Sample Unpack module at a data rate of 8184 samples per mini-second. Moreover, the data are composed of even and odd parts, and each part has 4096 samples. It always keeps 3 ms consecutive data, 3 ms_Buffer, in the external SRAM for the data correlation operation. In every millisecond, the DSP will provide pre-calculated parameters from 12 channels to the Correlation module in advance, including Doppler, code phase, sample position, satellite PRN number, initial phase and phase increment. The Correlation module will finish all calculations in Early, Prompt, and Late phase of 12 channels within one millisecond in order to allow the DSP to be able to initialize the signal phase-locked loop before start of the next one millisecond.

The detailed operations of 3 ms_Buffer in the Correlation module can be categorized into three conditions and are illustrated in Fig. 13. Taking the first condition for example, the input data fill the blue block of 4092 bytes at a rate of 4.092 MHz. The related logic circuits simultaneously retrieve data from the white region of 8184 bytes based on the sample position computed by the FSA while the DSP actuates the Correlation module per millisecond. In this way, the 3 ms_Buffer operating in this way can ensure that the data are always continuous and not overlapped.
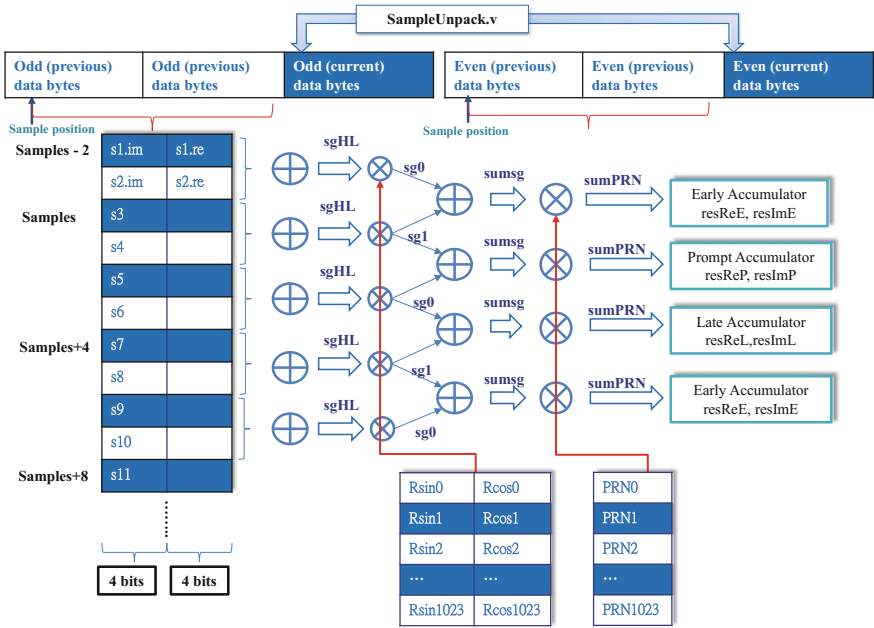
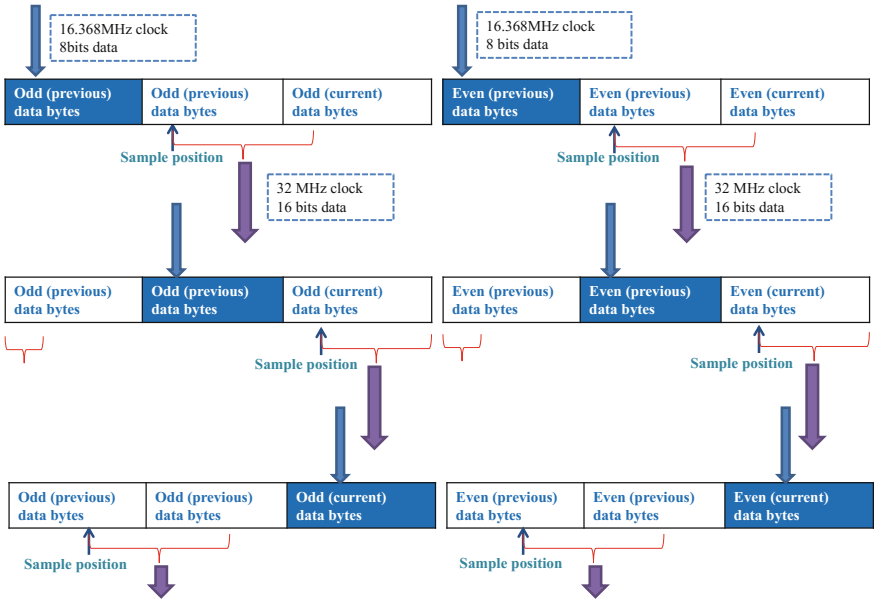**Fig. 12** The architecture of correlation

Fig. 13 The architecture of 3 ms_Buffer

It is noteworthy that improving utilization of the DSP bus lays the groundwork for hardware interface extension, and allows simultaneous processing of more different types of GPS data in the multi-mode and multi-band GNSS receiver. Besides, more effective utilization of CPU resources not only reduces calculation time of the Navigation function in DSP, but also reserves more available computing resources for future application. The feasibility of functional extension of NSPO's space qualified L1-GPS receiver is verified through a close loop test presented in this paper.

By using more powerful FPGA's, the speed of GPS signal acquisition and the channel number of GPS signal correlation can be enhanced and increased greatly in future development. Therefore, the new enhanced configuration can be compatible with varieties of existing GPS satellites, such as Russian's GLONASS, Europe's GALILEO and China's Compass (Beidou). The goal to develop a multi-mode and multi-band GNSS receiver should be achieved in the near future eventually.

2.2 Logic Analysis

All the logic functions presented in this paper use the MODELSIM tool for simulation and verification. The architecture of test bench is illustrated in Fig. 14. First, the recorded GPS signals will be stored in the ROM TABLE, such as GPS_DATA_ROM.v, and then these signals will be imported to the FPGA modules, including the SampleUnpack.v, the FSA.v and the Correlator.v, for test. Meanwhile, both the external RAM simulation model which is RAM_MODEL.v and the Ti-DSP EMIF simulation model which is BUS_MODEL.v are also created for the
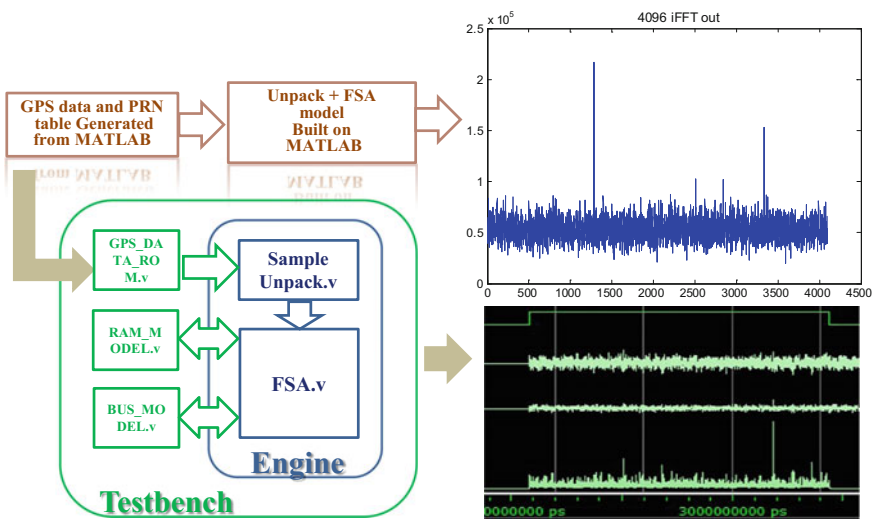

Fig. 14 RTL design simulation/verification flow

test bench in order to simulate all the logic functions as close as to the real condition as possible. The information, such as those GPS signals identified by the FSA module or the computation results of the Correlation module, can be obtained from post-simulation data analysis. Further double check with the algorithms allows us to know if the simulation results are correct as well.

3 The Remote Configuration of a Real-Time Platform

The approach of evaluating the LEO navigation is to build a test bench where resultant motion data due to environment and corresponding control activities can be fed into the RF generator in real time. A simulation scenario of FORMOSAT-7/COSMIC-2 is used to propagate the designated RF signals via a RF signal generator, the Spirent GSS7700, in this study. The proposed test bench consists of two key elements, ACDESS and the GSS7700, controlled by the ACDESS. In the following sections, the in-house ACDESS based on the National Instruments (NI) PXI products will be thoroughly introduced including its hardware interface with the GSS7700.

The original test bench is composed of a remote PC running Windows operation system and a hardware interface capable of maintaining clock synchronization between itself and the GSS7700. Note that, in order to integrate various hardware but still achieve high reliability, the original remote PC of GSS7700 is replaced by NI-PXI products, including the development software CVI, the controller card PXI-8106, the LabView-RT operation system, an M-series interface card PXI-6251, a serial port interface card PXI-8433/4 and so on. The PXI remote platform can provide remote motion data of simulation scenario to the SimGEN PC through TCP/IP network protocol at a rate of 64 Hz. Figure 15 shows the updated configuration of the proposed test bench, which is part of the ACDESS. In order to synchronize devices, the GSS7700 is commanded to generate pulse per second (PPS) signals which are then sent to the PXI platform for clock synchronization. A PXI-6521 in PXI remote platform is used to measure the synchronous pulse signals from the GSS7700 but the PXI does not send any timing compensation signals back to the GSS7700.

As shown in Fig. 15, the PXI products are separated into two dedicated standalone PXI subsets. One serves as the flight software controller and takes care of mode transition, guidance, navigation and attitude control, etc. The other plays the role of hardware modeling and provides information such as flight dynamics, space environment, and ephemeris, etc. The architecture of PXI subsets is illustrated in Fig. 16. Meanwhile, some of the acronyms in Fig. 16 are summarized in Table 4. All of the test results, the performance of ESGNSSR and verification performed on NI-PXI products, are going to be presented in experimental results.

The ESGNSSR is integrated into the ACDESS platform proposed in this paper to form a closed loop test system. The communication process of this system is roughly like this. First, the Dynamics / Environments Software Simulator running on a PXIe-8133 sends the satellite information generated from the numerical models to the GSS7700's controller, SimGEN, through an Ethernet cable. The GSS7700
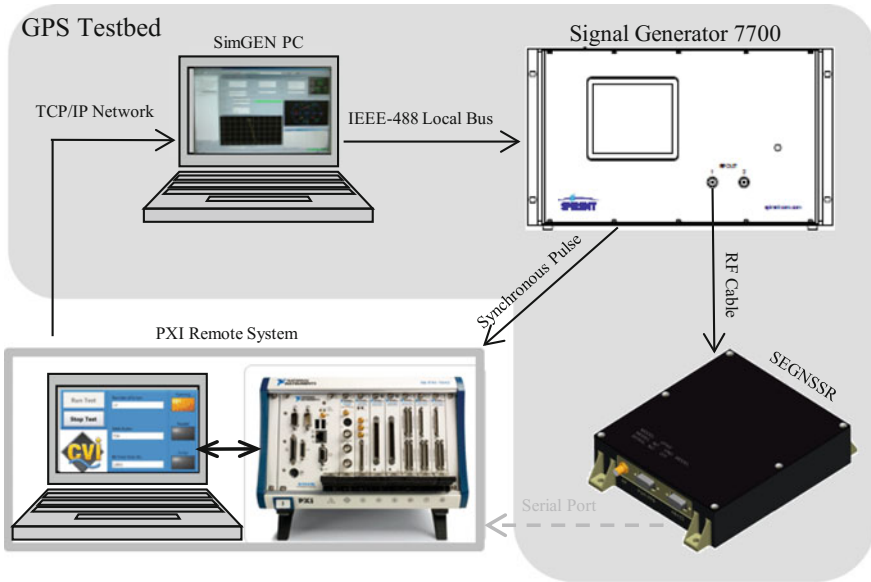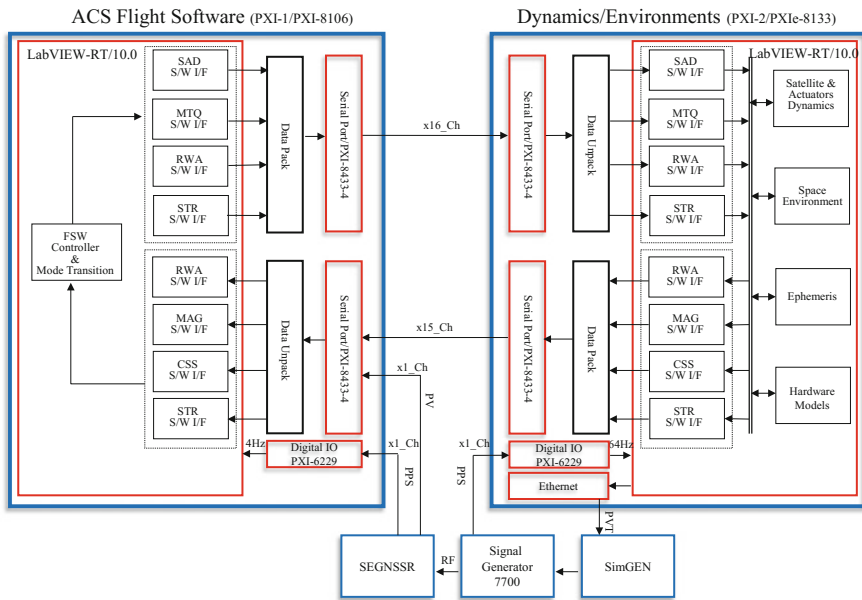
Fig. 15 Configuration of test bench

Fig. 16 The Architecture of ACDESS

Table 4 The Glossary of the acronyms used in Fig. 16

FSW	Flight software
SAD	Solar array driver
MTQ	Magnetic torquer
RWA	Reaction wheel assembly
MAG	Magnetometer
CSS	Coarse sun sensor
STR	Star tracker

broadcasts compatible RF signals based on the information received by SimGEN. The GSS7700 is then connected to the ESGNSSR via an RF cable of approximately 1 m long. The ESGNSSR will instantly resolve the PVT information according to the RF signal received at that time and then send the data to the PXI-8106, which is responsible for executing the Attitude Control Software Simulator via a serial port interface. Up to this moment, partial evaluation work of hardware-in-the-loop (HIL) has completed on such a satellite performance verification platform. But it is still necessary to prepare complete information of the remaining components, as listed in Table 4, to allow the whole system operating properly. It is not the purpose of this paper to integrate all components into this closed loop system. Therefore, the rest components will be substituted by corresponding numerical models, and complete their communication tasks through the serial port interface.

4 Experimental Results

4.1 New System Resource

After its function extended, the FPGA regularly provides the calculation results from FSA and Correlation to DSP per millisecond and the Tracking module in DSP will subsequently continue to finish all remaining algorithm steps. Next, the Navigation module is executed after the Tracking module and allowed to use only the remaining time in each millisecond. The real-time scheduling of tasks is depicted in Fig. 17.

Based on the usage of system resources, one can investigate the system performance after the functional extension work. The DSP bus occupies only 0.9% and the real-time tasks utilize less than 6% of the CPU time as shown in Fig. 18.

Significant reduction of the bus work-load can be attributed to no more large amount of RF raw data processed by DSP directly. Only some critical information instead of raw data is processed by the DSP. Similarly, the huge amount of calculation in the Correlator is also passed on to the FPGA. The number of CPU requests from real-time tasks finally decrease significantly in the DSP.
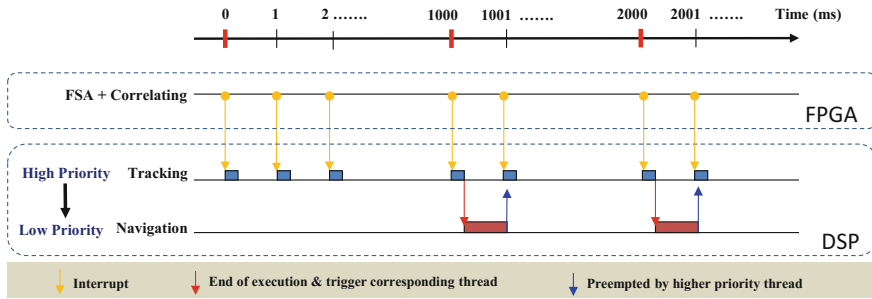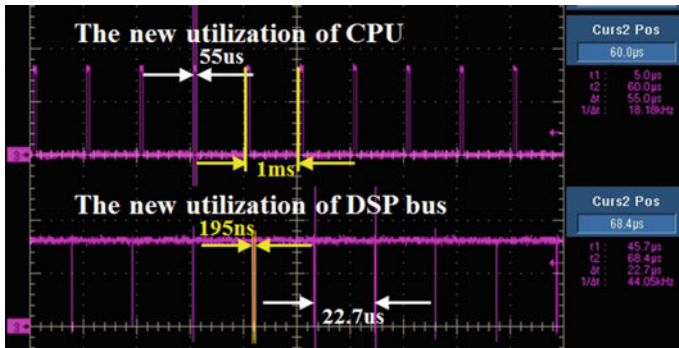
Fig. 17 The real-time scheduling of new tasks

Fig. 18 The system resource usage of ESGNSSR

Table 5 FPGA compile report of ESGNSSR

CORE	Used:	70191	Total:	75264	93.26%
IO_(W/_clocks)	Used	146	Total:	147	99.32%
Differential_IO	Used:	0	Total:	65	0.00%
GLOBAL_(Chip+Quadrant)	Used:	6	Total:	18	33.33%
PLL	Used:	1	Total:	2	50.00%
RAM/FIFO	Used:	109	Total:	112	97.32%
Low_Static_ICC	Used:	0	Total:	1	0.00%
FlashROM	Used:	0	Total:	1	0.00%
User_JTAG	Used:	0	Total:	1	0.00%

In the new FPGA logic design, the usage of logic gates and memory currently reaches about 93.26 and 97.32% separately. The resources of the FPGA are almost used up as listed in Table 5 which is derived from the report of system resource usage after the logic circuit compiled.

4.2 Performance Comparison with SGR

By using a GPS-L1 roof antenna mounted on the NSPO I and T facility with a repeater and a divider, a performance test was conducted to compare the GPS signal tracking capability of both versions of receivers, the original GPS software receiver and the one with extended functions. The test result is shown in Fig. 19 in which the blue line represents the number of satellites tracked and red line means the number of satellites used for navigation solution. Both receivers make no big difference in performance when they track GPS signals on the ground.

Moreover, a RF signal generator (GSS7700) manufactured by SPIRENT and a divider were used in another test in which the space scenario was set to be in Low Earth Orbit (LEO) of 600 km altitude for comparing GPS signal tracking capability of both versions of receivers. The test result is shown in Fig. 20 in which the blue line represents the number of tracked GPS satellites and the red line represents the number of GPS satellites used for navigation solution. Both receivers have similar performance in space scenario test case as well. Many other performance verification tests can be found in published papers and there is no need to elaborate any further here.
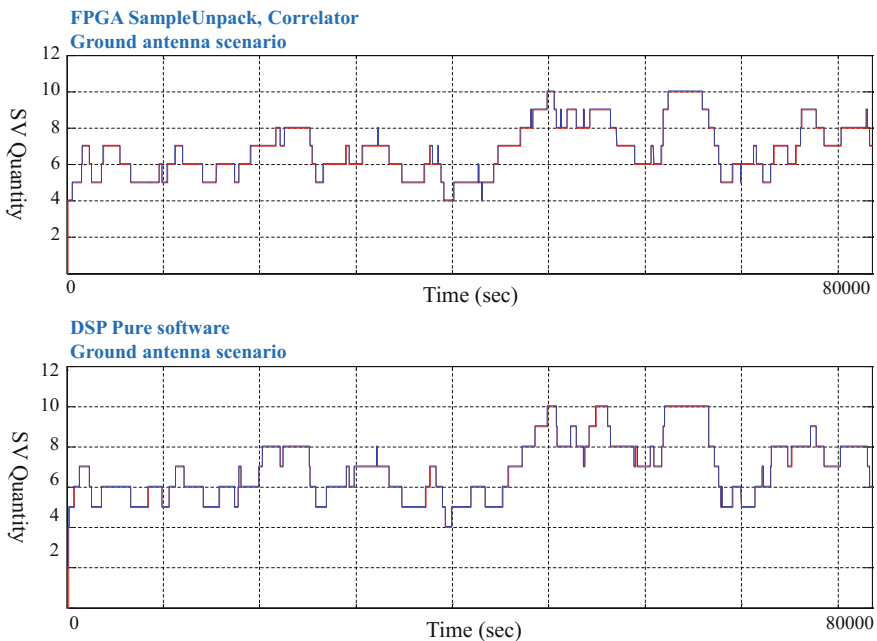

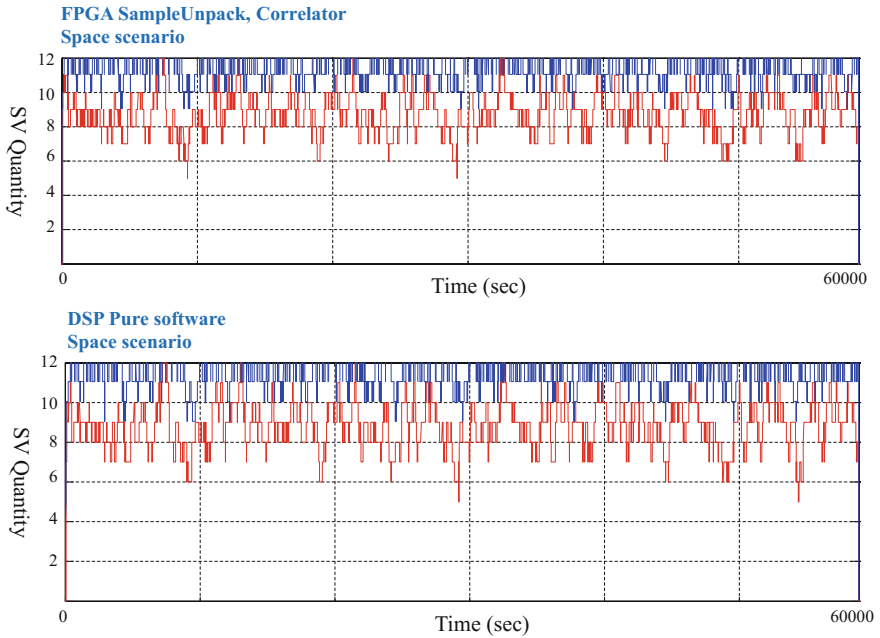**Fig. 19** Tracking performance in ground scenario

Fig. 20 Tracking performance in space scenario

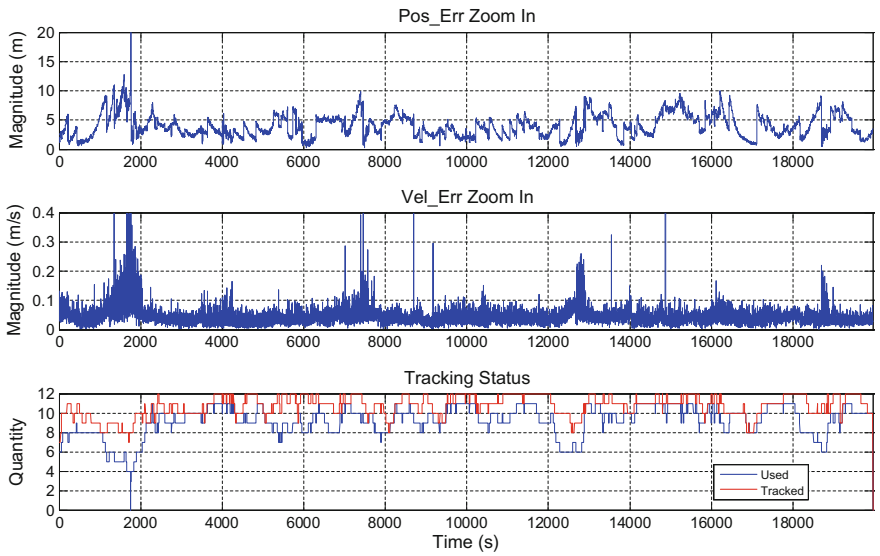**Fig. 21** The performance validation of ESGNSSR

4.3   Performance Validation in ACDESS

As shown in Fig. 21, this navigation error is calculated by the ESGNSSR when the satellite is operating in the Normal Mode scenario. Note that a significant navigation error occurred between 1800 and 1900s due to a large-angle maneuvering of the satellite. However, there are not significant navigation errors bursting during the remaining satellite maneuvers.

5   Conclusions

In this study, the Decimation-In-Time (DIT) Radix-2 is used for FFT. However, it is impossible to perform continuous computation (streaming mode) because its input and output have only one 128Kbits buffer individually. This constraint makes every new FFT process must wait until current FFT process is complete. Hence, the FFT process has a major impact on the real-time requirement in whole system. In addition to implementing new functions for logic circuits, much time and effort were spent on solving time-mismatch issues occurred during system integration involving the FPGA and the DSP. It is worth reminding that appropriate allocation of CPU resources is also an important determining factor in ensuring all real-time tasks completed as scheduled.

Moreover, improving utilization of the DSP bus has paved the way for successful interface extension allowed to accommodate more different types of GPS data in the multi-mode and multi-band GPS receiver. Saving more available CPU resources for future use could be deemed as a remarkable extra achievement of this study. The feasibility of functional extension of NSPO's space qualified SGR has been successfully verified through all tests presented in this paper.

The real-time ACDESS platform proposed in this paper is a functional evaluation system which completely emulates the engineering models of the FORMOSAT-7 NSPO-Built satellite. It includes the complete satellite attitude control algorithms, flight software operation time slots, the integration of hardware interface, and the implementation of dynamics/environment models. To sum up, the successful verification of functionality and performance of this receiver on the designated platform makes NSPO more confident in planning more upcoming satellite navigation missions.

References

1. Kuehl CTF et al (2008) GNSS goes LEON – a LEON-2 based GNSS receiver for space applications. In: 7th internationalesa conference on guidance, navigation and control system
2. Markgraf M et al (2002) A flexible GPS tracking system for sub-orbital and space vehicles. In: 9th Saint Petersburg international conference on integrated navigation systems

3. NSPO, Taiwan (2016) Self-reliant key component of FORMOSAT-7. http://www.nspo.org.tw/
   2008e/projects/project7/components.html
4. SURREY, UK (2016). SGR-20 space GPS receiver. https://www.sstl.co.uk/Products/
   Subsystems/Navigation/SGR-20-Space-GPS-Receiver
5. Airbus Defence and Space, France (2014) MosaicGNSS receiver. www.space-airbusds.com/
   media/document/pdh_8_gps-2014_bd.pdf
6. Gerner JL et al (2000) TOPSTAR 3000 – an enhanced GPS receiver for space applications. In:
   Final presentation of topstar 3000 and the experimental GPS attitude receiver
7. NSPO, Taiwan (2016) Program description of FORMOSAT-7. http://www.nspo.org.tw/2008e/
   projects/project7/program-description.html
8. Chang, H-Y et al. (2013) Performance demonstration of NSPO space-borne GPS receiver. In:
   Proceedings of ION