A parallel software-defined ultra-low-power receiver for a satellite message forwarding system

Raffael S. C. G. de Lima*, José Marcelo L. Duarte†, Diego V. Cirilo do Nascimento‡,
Reinaldo A. de Souza Filho*, Samuel Xavier-de-Souza*

*Federal University of Rio Grande do Norte, Natal-RN, Brazil
raffael@ufrn.edu.br,{reinaldo.souza,samuel}@dca.ufrn.br
†National Institute for Space Research, Natal-RN, Brazil
jose.duarte@inpe.br
‡Federal Institute of Rio Grande do Norte, Natal-RN, Brazil
diego.cirilo@ifrn.edu.br

Abstract—**Nanosatellites have become the standard solution for most space systems operating in Low Earth Orbit (LEO). However, this category of satellite imposes strong restrictions on the energy consumption of its subsystems due to the small size of its solar panels. This work presents a parallel software-defined multi-user Phase Shift Keying (PSK) receiver for a nanosatellite payload that will serve the Global Open coLlecting Data System (GOLDS), a message storage and forwarding system. For this, we chose the GAP8, an embedded multi-core RISC-V microprocessor. We use a parallel approach and dynamic voltage and frequency scaling (DVFS) to implement complex signal processing ensuring low power consumption and meeting the real-time operating condition. The receiver's input signals are 400 bps Machested encoded $\pm\pi/3-$PSK burst signals from terrestrial platforms, and the communication channel was modeled as AWGN with an independent flat fading per PSK signal. A MATLAB reference model was used for functional validation of the proposed implementation. Up to 12 signals can be decoded simultaneously requiring a maximum power consumption of 41 mW. The use of DVFS provided a maximum savings of 43% in dissipated power and 12% in energy consumption.**

Index Terms—**multi-user receiver, message storage and forwarding system, ultra-low power, multi-core architecture, parallel processing.**

I. INTRODUCTION

For applications that require the transmission of sensor data in remote areas, such as environmental monitoring, wildlife tracking, boat tracking, among others, the solution with the lowest structural cost is a message storing and forwarding system based on low earth orbit (LEO) nanosatellites. In such a system, the satellites act as intermediate stations. They receive messages from users, also referred to here as the platform transmitter terminal (PTT), and transmit them as soon as they are passing over a receiving station (RS). Finally, a distribution data center (DDC) offers a cloud service so that users have access to their platform's data. Figure 1 illustrates the components that compose such systems. Although a single satellite and a single RS are enough to provide the service worldwide, a large number of satellites is desirable to reduce the mean revisit time, the time elapsed between data uplink opportunities for the PTTs, and also some spread RSs to

increase the area where immediate data retransmission is possible.

Recently, this type of system has gained attention due to the use of LEO nanosatellites in the CubeSat standard, which has significantly reduced the developing and launching costs. Some systems such as ORBCOMM, Iridium, Argos-4, and more recently the Lacuna Space project use LEO satellites to compose their communication network. Newer systems have combined low-orbit satellites with terrestrial IoT (Internet of Things) networks. From this combination comes the concept of the Internet of Remote Things (IoRT) [1].
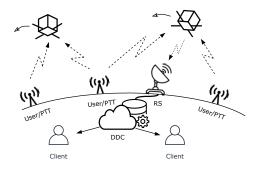

Fig. 1. Devices that make up the satellite message storage and forwarding system. [2]

The Brazilian Environmental Data Collection System (Sistema Brasileiro de Coleta de Dados Ambientais, SBCDA) [3], aims to collect data, such as temperature, humidity, wind speed, atmospheric pressure, wildlife observation, among others. The data collected is made available free of charge to clients registered in the system. The SBCDA is based on Argos-2 [4] and is maintained and operated by the National Institute for Space Research (Instituto Nacional de Pesquisas Espaciais, INPE). Observing its purpose, it has PTTs spread throughout the Brazilian territory. Currently, the satellites of this system use an analog transponder, making this a message forwarding system with limited coverage. The RSs decode

the retransmitted signals and make the data available to users through the DDC.

The Data Collection Subsystem (DCS) transponder [5] is the analog transponder used to relay the signals sent by PTTs to the RSs and is loaded on the satellites SCD-1 (1992), SCD-2 (1998), CBERS-4 (2014) and CBERS-4A (2019). This analog transponder has components with space qualification, which gives it high reliability, but its application is only possible on large satellites due to its volume, weight, and energy consumption. Carvalho et al. [6] proposed the renovation and expansion of the SBCDA constellation using nanosatellites in the CubeSat standard and cited a CubeSat compatible digital transponder based on commercial-off-the-shelf (COTS) components to replace the analog one, but this digital transponder still has a high energy consumption for nanosatellites.

The CubeSat standard defines a unit (1U) as a cube with a volume of 1 liter, having 10 cm edges and a mass between 1 and 1.33 kg. It is possible to combine units to have satellites of the 1.5U, 2U, 3U, 6U, and 12U, for example. This standard has been gaining adherents in the industry due to the low cost of production, development and launch. The use of COTS components eliminates the risk of embargo on the commercialization of components for space applications. Although the reduced dimensions promote a low launch cost, they also cause a limitation in the power generation to feed the subsystems that make up the satellite. For the 1U size, for example, the platform manufactured by ISIS provides an average payload power of 400 mW and a peak power of 2 W [7].

Duarte [8] presented the Environmental Data Collector (EDC) a multi-user receiver compatible with CubeSat. The EDC has a lower consumption than the digital transponder, about 1.2 W, making it possible to embed it in nanosatellites of size 1U [9]. By performing the on-board signal decoding, the EDC eliminated the need for a dedicated transmitter, since the decoded data can be transmitted to the RSs through the existing telemetry channel on the satellite. Moreover, the onboard signal decoding enables global coverage. This was one of the features proposed for the reformulation of the SBCDA into the Global Open coLlecting Data System (GOLDS) [2], [10], a message storage and forwarding system.

Observing the demand for COTS processors for space application Di Mascio et al. [11] presented a study of how the RISC-V architecture can contribute to the development of components for this market. The authors describe the types of processors that the aerospace industry demands and the proposed solutions based on the open-source RISC-V hardware architecture. The article also deals with the architecture of single-core and multi-core processors, the performance gain with increasing cores and the types of parallelism according to the concepts of Flynn's taxonomy. De-RISC NOEL-V [12] and CEVERO [13] are recent projects that seek to be inserted in the aerospace industry and that are based on the RISC-V architecture. Both projects aim to offer a complete platform composed of hardware and software for future developments in space and aeronautical applications.

In this context, this work presents an implementation of an multi-user receiver algorithm for use with nanosatellites. For this implementation of the algorithm proposed by Duarte et al. [14], fixed-point C was used. The application was embedded in the GAP8, an IoT application processor based on the PULP platform which itself implements an extended version of the open-source RISC-V instruction set [15]. PULP platform offers levels of code parallelism and controllable operating points of Dynamic Voltage and Frequency Scaling (DVFS) to allow energy-consumption optimization. The analysis of the obtained results is performed in terms of compliance with the real-time operation restriction, energy consumption and power dissipation. The tests will be performed from the processing of stimulus loaded in the processor's data memory. A MATLAB model of the decoder was used as a reference for the development of the fixed-point C code.

Our proposal aims to meet an expectation of increased energy efficiency in a critically energized environment. Allowing better use of the battery, preserving its useful life on the nanosatellite. Furthermore, the modularity of the RISC-V architecture combined with its open and free nature facilitates the integration of this project into a processor suitable for the space environment. Which, in general, increases the reliability of the application. Specifically, we propose here an alternative for implementing a receiver in a hybrid architecture.

- This work presents a practical implementation of a receiver that meets the requirements and attend the demand of a real satellite communication system.
- The software-based receiver was implemented in fixed-point C, ensuring a simpler architecture than presented in the EDC [8]. The software-based solution facilitates updates of the receiver to meet future system specification while maintaining the computational complexity of the algorithm proposed in Duarte, et. al. [14].

This document is organized as follows. Section II discusses the works related to our proposal. Section III presents the system description, the PTT signal characteristics and the multi-user receiver algorithm. Section IV presents the architecture of the GAP8 and the parallel approach for implementing the algorithm in this architecture. In Section V we present and discuss the experimental results. Finally, in Section VI we have the conclusions.

II. RELATED WORK

Escrig et al. [16] addressed the problem of multiple access interference caused by multi-user receivers in the Argos-2 system. The authors presented the matched filter used for demodulation of received signals and proposed several techniques such as the Maximum Likelihood (ML) detector, the successive interference cancellation receiver (SIC), among others. The authors analyze the results by the bit error rate (BER) obtained by the presented algorithms. For the same system, Fares et al. [17] compares the ML detector with the SIC receiver and analyzes the parameter estimation for this receiver. The algorithm chosen for this work has less computational complexity and less use of memory resources.

Rae [18] made a status study of the SBCDA, estimated PTT signal power levels at the satellite receiver and also the number of signals coexisting in a time instant, and proposed a PTT signal detector by spectral analysis for the SBCDA/Argos-2. Duarte et al. [14] proposed the detection and decoding algorithm that was used in its implementation on the EDC and that will be used in the development of this work. The authors present the frame error rate (FER) and the effect of the average number of coexisting signals in the correct detection rate. The performance evaluation of detection and decoding algorithms is not part of the scope of this work, which will focus on the performance of the GAP8 implementation.

Duarte [8] presented the EDC implementation architecture which details the implementation partition between the FPGA and the microcontroller. At the FPGA core, the were implemented the calculation of samples of the spectrum, used for signal detection, and a sequential demodulator of signals from multiple users. The microcontroller performs the control of the signals presence detections, demodulation and bit detection processes in addition to the external communication of this subsystem with the on-board computer. It is easy to see that the computational effort was the factor that defined this division of tasks.

An implementation in a hybrid architecture of a multi-user satellite receiver with OQPSK/TDMA modulation is presented in [19]. The authors use four multi-core DSP in conjunction with two Xilinx Kintex FPGAs and four Xilinx Virtex FPGAs to process 12 channels in split-layer processing. Performance analysis was performed by measuring the bit error rate in addition to complying with the real-time processing criterion. However, there was no measurement of energy consumption.

An FFT parallel implementation is presented in [20], pointing the single instruction and multiple data (SIMD) processor arrays are promising to provide high processing performance. As the analysis of the performance, the authors compare the parallel approach with sequential DSP implementation, the results show the parallel approach improves the performance and energy efficiency. In our implementation, we use spectral analysis to detect the presence of the signal, we use a single program multiple data (SPMD) technique and offer a different parallel approach for FFT implementation.

The PULP platform is introduced in [21]. This platform aims to meet applications that require significant performance but have energy consumption restrictions. For this, PULP uses RISC-V cores exploring data parallelism or task parallelism and, at the same time, controllable operation points, making it possible to adjust the tradeoff between performance and energy consumption. A convolutional neural network was used as proof of performance. The analysis compared the performance of the proposed platform with other multi-core platforms for embedded computing. Existing works have evaluated the energy efficiency of the PULP platform by implementing applications with energy consumption restrictions and real-time processing [22], [23]. The control of operational points was previously studied by [24]. The authors propose that dynamic control of supply voltage and operating frequency at runtime is a key technique to reduce energy consumption and dissipation.

The use of COTS components in aerospace environments has become common due to the popularization of the CubeSat platform. CubeSats offer cheaper designs with little volume and mass, which lowers the launch cost. Now the industry is preparing to make these components more reliable and resistant to space radiation with the development of hardware extensions that can verify and correct the processing in case of failures. The use of RISC-V architecture in the aerospace industry comes to cover the demand for a flexible and scalable hardware platform. This work presents an application embedded in an architecture based on the next generation of aerospace industry processors. As far as is known, there is no public work available on the parallel implementation of ultra-low-power satellite multi-user receiver.

III. BACKGROUND: SYSTEM DESCRIPTION

In this section, we address the PTT signal characteristics and the detection and decoding algorithms for these signals.

A. Input Signal Model

The PTTs transmit messages periodically, with a minimum interval of 60 seconds between transmissions, using short signals with a minimum duration of 360 ms and a maximum of 920 ms. Each PTT is configured with a fixed transmission frequency within a range of 60 kHz centered in 401.635 MHz. At the satellite receiver, the signals undergo a variable Doppler shift with approximate limits of ±9 kHz that helps to further spread the signal in the frequency domain. PTT signal is specified as 400 bps $\pm\pi/3$-PSK Manchester (bi-ϕ-L) coded [4], which results in a bandwidth of approximately 1.6 kHz, and is further defined by a pure carrier period of 160 ms, followed by a 24-bit synchronization bit pattern (FFFE2F$_{hex}$), and finally a user message of variable size between 7 to 35 bytes. The user message contains a message length code (4 bits), the PTT identification (20 bits), and the user data of variable size. The user data usually carries data from sensors (temperature, humidity, wind speed, among others) installed in the platforms.

The satellite receive signal is therefore composed of the sum of multiple PTT signals in passband in addition to the channel noise. Each of these signals has its time shift, power, carrier frequency, carrier phase, and user message, all randomly defined within the valid range. This was modeled using the following equation:

$$r(t) = \Re\left(\sum_{p=1}^{P}\sqrt{\frac{2Ep}{T}}s_p(t-\tau_p)e^{i\theta_p(t-\tau_p)}\right) + n(t), \quad (1)$$

in which P is the number of PTT signals present during the observation period, $s_p(t)$ is the baseband PTT signal of index p, τ_p is the starting time of the p-th signal, E_p is the bit energy, T is the bit period, $\theta_p(t)$ is the carrier phase signal, and $n(t)$ is the additive white Gaussian noise (AWGN).

PTT baseband signal is modeled as:

$$s_p(t) = \begin{cases} e^{im_p(t-T_C)}, & 0 < t < T_p, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

where $m_p(t)$ is equal to $0 \; rad$ for $t < T_C$ and alternates between $\pi/3$ and $-\pi/3 \; rad$ for $t \geq T_C$ according to the transmitted data. T_C is the pure carrier period and T_p is the total signal duration of the p-th PTT signal. In Rae [18], the dB ratio between the maximum received signal power Smax and the noise power density was calculated from the parameters of users' transmitter power, transmitter antenna gain, link loss, receiver antenna gain, and losses due to radio receiver circuit. The range of possible values is $S_{max}/N_0 = 61.25$ dB and $S_{min}/N_0 = 43.59$ dB. In [14], the authors calculate the S_{min}/N_0 to the receiver sensibility at 40 dB. This value was calculated for a bit error rate for a BPSK signal of 10^{-5} under AWGN conditions and considering losses due to the residual carrier and losses due to the radio receiver circuit.

Since multiple PTT signals usually coexist in time at the receiver, a multi-user receiver is necessary. The authors in [14], [18] considered an average of 7 and 5 PTT signals coexisting in time, respectively.

B. Multi-User Receiver Algorithm

The multi-user receiver algorithm proposed by Duarte et al. [14] processes the input signal in segments of 10 ms seconds at a time. The authors defined a sample rate of 128 kilo samples per second, leading to 1280 samples per segment. For each input segment, Signal Detection and Decoding steps are executed. In the Signal Detection step, a search for new user signals is performed using the spectrum of the input segment. For each detected new signal, a single-user decoding process is instantiated. In the Decoding step, each active single-user decoding process is updated providing the input segment as input. It is important to note that a single-user decoding process should remain active for multiple input segments since a user signal lasts longer than 10 ms. Therefore, after processing the input segment, a decoding process can remain active to continue its processing in the next input segment, or finish, with success or not. If a decoding process finishes, an output structure is generated containing the signal's receiving frequency and power, along with the decoded message. In Figure 2, we can see a flowchart of the multi-user receiver algorithm.

1) Signal Detection: The spectrum of the input segments (1280 samples) is computed using Equation 3,

$$X_k = \left| \frac{1}{N} \sum_{n=0}^{N-1} r_n(-1)^n e^{-\frac{2\pi ikn}{M}} \right| \quad (3)$$

where $k \in \{0, 1, \ldots, M-1\}$ and $r_0, r_1, \ldots, r_{N-1}$ are the samples from the input segment, $N = 1280$ is the input segment length, $M-N = 768$ is the length of the zero-paddings applied before computing the 2048-point DFT and $|.|$ is the absolute value operation applied in DFT result.
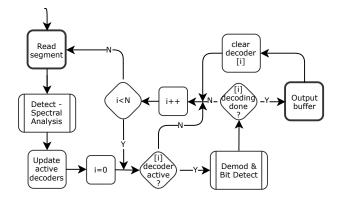

Fig. 2. Multi-user receiver block diagram.

After computing the spectrum of the current segment, a signal detection loop takes place. This detection loop looks for the spectrum bin with the highest value that obeys the following criteria:

i. Its value is above the decoding sensibility level.
ii. Its frequency position distances more than 3.2 kHz from the setup frequency of all active single-user decoding processes.
iii. It also succeeds to obey two previous criteria at the end of the detection loop of the previous input segment.

If the detection process finds a bin that obeys all these criteria, this bin is taken as a new signal indicator, and the detection process activates a single-user decoder providing the frequency ($\hat{f}$) and magnitude ($\hat{a}$) of this bin as setup parameters. The loop then returns to perform a new search, considering the newly activated decoding process. This process is continued until no more signal is found or there is no single-user decoder available.

2) Single-User Decoding: A phase-locked loop (PLL) presented in Figure 3 is used to demodulate PTTs signals. The demodulator receives the detected signal's frequency and magnitude as configuration parameters. The frequency sets the PLL central frequency and the magnitude sets a variable gain amplifier that normalizes the signal amplitude. The PLL phase error signal is generated using a vector rotation followed by a matched filter with decimation and a cartesian to the polar converter. The phase error signal passes through the PLL Loop Filter (LF) that adjusts the local oscillator's frequency. The parameters of the LF were adjusted considering a maximum phase error of 10°. The Zero Order Hold (ZOH) is used to upsample the LF output back to the input sample rate. The accumulator converts the input discrete-time frequency into a discrete-time phase.

In order for the transmitted symbol to be identified, the optimal demodulated signal sampling instant must be calculated. The chosen technique was the continuous-time filter and square timing recovery presented by Oerder [25] and consists of raising the signal to the square and shifting it in frequency, to bring the spectral component of the symbol to zero hertz. A low-pass filter is applied in the frequency-shifted signal and
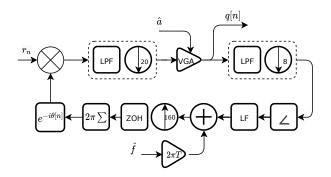
Fig. 3. PLL flowchart.

the ideal sampling time is obtained from the resulting signal phase.

The calculation of the sampling time offset (τ_e) in multiples of symbol period is defined in Equation 4,

$$\tau_e\,[n_0] = \frac{1}{2\pi} \arg\left\{ \frac{1}{\sigma\epsilon} \sum_{n=\epsilon n_0}^{\epsilon n_0 + \epsilon\,\sigma - 1} |q[n]|^2 \cdot e^{-\frac{i\pi n}{\epsilon}} \right\} \qquad (4)$$

in which $q[n]$ is the symbols of the demodulated signal; $\epsilon = 8$ is the oversample rate, the ratio between sample rate and symbol rate; $\sigma = 16$ is the moving sum filter length in number of symbols. A buffer stores 2ϵ samples promoting fixed data rate. The symbol sampler produces an output sample for every ϵ input samples, through linear interpolation in conjunction with a decimation.

Finally, the bit detection block implements Manchester decoding, in this type of encoding two consecutive symbols represent a bit. A low-to-high transition represents a bit 0 and a high-to-low transition represents bit 1. In addition, there is bit synchronization by comparing the decoded bits with the 24-bit synchronization bit pattern at the beginning of the decoding of the message. From there, the signal is decoded until the message size is reached.

IV. ARCHITECTURE

In this section, we present the GAP8 architecture and the parallel approach used to implement the software-defined receiver.

A. Hardware Architecture

To meet the criteria of real-time processing and maximum energy efficiency, it is necessary that the implementation of the multi-user receiver be optimized for the GAP8 architecture. GAP8 is an IoT application processor by GreenWaves Technologies [15] and it is based on the open-source PULP platform. The implementation was carried out using a GA-Puino, GAP8 development board.

The System-on-Chip incorporates nine RISC-V cores, one of which is the Fabric Controller (FC) that serves the microcontroller unit (MCU) and is responsible for control, communication and security. The other eight cores are part of the cluster (CL) that can be used for vectorized and parallelized

algorithms. The MCU has a GPIO bus, supports I2C, I2S, UART serial communication, among others. The L2 memory is shared with all devices and has four banks with 128 KB each, totaling 512 KB. The maximum operating frequencies for the GAP8, according to the GAP8 datasheet, are 250 MHz in the FC and 175 MHz in the cluster. To ensure efficiency control, all cores and peripherals interfaces have power switches in addition to adjustable voltage and frequency. Existing DC/DC regulators and clock generators, through frequency-locked loop (FLL), allow the application's processing requirements to be met with the lowest power consumption [15].

The cluster cores are enhanced for digital signal processing and embedded deep inference. All cores share access to an L1 memory area (64 kB) and instruction cache (4 kB). Several Direct Memory Access (DMA) units allow for autonomous, fast, low-power transfers between memory areas. A memory protection unit is included to allow applications to run safely on the GAP8. The L1 memory is banked and connected to the cluster cores via a logarithmic interface that is sized to provide single-cycle access in 98% of cases. The combination of instruction memory in the cluster and high-speed shared data provides an ideal memory architecture for executing code that implements parallelized algorithms [15]. Figure 4 shows the main building blocks of a GAP8 architecture.
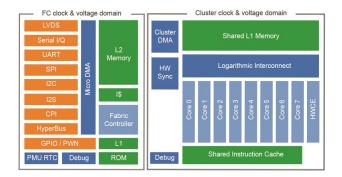

Fig. 4. Building blocks of GAP8 architecture. [15]

B. Software Architecture

The main strategy is to load the signal to be processed and each core to manage its processing. By making a SIMD version of the receiver by data parallelism, the GAP8 has loop vectoring instructions that can be used to improve performance. GAP8 has only one instruction cache to manage the cluster. This makes task parallelism considered inefficient for this processor. It would need to load a large number of instructions into its cache, which would considerably affect performance.

The proposed architecture makes use of a variable number of cores in the cluster according to the number of concurrent PTT signals processed. Data parallelism occurs where more processing is needed. In the detection module, this occurs when calculating the FFT of 2048 points. In the decoding module, it occurs in the demodulation of the signal.

The input signal is saved in the L2 memory, the sample sequence of each time window corresponding to 10 ms is

transferred to the L1 memory in the processing stage in the cluster. All processing is performed on the cluster.

In the parallel implementation of the FFT, the approach of grouping the butterfly calculations in the core in a proportional manner by stage was used. This approach is possible because all butterfly calculations per stage can be performed independently. At the end of each stage, the results are updated in the respective memory locations. This provides the maximum scope of parallelism. Butterfly groupings are performed at the beginning of each stage where a barrier occurs. In the end, each core calculates 256 butterflies in each of the 11 stages. To optimize the process, the twiddle factors were placed on a look-up table. The core 0 processes the Detection Loop, Update the active decoders, and instantiates the single-user decoders for each new signal founded.

In the Single-User Decoding module, the processing fork is performed in the signal demodulation operation. The use of cores in the cluster will vary depending on the number of detected signals. The update of active decoders is done sequentially in core 0. And dynamically from the amount of active decoders the fork is executed. As long as the number of active decoders is less than or equal to the number of cores in the cluster, each active core will only process one signal. When the number of active decoders is greater than the number of cores, the cores can process up to two signals sequentially. Memory limitation prevents processing of more than 12 signals simultaneously. Figure 5 illustrates the processing of an input segment when 12 decoders are active, D_n represents the demodulation processing of each segment. The Bit Detection is performed on core 0, if there is an end-of-operation in any single-user decoder a package with the frequency, amplitude, and the message decoded is loaded to output buffer on the L2 memory.

Fig. 5. Parallel approach of the multi-user receiver.

V. EXPERIMENTAL RESULTS

To evaluate the receiver, we loaded into the GAP8 data memory digitized signals from baseband PTTs, as well as the expected results generated from a model developed in MAT-LAB. Measurements are performed during signal processing

with a size equivalent to a minimum transmission size of the PTT signal, corresponding to 360 ms. This signal is composed of a variable number of concurrent PTT signals (nConcPTT), the frequencies were spaced so that there was no spectral overlap. The initial phase and the Doppler rate were randomly generated respecting the intervals of $-\pi$ to $+\pi$ rad and -120 to $+120$ Hz/s, respectively. Furthermore, the generated signals had power per noise density of $S_{test}/N_0 = 43$ dB. For files with more than one user signal, the frequencies were spaced so that there was no spectral overlap. We evaluate performance in terms of compliance with time constraints and energy consumption, for this we use the MAGEEC board [26] in conjunction with pyenergy firmware and host software. Energy consumption, run time, and average power dissipation are reported to the host.

Initially, the receiver was optimized for GAP8 in the FC using L2 memory. The processing time of the single-user receiver in sequential approach did not result in a real-time operation even using the highest allowed clock rate of 250 MHz. However, when running the same test in the CL using the L1 memory, the results obtained were promising, at a lower clock rate of 175 MHz, real-time processing was achieved for the single-user receiver.

The runtime measurements performed in the sequential implementation of the receiver algorithm showed that the FFT calculation corresponds on average to 80% of the total processing time of the detection module. With the parallel implementation of the FFT algorithm, it was possible to observe a reduction of 70.2% on average in the detection module execution time for the processing of each input segment.

In order for the fork to be executed, it is necessary to pass as a parameter a data structure that will be used as memory for each of the active cores. Because of this, there was a restructuring of the data architecture to reuse memory. This resulted in an improvement in the execution time of the decoding module, even for processing one signal. Table I shows the results of the average runtime measurements per segment (1280 samples) of the detection, decoding and total modules for the complete transmission of a PTT signal. The dissipated power and energy consumption measurements are for the total 360 ms signal transmission. The measurements consider the current drawn by the entire GAP8 chip. It is possible to see that although the average power increases due to the use of the cluster, the total energy consumed decreases due to the decrease in processing time.

TABLE I
RESULTS OF TIME AND ENERGY CONSUMPTION MEASUREMENTS PER WINDOW IN DIFFERENT APPROACHES.

	Time (ms)			Power (mW)	Energy (mJ)
Approach	Detect	Decoding	Total	Average	Total
Seq. FC_L2@250MHz	6.6	3.7	10.4	12.9	4.97
Seq. CL_L1@175MHz	4.7	3.4	8.1	24.3	7.28
Par. CL_L1@175MHz	1.4	2.8	4.2	29.7	4.48

Figure 6 presents the processing time results of complete signal transmissions for the Parallel, Sequential CL and Se-

quential FC implementations. To carry out these tests, a variable number of simultaneous PTT signals (nConcPTT) were used in each transmission. The measurements taken are associated with the maximum operating frequencies (250/175 MHz) and the 1.2V supply voltage. The highlighted red line represents the real-time operating threshold. Comparing the implementations, it is possible to notice that the parallel approach makes it possible to process up to 12 simultaneous user signals in real time. While the sequential CL implementation exceeds the real-time operation threshold when nConcPTT is equal to 2.
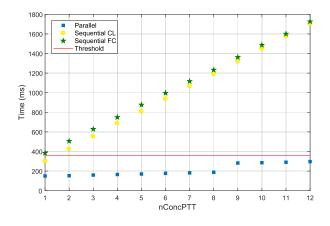

Fig. 6. Total processing time per number of concurrent PTT signals for minimum duration PTT signals. Comparison between sequential and parallel approaches.

The margin to meet the real-time constraints presented by the parallel approach allows the use of a lower operating frequency, consequently allowing a reduction in the processor's supply voltage. This reduces the processor's peak power consumption. Furthermore, it decreases the difference between dynamic and static power consumption. This is especially seen when the number of active single-user decoders is equal to or less than the total number of cores in the CL.

Figure 7 shows the different results of the time measurements considering the operation at a maximum frequency and voltage (MAX) and the operation at a variable voltage and frequency (VF). In VF, the voltage and frequency adjustments were performed according to boundaries of GAP8 datasheet and so that the processing time of a signal was at a distance of 15% from the threshold value, which corresponds to 300 ms. With the frequency reduction, it was possible to decrease the supply voltage to 1 V in cases where nConcPTT is equal to or less than eight. As a result of the variation in operating points, we had a maximum reduction of 43.4% in average power dissipation and 12% in energy consumption. This occurs when the nConcPTT is equal to the number of cores in the CL, as shown in Figure 8. The result is more relevant since the average number of coexisting signs is within the range of greatest savings.

From the EDC [8] consumption data provided by INPE, a comparison can be made with the power consumption mea-
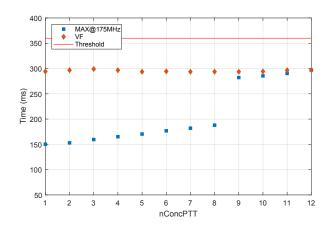

Fig. 7. Processing time per number of concurrent PTT signals for minimum duration PTT signals. Comparison between parallel approaches.
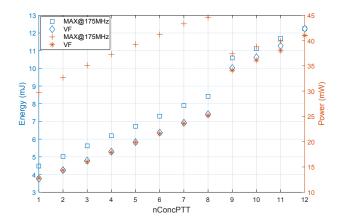

Fig. 8. Energy consumption and Power to process concurrent PTT signals for minimum duration PTT signals. Comparison between parallel approaches.

sured by the implementation proposed by this work. The dynamic and static power consumption of the EDC was extracted from the SmartPower software provided by the manufacturer of the FPGA used. To calculate the power consumption values, a pre-synthesis simulation '.vcd' file was used. With this file, the software identifies the existing frequency domains and calculates the consumption according to the probability of variation in the logical level of the hardware resources used. The total usage power extracted was 168.8 mW, with a static power of 15 mW as shown in the Figure 9. With this result, the implementation proposed by this work presents savings of approximately 75.7% when there is processing of 12 simultaneous signals.

VI. CONCLUSION

This paper presented a parallel software-defined implementation of a multi-user PSK receiver in a GAP8 Parallel Ultra-Low-Power processor for a nanosatellite payload that will serve the Global Open coLlecting Data System (GOLDS), a message storage and forwarding system. The use of the parallelism approach to achieve low energy consumption en-

```
Design:                 edc_m2s
Family:                 SmartFusion2
Die:                    M2S025
Package:                484 FBGA
Temperature Range:      IND
Voltage Range:          IND
Operating Conditions:   Typical
Operating Mode:         Active
Process:                Typical
Data Source:            Production


Power Summary
+--------------+-----------+------------+
|              | Power (mW) | Percentage |
+--------------+-----------+------------+
| Total Power  |  168.798  |   100.0%   |
| Static Power |   15.315  |    9.1%    |
| Dynamic Power|  153.482  |   90.9%    |
+--------------+-----------+------------+
```

Fig. 9. EDC power usage exported from SmartPower.

sures high processing efficiency. Up to 12 signals can be decoded simultaneously. The implementation of the receiver in the GAP8 processor was validated through the processing of stimuli containing signals from multiple users saved in its data memory. The decoded bits were compared with the bits used to generate the stimuli. A MATLAB model was used as a reference for the development of the algorithm in the processor. The results showed that the use of DVFS can provide savings of 43% in the dissipated power and 12% in the energy consumption. The fixed point C implementation proposed by this work has a simpler architecture than the one presented by EDC, which has a hybrid SoC FPGA architecture. Futhermore, the results showed that the implementation proposed by this work presents savings of up to 75.7% in power consumption when comparing with EDC.

References

[1] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting internet of remote things," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 113–123, 2016.

[2] J. M. L. Duarte and F. J. T. Vidal, "Global open collecting data system (GOLDS): Proposta de requisitos para o desenvolvimento da especificação do sistema," 2021.

[3] W. Yamaguti, V. Orlando, and S. d. P. Pereira, "Sistema brasileiro de coleta de dados ambientais: status e planos futuros," in *XIV Simpósio Brasileiro de Sensoriamento Remoto - SBSR*. INPE, 2009.

[4] "Argos - worldwide tracking and environmental monitoring by satellite," Website. [Online]. Available: http://www.argos-system.org/, 5 2019.

[5] Omnisys, "Subsistema de coleta de dados," Datasheet. [Online]. Available: https://www.omnisys.com.br/data-seet/subsistema-DCS.pdf, 2010.

[6] M. J. M. de Carvalho, J. S. d. S. Lima, L. d. S. Jotha, and P. S. de Aquino, "Conasat - constelação de nano satélites para coleta de dados ambientais," in *XVI Simpósio Brasileiro de Sensoriamento Remoto*. INPE, 2013, pp. 9108–9115, available: http://urlib.net/rep/3ERPFQRTRW34M/3E7GL7J.

[7] ISIS, "Isis 1u cubesat," Website. [Online]. Available: https://www.isispace.nl/wp-content/uploads/2018/07/ISIS-1U-CubeSat-Brochure.pdf, 7 2018.

[8] J. M. L. Duarte, "Cubesat payload for environment data collection," Oral presentation at the UN/Brazil Symposium on Basic Space Technology. [Online]. Available: https://www.unoosa.org/documents/pdf/psa/activities/2018/Symposium_Brazil_BSTI/presentations/oral/S41_02_Jose_Duarte.pdf, 9 2018.

[9] INPE. "Environmental Data Collector (EDC)". [Online]. Available: http://www.inpe.br/nordeste/projetos/edc.php

[10] F. Mattiello-Francisco, M. J. M. De Carvalho, M. A. F. dos Santos, and G. P. Garbi, "Fostering environmental data collection with golds constellation," Oral presentation at the UN/Brazil Symposium on Basic Space Technology. [Online]. Available: https://www.unoosa.org/documents/pdf/psa/activities/2018/Symposium_Brazil_BSTI/presentations/oral/S41_01_Fatima_Mattiello-Francisco.pdf, 9 2018.

[11] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, "Leveraging the Openness and Modularity of RISC-V in Space," *Journal of Aerospace Information Systems*, vol. 16, no. 11, pp. 454–472, 2019, doi: 10.2514/1.i010735.

[12] J. Andersson, "Development of a noel-v risc-v soc targeting space applications," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 66–67, available: https://doi.org/10.1109/DSN-W50199.2020.00020.

[13] I. Silva, O. do Espírito Santo, D. do Nascimento, and S. X. de Souza, "Cevero: A soft-error hardened soc for aerospace applications," in *Anais Estendidos do X Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. Porto Alegre, RS, Brasil: SBC, 2020, pp. 121–126. [Online]. Available: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13100

[14] J. M. L. Duarte, R. S. C. G. de Lima, V. S. Ramos, and M. J. M. de Carvalho, "A multiuser decoder based on spectrum analysis for the brazilian environmental data collecting system," *International Journal of Satellite Communications and Networking*, vol. 39, no. 2, pp. 205–220, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1381

[15] G. Technologies, "Gap8 manual," [Online]. Available: https://greenwaves-technologies.com/manuals/BUILD/HOME/html/index.html, 2020.

[16] B. Escrig, F. Fares, M.-L. Boucheret, T. Calmettes, and H. Guillon, "Multi-user detection for the argos satellite system," *International Journal of Satellite Communications and Networking*, vol. 33, no. 1, pp. 1–18, 2015.

[17] F. Fares, B. Escrig, M.-L. Boucheret, T. Calmettes, and H. Guillon, "Non data aided parameter estimation for multi-user argos receivers," in *Wireless Telecommunications Symposium 2012*. IEEE, 2012, pp. 1–5.

[18] J. P. Rae, "Detector de sinais para os satélites do sbcd usando análise espectral digital," Master's thesis, Technological Institute of Aeronautics (ITA), 2005.

[19] K. G. Dileep, P. Laxmaiah, S. N. Kumar, P. Goutam, S. V. Prasad, M. Soundarakumar, and V. Tyagi, "Multi-core DSP-based implementation of variable data rate OQPSK/TDMA satellite receiver," *International Conference on Electronics, Information and Communication, ICEIC 2018*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.23919/ELINFOCOM.2018.8330580.

[20] J. Choi, J. Kim, and C. H. Kim, "Parallel implementation of the FFT algorithm using a multi-core processor," *2010 International Forum on Strategic Technology, IFOST 2010*, pp. 19–22, 2010, doi: 10.1109/IFOST.2010.5668106.

[21] F. Conti, D. Rossi, A. Pullini, I. Loi, and L. Benini, "Energy-efficient vision on the PULP platform for ultra-low power parallel computing," *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, pp. 6–11, 2014, doi: 10.1109/SiPS.2014.6986099.

[22] M. Magno, A. Ibrahim, A. Pullini, M. Valle, and L. Benini, "Energy efficient system for tactile data decoding using an ultra-low power parallel platform," *Proceedings - 2017 1st New Generation of CAS, NGCAS 2017*, pp. 17–20, 2017, doi: 10.1109/NGCAS.2017.56.

[23] M. Eggimann, J. Erb, P. Mayer, M. Magno, and L. Benini, "Low Power Embedded Gesture Recognition Using Novel Short-Range Radar Sensors," *Proceedings of IEEE Sensors*, vol. 2019-October, no. I, pp. 90–93, 2019, doi: 10.1109/SENSORS43011.2019.8956617.

[24] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," *Operating Systems Review (ACM)*, vol. 35, no. 5, pp. 89–102, 2001, doi: 10.1145/502059.502044.

[25] M. Oerder and H. Meyr, "Digital filter and square timing recovery," *IEEE Transactions on communications*, vol. 36, no. 5, pp. 605–612, 1988, doi: 10.1109/26.1476.

[26] S. Hollis, "The MAGEEC energy measurement board," Website. [Online]. Available: http://mageec.org/wiki/Power_Measurement_Board, 5 2013.