Chapter 2
Brazilian Nano-satellite with Reconfigurable SOC GNSS Receiver Tracking Capability

Glauberto L.A. Albuquerque, Manoel J.M. Carvalho, and Carlos Valderrama

Abstract This paper presents a flexible architecture for a GPS receiver using Partial Reconfiguration (PR) on a System on Chip (SoC) device consisting on an FPGA and two ARM cores. With built-in error-correction techniques offered by modern SOCs, this device meets the requirements of a Brazilian nanosatellite for CONASAT constellation. This receiver benefits from PR, thereby increasing system performance, hardware sharing, and power consumption optimization, among others. Additionally, all the advantages favor in-orbit reconfiguration. The proposed architecture, as requested, uses COTS components.

2.1 Introduction

CubeSats became an affordable alternative for space missions of emerging countries [1] and even for developed ones. Indeed, the CubeSat specification makes possible to decrease launching costs and development time of small satellites. This specification, which began in 1999 from collaboration between the California Polytechnic State University and the Stanford University, has helped universities around the world developing science and space exploration. Although CubeSats were primarily intended for use with educational purposes, nowadays there are commercial, military and interplanetary space missions using this technology, as a valuable alternative for many space mission profiles [2–4].

G.L.A. Albuquerque (✉)
Barreira do Inferno Launch Center—CLBI, Parnamirim, Brazil
e-mail: glauberto@engineer.com

M.J.M. Carvalho
CRN—Centro Regional do Nordeste, Instituto Nacional de Pesquisas Espaciais—INPE, Natal, Brazil
e-mail: manoel@crn.inpe.br

C. Valderrama
SEMi—Electronics and Microelectronics Department, University of Mons, Mons, Belgium
e-mail: carlos.valderrama@umons.ac

F. Kastensmidt, P. Rech (eds.), *FPGAs and Parallel Architectures for Aerospace Applications*, DOI 10.1007/978-3-319-14352-1_2

Advances in electronics and MEMS combined with techniques such as Software Defined Radio (SDR) and Digital Signal Processing (DSP) have contributed to reduce costs while facilitating their development. In particular, Field Programmable Gate Arrays (FPGAs) has proven to be a cost effective tool for the development of projects in different areas beyond SDR. In addition to the reconfiguration flexibility, its main advantage over other devices is their low power consumption [5]. Indeed, this is a very important attribute for space applications. In orbit, a satellite can easily get energy from solar panels and batteries, but at the cost of adding extra weight to the structure. Thus, to reduce the total volume, satellites must be designed from devices with reduced size and low power consumption.

Apart from specificities of each space mission profile, all satellite payload contain some kind of communication link and navigation control, for which Global Navigation Satellite System (GNSS) receivers are envisioned nowadays. Additionally, such sub-systems must be robust and reliable to operate in hostile environments without failure. Regarding this concern, the Partial Dynamic Reconfiguration (PDR) capability of FPGAs could be an additional attribute for space applications [6, 7]. This procedure, not only allows adaptable payload in orbit, but also offers a certain degree of radiation tolerance (e.g. faulty system re-initialization, replacement and upgrade).

This paper proposes a low cost GPS receiver architecture based on FPGA SoC COTS to meet the requirements of CONASAT satellites. This receiver intends to take advantage of modern FPGA-based SoC and Partial Reconfiguration techniques for use in space applications and mission recovering.

2.2 CONASAT

2.2.1 CONASAT Project

CONASAT is a project based on a nanosatellites constellation funded by INPE, Brazil's National Institute for Space Research. Its main mission is to collect environmental data from thousands of DCPs (Data Collection Platforms) distributed throughout the Brazilian territory and its seacoast. This constellation will replace the former SCD1 and SCD2 satellites, still active, although they have already exceeded their design life.

Some relevant guidelines concerning the CONASAT project are [8]:

- To develop expertise in the field of space missions, especially on nano-satellites;
- It must satisfy the lowest possible cost for an acceptable level of reliability and mid-term life-time of 5 years;
- It must use COTS components and commercial subsystems as much as possible;
- It must provide such a flexible and modular platform that could be adopted by subsequent generations of satellites of the constellation;

- CONASAT satellites must be CubeSat compliant;
- It must generate opportunities for Brazilian technology industry.

CONASAT will be the spatial segment of the Brazilian System for Environmental Data Collection (SBCDA). Brazil already produces its own DCPs and some parts of a CONASAT satellite. As much as possible, other parts of the satellite should be produced by Brazilian experts. For instance, the current communication protocol between DCPs and satellites will be modified to allow bidirectional data exchange.

CONASAT satellites will use Low Earth Orbits (LEO—altitudes from 500 to 800 km). Thus, satellites will not be over the Brazilian territory all the time. Downtime will then be occupied by other applications or services. For instance, it is planned to extend SBCDA services for monitoring fishing boats. In these cases, it's desirable to have CONASAT parts implemented on reconfigurable hardware supporting tasks on demand. Regarding radiation tolerance, it is important to note that the satellite orbit, at an altitude of about 600 km, belongs to a region with low ions density.

2.2.2 The CONASAT Satellite Architecture

Generic architecture of the satellite, shown in Fig. 2.1, is not remarkable compared to others. It consists on a full redundancy of all major subsystems, including the *Power Management* one. Thus, it can be considered as having two satellites within one mechanical infrastructure. This choice intends to increase overall system reliability due to the fact that the design guidelines of CONASAT allow the use of COTS components. Another reason is the MTBF (Mean Time Between Failures) of CubeSat parts readily available on the market. They are not prepared for a midterm lifetime.

The *Redundancy Control* subsystem decides which sub-system to activate each time. The *Attitude Control* subsystem includes a magnetorquer (iMTQ), stellar gyroscope, 3-Axis gyroscope, star tracker and reaction wheels. This satellite also uses a GNSS Receiver (GPS receiver, in this case) to simplify orbital prediction. The use of multiple sensors obeys to the principle of achieving maximal reliability. However, while the combined use of sensors increases its efficiency. On the other side it also raises the weight of the satellite and its power consumption. Moreover,

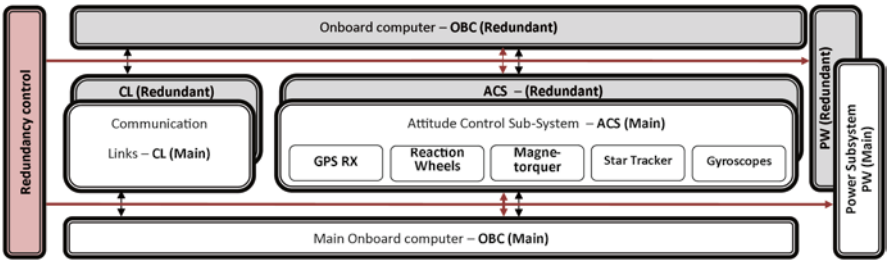

Fig. 2.1 CONASAT functional architecture (adapted from [8])

the processing capability of the GPS receiver must be adapted to the orbital velocities. Therefore, the way of space GPS receivers handle data must be carefully adapted. The *Communication Subsystem* is just composed by an UHF uplink and S-Band downlink. It is responsible to retransmit to ground stations data received from DCPs. The *Power* and *Attitude Control* subsystems have in-orbit so specialized tasks which cannot take other responsibilities.

The GPS Receiver is the only subsystem whose functionality should be modified in orbit, on demand, to accomplish a particularly required task. For that reason, this receiver must be built based on a software platform. Moreover, due to the requirements of performance and power, this flexibility must be supported by reconfigurable hardware. However, there is no such a device on the market, an "on-orbit reconfigurable GPS receiver for Cubesats". With an optimal choice of the FPGA device, unused logic elements could provide added functionality or even, when the receiver is idle, could also be possible to share the entire platform. This would reduce the physical size and the number of electronic devices, with favorable effects on energy consumption and the satellite's overall weight.

2.3 Software GNSS Receivers Architecture

As we saw above, the software-based approach for a GNSS receiver was a natural choice in terms of design, especially because, in the case of a GPS, signals from the GPS satellites constellation use digital modulation (BPSK). Taking this into consideration, the assembly of a GPS Receiver (or other GNSS System), despite some difficulties, is not an unattainable task [9]. Because of the dominance of GPS in this domain, the remainder of this paper will consider the GPS as a reference to explain the proposed architecture.

According to the chipset used in the design we can identify two approaches: hardware or software receivers. Hardware receivers use ASIC devices to accomplish all tracking and navigation tasks. Those commercially available have limited or no applicability in aeronautics or spatial domain. In software receivers, signal processing tasks are programmable, by using a GPP (General Purpose Processor), DSP, GPU, or even reconfigurable hardware (FPGA). Sometimes, developers work with a combination of these devices [10–12].

We can see the GPS receiver basic architecture in Fig. 2.2. Although the different types of GNSS receivers available are tailored to the different target applications, all these basic architectures include the same functional blocks.

After the *Antenna*, required to amplify and filter the incoming radio signal, the *Front-End* is responsible for down-conversion and digitalization of this analog signal. The *Baseband Processing* block acquires and monitors each incoming signal to calculate its own position and speed. For each tracked satellite it is required to have one of these blocks. Thus, it extracts observable and navigation data from each processed channel. Theoretically, up to 12 GPS satellites can be tracked at the same time, but to calculate its position the receiver only needs four of them. After correctly tracking the signals, the measurement data obtained are sent to the *Application Processing* block.

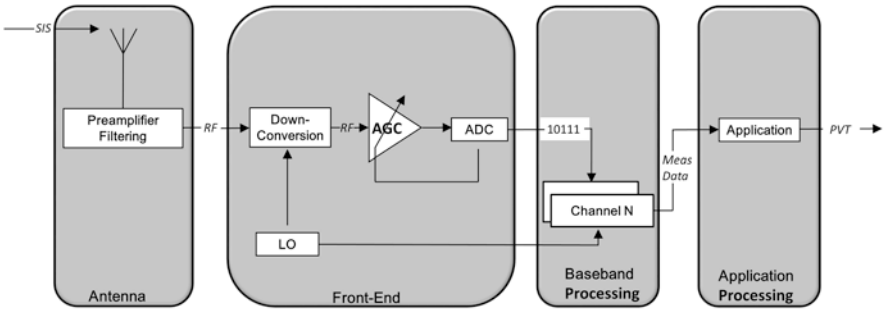

Fig. 2.2 Generic GPS receiver architecture [13]

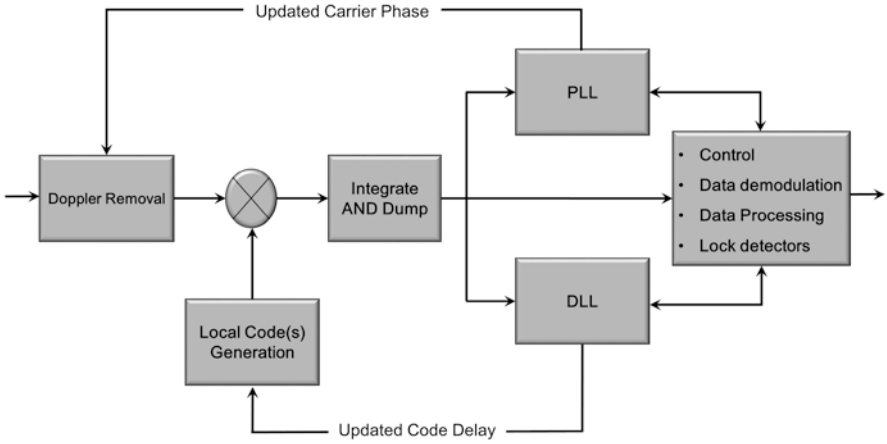

Fig. 2.3 Baseband signal processing [13]

This block uses the information from the tracking loops for different purposes. Typical applications are: ionosphere parameters monitoring, DGPS (Differential Global Positioning System) calculation, static and kinematic surveying.

The processing time of the *Baseband Processing* determines two categories of receivers: real-time and post processing. In post processing, the baseband information is used to obtain correlations between the incoming signals and an internal replica, used as reference. This produces intermediate data stored to be further processed in batch mode by complementary algorithms. Thus, the receiver is not able to locate the position in real-time. That delay is critical for orbital speed navigation, implying additional power processing and control over tracking algorithms.

Baseband Processing includes all the algorithms to find and follow a visible GPS signal, through the synchronization with a known PRN code, and remove errors, as best as possible. This process is built around the principle of signal correlation: the incoming signal is repeatedly correlated with a replica of the expected PRN code, which is known a priori. Its functional structure is depicted in Fig. 2.3. To extract a valid significance from the correlation, the local replica is generated in the receiver

taking into account the signal carrier phase, code delay, Doppler frequency, and PRN code [12]. To obtain maximum correlation, the *DLL* and *PLL* blocks are in charge of follow the code and carrier delay, respectively.

2.4 Hardware Design

2.4.1 The Front-End

Even for software GNSS receivers, most of front-end modules are ASIC devices. On the market there are dozens of options, even a reconfigurable alternative has recently emerged [14]. Brazilian scientists have used the GP2000 chipset to build a GPS receiver for sounding rockets [15]. Moreover, as demonstrated in [16], the GP2000 chipset is sufficiently radiation-proof for use in LEO without major modifications. However, many other GPS receivers for space applications are based on the GP2015 front-end, for instance, those produced by DLR and Surrey Technologies [17, 18]. So, the GP2015 family can be considered as a certified choice.

2.4.2 Baseband Processing Module

Although the GP2015 front-end module is a good choice for this receiver, the use of the other chips of the family will lead us to a hardware receiver; losing all the advantages of the software approach in terms of algorithm flexibility and associated data processing efficiency.

The GP2015 front-end at a sampling frequency of 5.71 MHz provides 2-bit samples. The bandwidth required by the sample data rate is:

$$f_s = 5.71\,\text{Msamples/s} \tag{2.1}$$

$$\text{N}_{\text{Samples}} = 2\,\text{bits}(\text{sign/magnitude}) \tag{2.2}$$

$$\text{BW} = f_s * \text{N}_{\text{Samples}} = 11.42\,\text{Mbps} \tag{2.3}$$

This bandwidth can be easily achieved with modern FPGA transceivers of up to 1 Gbps and, if necessary (e.g. Doppler removal) incoming data can be oversampled.

A generic tracking channel is depicted in Fig. 2.4. This channel, composed of *accumulators* and *carrier/code generation* units, requires around 1.5 k logic elements on a single FPGA [9]. Remaining modules, acquisition and tracking loops, will take 3 and 6 k logic elements, respectively. Since most of operations are binary, random SEU have not major influence on the final correlation.

Modern FPGAs can provide more that 100k logic elements. This is enough to contain a GPS receiver with ten parallel baseband signal processing units. This can be extended by introducing pipeline techniques to share single tracking channels.

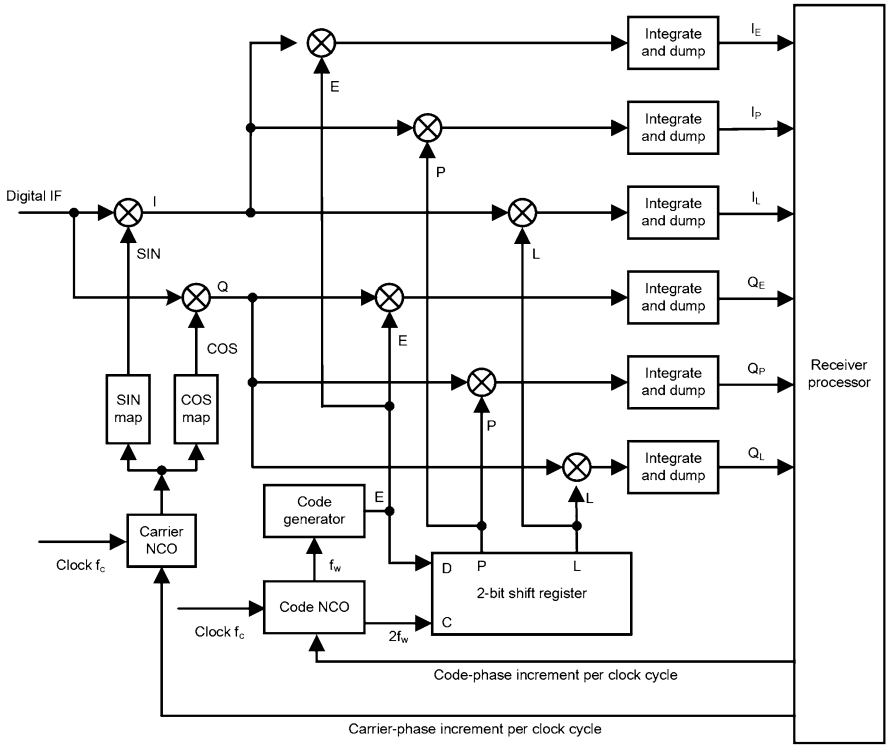

Fig. 2.4 Generic digital receiver channel block diagram

For instance, operating at 200 MHz with an 11.42 MHz sample clock, a given channel can track up to 16 GPS satellites at a time. However, CONASAT imposes orbital velocities, thus parallel tracking channels are better suited.

In the case that power consumption is not a constraint, GPPs, DSPs and GPUs, have enough power processing to build real-time receivers. However, when looking for balance of power processing and low power consumption, FPGA are a better choice. If necessary, additional tracking channels may even become available on demand by using the DPR technique (Dynamic Partial Reconfiguration). Moreover, DPR alone can also be used to mitigate SEU, as in [7, 19, 20], or even combined with TMR as in [21, 22]. As will be shown later, those alternatives have also been considered to meet the requirements of our proposal.

2.4.3 Application Processing Module

Application tasks must be quickly created to support the specifics of a particular mission. This adaptability is a key requirement to ensure the multiplicity of application cases and the sustainability of such a platform. ARM microprocessors appears

as a software processing module in different commercial GPS receivers [17, 18] with the added value of Linux OS. In FPGA there are also softcores like, for instance, NIOS, but not powerful enough for additional tasks. There is also the hardened version of the LEON processor. However, modern SOC FPGAs provide dual core ARM processors on the same package and the possibility to apply some fault mitigation and correction techniques such as in [23]. Although, to assure reliability of the overall system, some radiation hardened devices must still be used. This requirement particularly applies to the memory device, which must keep protected critical data for both, the FPGA and the processors. In addition, preserved application software or reconfiguration data are used when needed or to replace faulty modules.

2.4.4 SEU Mitigation in COTS FPGA and SOC

Radiation hardened devices, combined with Single Event Upset (SEU) error mitigation and CRC, is an important requirement not always supported by FPGAs. Looking at the market of new devices, we found modern ones with built-in SEU error mitigation based on CRC method. This on-chip error detection performs the following operations without any impact on the fitting performance of the device [23]:

- Auto-detection of CRC errors;
- Optional CRC error and identification in user mode;
- Testing of error detection functions by deliberately injecting errors through the JTAG interface.

At the same family of chip there is a SOC device. This device includes high speed transceivers and dual core ARM processors.

Apart of internal mitigation of SEUs, aluminum shielded is included in CONASAT design. According to [24] a 1 mm thick aluminum box absorbs approximately 6000 rad.

2.4.5 Proposed Architecture

As we can see in the Fig. 2.5, the architecture is designed to take advantage of all built-in circuits and Partial Reconfiguration in order to achieve a reliable receiver to be used in spatial applications. This architecture is better than the proposed in [25] in terms of power consumption. Literature survey has showed that high-end FPGAs have a huge throughput advantage over high performance DSP processors for certain types of signal processing applications. FPGAs use highly flexible architectures which can be of greatest advantage over regular DSP processors [26].

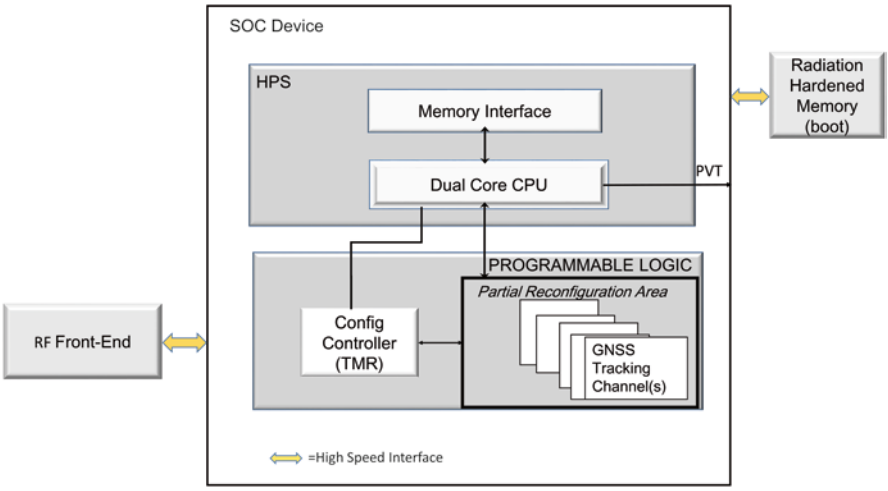

Fig. 2.5 Proposed architecture

The one-chip architecture also take advantages in terms of radiation protection since the area of silicon components are obviously smaller than any other architecture with two or more devices.

The overall architecture is seen in Fig. 2.5. The *Config Controller* is responsible to verify all parts of the algorithm are working correctly. It is also responsible for the FPGA reconfiguration, error recovering or to change the application. After critical errors not recovered by the built-in *CRC control*, the *Config Controller* is able to restart the receiver. To improve reliability of the overall system this part of software is designed using the TMR technique. The two ARM cores in the SOC so the system (HPS block) could take advantages of the dual CPU fault tolerance techniques [27]. Critical parts of the software code are stored in a radiation hardened memory.

2.4.6 Improving Cold Start Time

The *Doppler Removal* module we see in Fig. 2.3 is responsible to correct inaccuracies in the apparent Doppler frequency of the satellite and "zero-beat" the signal. A Doppler shift is the change in frequency of a wave (or other periodic event) for an observer moving relative to its source. If we take the relative motion between the GPS satellite, with orbital speed of 3.9 km/s, and a car, assuming at 40 m/s (150 km/h) traveling over the Equator (greatest Earth rotational speed: about 460 m/s) we could reach, at a maximum, 1.3 km/s, which is equivalent to a Doppler shift of ±6.8 kHz. If we replace the car by a LEO satellite, with orbital speed up to 9 km/s, this generates a significant Doppler frequency shift amounting to ±45 kHz.

On Cold Start mode, when no prior information about Doppler shift, the incoming signal is first stripped of its Doppler frequency, and then correlated with one (or more) PRN code replicas generated locally (according to the current estimation of code delay). When the receiver does not have a good estimation of the initial Doppler, the receiver must correlate the signal with a range of all possible Doppler shifts. Once all Doppler and code shifts have been composed, the peak magnitude is compared to a predefined carrier-to-noise threshold to determine if a GPS satellite has been located. This method consumes fewer hardware resources, but increases the cold start time.

On the ground, a GPS receiver can see a given satellite for several hours. In space applications the visibility time is, in most cases, less than 50 min. Besides that relative motion speed between each GPS satellite and CONASAT changes very quickly, so the receiver must improve the cold start time in order fix a navigational solution.

In this architecture each GPS channel is responsible to track a specific PRN code. Once an entire PRN code is transmitted in 1 ms, the accumulation period is typically between 1 and 20 ms. With a sample data of 5.71 MHz and, for instance, a clock system of 400 MHz, we could make about 70 times the correlation with the same data. In each time slice the generated code is created with different Doppler shifts. With this strategy, the time to track the first GPS satellite signal decreases to some milliseconds.

PR is a useful technique to implement this architecture because after Cold Start all unnecessary FPGA's resources could be released to another application. PR also allows to create an optimal Sleeping Mode, when the CONASAT has no visibility over Brazilian territory and only critical data and applications must be preserved. The receiver could benefits from PR in other phases of the receiver operation since some parts of the hardware resources could run specialized algorithms under certain conditions and thus, this resource can be released when becomes not needed anymore.

2.5 Market Options

Looking at the market of GNSS spaceborne receivers most of available devices have a mass of some kilograms and power consumption of tens of watts. These receivers are not suitable for nanosatellites. Some are constructed with COTS components and can be used in space missions within a low radiation orbit. In [28], we can find a detailed list of spaceborne receivers available on the market. This list was published in 2008 but currently it has no significant changes because performances of new products are very similar to old ones. Other alternatives are the dedicated chips used in Cubesat products. However, some experiences with such miniaturized ASCI receivers fail to provide valid navigation fixes [29, 30]. None of these receivers in the market could be reconfigurable in-orbit to perform a completely different application.

2.6 Conclusions

With the proposed architecture, CONASAT could take advantage of COTS components in order to accelerate design process and decrease costs. This device, using PR presents high level of adaptability. This electronic framework could be used to develop other applications under SDR techniques. One of natural improvement to this receiver is include GALILEO tracking channels.

References

1. Woellert K, Ehrenreund P, Ricco AJ, Hertzfeld H (2010) Cubesats: cost-effective science and technology platforms for emerging and developing nations. Adv Space Res 47:678–679
2. Taraba M et al (2009) Boeing's CubeSat TestBed 1 attitude determination design and on-orbit experience. In: Proceedings of the 23rd annual AIAA/USU conference on small satellites
3. Weeks D, Marley AB, London III J (2009) SMDC-ONE: an army nanosatellite technology demonstration. In: Proceedings of the 23rd annual AIAA/USU conference on small satellites
4. Staehle RL, Anderson B, Betts B, Blaney D, Chow C, Friedman L, Hemmati H, Jones D, Klesh A, Liewer P, Lazio J, Lo M, Mouroulis P, Murphy N, Pingree PJ, Puig-Suari J, Svitek T, Williams A, Wilson T (2012) Interplanetary CubeSats: opening the solar system to a broad community at lower cost. In: Final report on NIAC phase 1 to NASA Office of the Chief Technologist, Jet Propulsion Laboratory, 2012. (Submitted to Journal of Small Satellites. http://www.nasa.gov/pdf/716078main_Staehle_2011_PhI_CubeSat.pdf)
5. Choi S et al (2003) Energy-efficient signal processing using FPGAs. In: Proceedings of the 2003 ACM/SIGDA eleventh international symposium on field programmable gate arrays. ACM
6. Savani VG, Mecwan AI, Gajjar NP (2011) Dynamic partial reconfiguration of FPGA for SEU mitigation and area efficiency. Int J Adv Technol 2(2):285–291
7. Zhang J, Guan Y, Mao C (2013) Optimal partial reconfiguration for permanent fault recovery on SRAM-based FPGAs in space mission. Adv Mech Eng
8. INPE (2011) Constelação de nano satélites para coleta de dados ambientais: documento de descrição da missão DDM. http://www.crn2.inpe.br/conasat1/Documentos/gerais/Documento%20de%20Descri%E7%E3o%20da%20Miss%E3o%20%28Equipe%20CONASAT%29.pdf
9. Shapiro AM (2010) FPGA-based real-time GPS receiver. Dissertations, Cornell University
10. Hobiger T et al (2010) A GPU based real-time GPS software receiver. GPS Sol 14(2): 207–216
11. O'Hanlon B et al (2011) CASES: a smart, compact GPS software receiver for space weather monitoring. In: Proceedings of the ION GNSS meeting
12. Dovis F et al (2001) On the tracking performance of a Galileo/GPS receiver based on hybrid FPGA/DSP board. In: Proceedings of the 18th international technical meeting of the satellite division of institute of navigation (ION GNSS 2005)
13. ESA Navipedia (2014) Generic receiver description. http://www.navipedia.net/index.php/Generic_Receiver_Description#Receiver_overview. Accessed 22 June 2014
14. Noroozi A (2013) A reconfigurable GPS/Galileo receiver front-end for space applications. Dissertations and Theses, Delft University of Technology, Netherlands. Web. 12 June 2014
15. Francisco M, Albuquerque G, Rapôso T (2011) A GPS receiver for use in sounding rockets. In: 20th symposium on European rocket and balloon programmes and related research, vol 700
16. Unwin MJ, Oldfield MK, Underwood CI, Harboe-Sorensen R (1998) In: Proceedings of the 11th international technical meeting of the satellite division of the Institute of Navigation (ION-GPS-1998), Nashville, 15–18 Sep 1998, pp 1983–198

17. Markgraf M et al (2001) A low cost GPS system for real-time tracking of sounding rockets. European space agency-publications-ESA SP 471, pp 495–502
18. Underwood C et al (2004) Radiation testing campaign for a new miniaturised space GPS receiver. In: IEEE radiation effects data workshop, July 22, Atlanta, USA, pp 120–124
19. Graczyk R et al (2012) Dynamic partial FPGA reconfiguration in space applications. In: Photonics applications in astronomy, communications, industry, and high-energy physics experiments 2012. International Society for Optics and Photonics
20. Jan K, Straka M, Kotasek Z (2012) Methodology for increasing reliability of FPGA design via partial reconfiguration. In: The first workshop on manufacturable and dependable multicore architectures at nanoscale (MEDIAN'12), Annecy
21. Azambuja JR, Sousa F, Rosa L, Kastensmidt FL (2009) Evaluating large grain TMR and selective partial reconfiguration for soft error mitigation in SRAM-based FPGAs. In: On-line testing symposium, IOLTS 2009, pp 101–106
22. Pilotto C, Azambuja JR, Kastensmidt FL (2008) Synchronizing triple modular redundant designs in dynamic partial reconguration applications. In: SBCCI '08: Proceedings of the 21st annual symposium on integrated circuits and system design. ACM, New York, pp 199–204
23. Altera Corporation (2012) SEU mitigation for cyclone V devices. http://www.altera.com/literature/hb/cyclone-v/cv_52008.pdf. Accessed 2 May 2014
24. Wertz JR, Larson WJ (1999) Space mission analysis and design. Kluwer Academic, Dordrecht
25. Bedmutha ND, Biraris PN, Shah JP (2013) A low cost GNSS software receiver design with SEE mitigation approach for microsatellites. In: Space science and communication (IconSpace), 2013 IEEE International Conference on IEEE
26. Hayim A, Knieser M, Rizkalla M (2010) DSPs/FPGAs comparative study for power consumption, noise cancellation, and real time high speed applications. J Softw Eng Appl 3(4):391
27. Ferlini F et al. (2012) Non-intrusive fault tolerance in soft processors through circuit duplication. In: Proceedings of the 2012, 13th Latin American test workshop (LATW), IEEE, 2012
28. Montenbruck O (2008) GNSS receivers for space applications. Lecture. In: ACES and future GNSS-based earth observation and navigation
29. Hoyt R, Voronka N, Newton T, Barnes I, Shepherd J, Frank SS, Slostad J, Jaroux B, Twiggs R (2007) Early results of the multi-application survivable tether (MAST) space tether experiment; SSC07-VII-8/048; 21st annual AIAA/USU conference on small satellites, 13–16 Aug 2007, Logan, UT, USA
30. Scholz A, König F, Fröhlich S, Piepenbrock J (2009) Flight results of the COMPASS-1 Mission. http://www.raumfahrt.fh-aachen.de/compass-1/download/COMPASS-1%20Flight%20Results.pdf