ISSN 2320 - 2602

Volume 12 No. 7, July 2023

International Journal of Advances in Computer Science and Technology

Available Online at http://www.warse.org/IJACST/static/pdf/file/ijacst011272023.pdf

https://doi.org/10.30534/ijacst/2023/011272023

FPGA as an Accelerator in Satellite Ground Station for Data Reception and Processing

Dr. G.Prasad
Scientist "SG", NRSC/ISRO, Hyderabad, India
gurram_p@yahoo.com

Received Date : May 26, 2023 Accepted Date : June 17, 2023 Published Date : July 07, 2023

ABSTRACT

In this paper a data acquisition system which is a combination of hardware and software modules provide solution for high frequency data acquisition and processing for the upcoming satellites. The data acquisition system consists of FPGA based hardware for high frequency satellite data reception, preprocessing along with software modules for few elements of higher level processing thus providing near real time processed output products. The demodulator which receives the Intermediate Frequency data demodulates it, can also archive data in the raw form without any preprocessing done on it. But with the in house developed hardware presented in this paper, it enables us to receive and process the demodulated signal from the demodulator on this hardware and transfer to the system housing the hardware, thereby enabling real time data processing of data and structured data archival on system for further processing in near real time. Thus the FPGA provides the service of data acquisition and processing for all the missions in the ground station before it is archived for permanent and structured data. A new high frequency data acquisition system designed to receive data at 2.88 GHz coming from satellites are encoded, encrypted and compressed which are decoded, decrypted and decompressed in the hardware using the data acquisition system. The FPGA hardware, software and server architecture enable us to receive and process high frequency satellite data, such as those transmitted by Cartosat-2S, Cartosat-3 and the upcoming NISAR.

Key words: ASIC, CPU, Cortex HDR, FPGA, PCIe, NISAR, RAID.

1. INTRODUCTION

High frequency satellite data, involves large processing to be done because of the volume and complexity of the data. Several new applications have evolved to process this data. For example, the RS decoding, CCSDS decompression can process data faster by utilizing hundreds of cores simultaneously. These tools, however, are processor intensive and the CPU can become a bottleneck [1]. Using FPGAs, for these processing algorithms can implement these specific applications at a hardware level significantly increasing the system's overall performance. An important feature of FPGA services is the need for high-speed data exchange interfaces between hardware platform and software. Also, a considerable part of such services is formed class of tasks that requires the ability to process large amounts of data inside the integrated circuit with a small time delay. FPGAs have proven to be one of the preferred digital platforms for implementing high-performance computing. FPGA is a concurrent processing device that can implement any function that can be executed on a processor. FPGAs have a lot of resources that can be programmed and configured to implement any custom architecture and achieve almost any level of concurrency, resulting in improved performance with less power dissipation. To realize the benefits of accelerating software one should pay attention to the acceleration of the large computing resources of an application at the hardware level. Today's CPUs are evolving to contain more and more cores, but the bandwidth to external memory is not growing at the same pace of as this multi-core computing power. FPGAs can relieve the CPU data access bottlenecks by providing compression, filtering, and de-duplication functions. With FPGAs, users can perform real-time processing such as derandomization, decryption, decoding and decompression, and can complete the processing in near real time, one or more FPGAs can be utilized for the purpose.

Without the promise of ever-faster central processing units (CPUs) at a fixed power consumption, users have been forced to search elsewhere for solutions to their ever-growing computing needs. Some improvements in processor performance have come from the advent of multi-core processors and multi processors. However, there is growing interest in alternative computing architectures, such as graphics processing units (GPUs), field-programmable gate arrays (FPGAs), and application-specific integrated circuits(ASICs). Heterogeneous computing denotes systems which make use of more than one type of computing architecture, typically a CPU and one of the alternative architectures such as graphics processing units (GPUs), field-programmable gate arrays (FPGAs), and application-specific integrated circuits(ASICs) [2]. The alternative architecture is typically referred to as the “coprocessor” or “accelerator.” The advantage of this computing paradigm is that each algorithm can be run on the best-suited architecture.

2. ARCHITECTRE FOR USING FPGA AS AN ACCEERATOR

The standard FPGA as a service platform consists of three main components, i.e., the computer system, the FPGA, and the software utility [3]. The utility consists of tools, libraries, and drivers required developing FPGA’s hardware configuration files and facilitate efficient communication between the host and the FPGA. FPGA Setup and Initialization: At power-up, the initialization bit stream stored in flash memory configures PCIe IP block, memory blocks, and other required control logic. This initial configuration of the FPGA is required for the host system to identify FPGA as a valid PCIe peripheral device and register it with a unique ID during PCIe.

Hardware Configuration File and Host Application: After the initialization phase, the peripheral FPGAs are visible and accessible through the host. The user configures the FPGA with the partial reconfiguration bit stream consisting of the custom hardware. On the host side, a user has the custom software application that interacts with FPGA for performing the required function i.e. APIs to interact with FPGA. These APIs provide high-level abstractions of the communication protocols, which then invokes device drivers implementing low-level communication protocols to interact with the FPGA.

Security and Authenticity Checks: The user’s hardware accelerator is specifically a bit stream. This bit stream needs to be compliant with the FPGA’s base design (Initialization bit stream used at power-up). Therefore, the FPGA’s configuration manager performs compatibility tests before programming the FPGA with the user’s bit stream.

3. SATELLITE DATA ACQUISITION AND PROCESING IN THE PAST

Earlier Cortex modulator and demodulator was the solution for high data rate reception in X and Ka-bands. It can handle high speed demodulation of up to six channels with an aggregate bit rate as high as 10Gbps. With its flexible satellite check-out interfaces, it will make mission qualification easier and faster. Enabling a wideband RF recording and reproducing mode (2.4 GHz bandwidth) on top of its demodulation capabilities, Cortex modulator and demodulator ensures we will not lose any critical data and proves to be a comprehensive tool for ground station readiness testing (collimation tower) and satellite check out operations.

Data is recorded in the Cortex HDR and played back for further processing after the real time activity is completed as shown in Figure 1. Data from the satellite is recorded onto the Cortex unit during the entire duration of the real time activity i.e. while the satellite passes over the ground station [4]. After the real time activity, the data needs to be transferred from the cortex unit to a server for the processing to be done and product to be generated. Latency in processing was observed due to recording of data from satellite in real time in HDR and transfer to a computer system for processing. This two step method of satellite processing was not able to meet the urgent/disaster requirements in the defined time lines. In this context, heterogeneous computing, specifically as-a-service computing, has the potential for significant gains over traditional computing models. Over the years the satellite data acquisition and processing has seen a significant change and improvements in the time lines, compute power requirements and methodologies. Two earlier used methods of data acquisition are explained below.

3.1 Type I Data Acquisition System

The satellite raw data is acquired in real time and logged in the cortex unit initially and manually is transferred to the system through the network port to the server and stored in the hard disk for further processing in software.

The activity as shown in Figure 1 can be classified as real time activity i.e. acquisition of data from the satellite and storing into the demodulator unit. Data transfer from the demodulator unit to the computer system after the real time activity is completed. Initialization of the processing software which is dependent on the system configuration like CPU, memory and network interface [5]. The processing is again divided into two i.e. level '0' processing and higher data processing i.e. level 1 and level2 processing. The level 0 processing involves segregation and decommutation of data and level 0 processing modules to be executed. Thus the level 0 product would be completed around T + 60 minutes which is also dependent on the configuration of the server being used. The higher level production chain would only be initiated after the successful completion of the level '0' product and this is carried out on another system and preprocessed output has to be transferred to the higher level processing system. Thus two server systems were involved in products being generated.

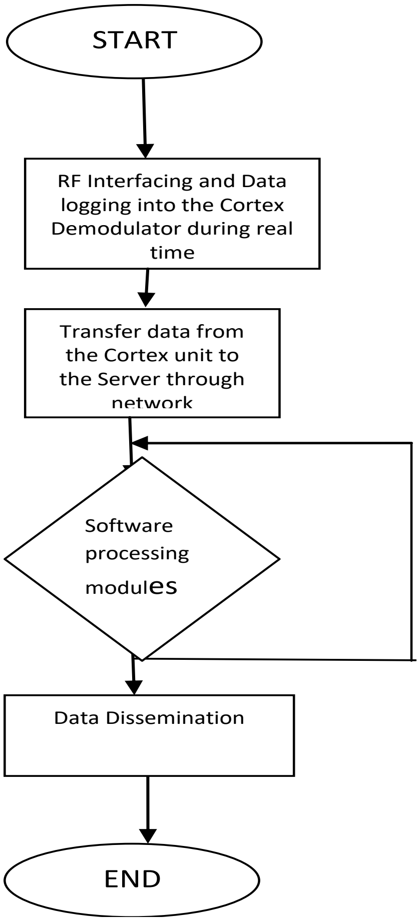
START
RF Interfacing and Data logging into the Cortex Demodulator during real time
Transfer data from the Cortex unit to the Server through network
Software processing modules
Data Dissemination
END

Figure 1: Data acquisition and processing using software.

3.2 Type II Data Acquisition System

The RF signal is interfaced to the cortex unit and the output of the demodulator is the derived clock and data after demodulation. The data and clock are fed to the in-house developed hardware unit as shown in Figure 2.

During real time activity i.e. acquisition of data from the satellite the data is frame formatted, segregated and decommutated in the in house developed hardware. Parallel data is stored in the onboard FIFOs and using the device driver and application software the data is archived in the system local storage a RAID. Data loss is avoided by building the Direct Attached Storage with RAID 1+0 configuration and also high read throughput from the hardware to disk. After the data logging is completed than level '0' processing software is initiated on the archived data and the software processing includes derandomization, decoding, decryption and decompression and the processing time is also dependent on the system configuration like CPU, memory and network interface. The time taken for the completion of level 0 processing was significantly reduced and the level 0 product would be completed around T + 25 minutes. The higher level production chain would only be initiated after the successful completion of the level '0' product.

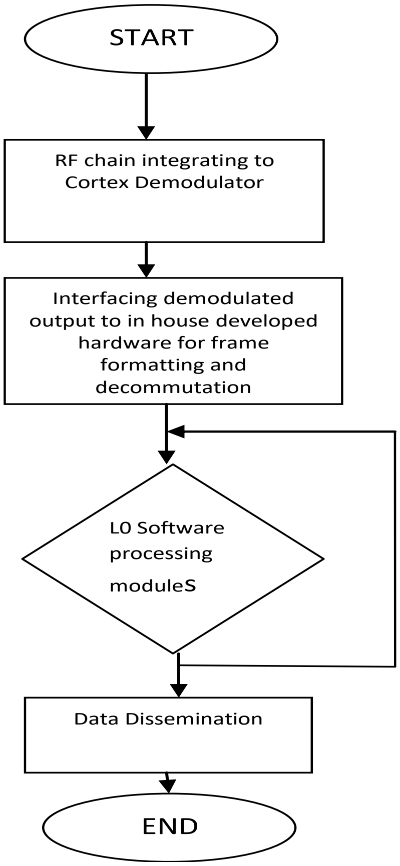
START
RF chain integrating to Cortex Demodulator
Interfacing demodulated output to in house developed hardware for frame formatting and decommutation
L0 Software processing moduleS
Data Dissemination
END

Figure 2: Data Acquisition and processing using hardware and software.

4. FPGA BASED SATELITE DATA ACQUISITION AND PROCESING

Data Acquisition systems are very crucial elements in satellite Ground stations. The principle objective of this paper is to develop a generic multi mission satellite data acquisition hardware with all the related modules being developed for housing into standard server class machines. Real time autocorrelation of the incoming satellite data to detect the valid or invalid frame, log the data of the so that errors can be corrected in near real time.

Decommutation of the data into 64bit Qwords and 64bit word clock thus enabling high data content in a word [6]. Memory modules were designed, utilizing the high speed embedded memory blocks of the FPGA, the decommutated data is first written into the on chip memory of the FPGA in suitable frames depending upon the satellite selected[7]. The on chip memory enables framing accuracy. Later it is transferred to the onboard FIFO, thus enabling mass storage at fast write and read speeds [8].GPS based parallel time code data from a Time code unit is read out, synchronized and embedded into the data frame as an extra word. Interfaced the data from the FIFO to the cache memory of the Server through a PCI express master core for storage on the disk for real time and near real time processing [9],[10] and as shown in Figure 3.

The same hardware can be configured to function as a data simulator. Data patterns as per the format of each satellite is developed and stored in the hard disk of the system [11]. With firmware changes the same hardware can read data from the disc and serialize the data so as to work as simulator for the validation and chain testing of the Satellite ground segment elements.

As the pre-processing of the acquired satellite data is done by CPU in the system where the Embedded hardware is installed and integrated, the FPGA is being used as accelerator since modules like RS decoding, decryption, decompression are all done in real time and in the FGPA thus reducing the time of level '0' processing [12]. As computing needs for high frequency data are intensive and are expected to increase drastically this heterogeneous computing method, specifically FPGA as an accelerator computing, has the potential for significant gains over traditional computing models [13]. The activities carried out by FPGA are shown in Figure 4. During real time activity i.e. acquisition of data from the satellite the data is frame formatted, segregated, decommutated, derandomization, RS decoding, decryption and decompression is done in the FPGA. Level '0' processed data is stored in the onboard memory and using the device driver and application software the data is archived in the system local storage a RAID [14]. Data loss is avoided by building the Direct Attached Storage with RAID 1+0 configuration and also high read throughput from the hardware to disk

The level '0' processing is done in real time and higher level of data processing i.e. level 1 and level2 processing are subsequently taken up. The time taken for the completion of level 0 processing was significantly reduced Thus the level 0 product would be completed around T + 2 minutes as shown in Fig 4. Since a common system is being used for all levels of processing and level 0 processing is done in hardware the higher level data processing is taken up by combination of hardware and software.

Due to the advances in computer resources the ground segment software for higher level processing is designed for near-real time data processing. Thus sequential processing approach is replaced with pipeline based processing resulting in optimal utilization of hardware resources. The main features of the processing framework are that Multiple processes can run in parallel because of the hardware and software combination and consume the data in real-time. Pipeline based processing approach to minimize wait time for data processing.

All the processes are executed in parallel based on Multi thread/ Multi process architecture to utilize maximum logical cores of CPU. Concurrent-execution enables overlapping of I/O and CPU activity, in turn reducing the amount of time that disks and processors are IDLE and increase system throughput [15]. Semaphores and shared memory is used to increase the throughput of each process and reducing disk I/O. Processes are also made fault tolerant to recover automatically for various conditions of failure.

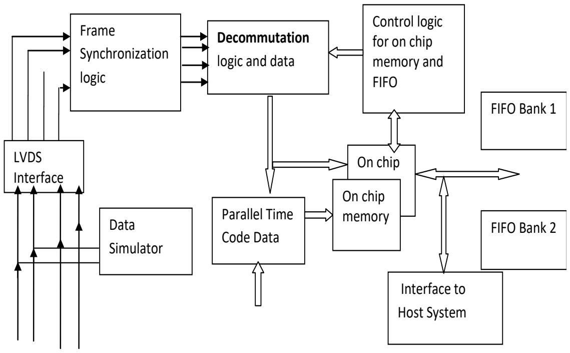

Figure 3: Data Acquisition logic designed in FPGA

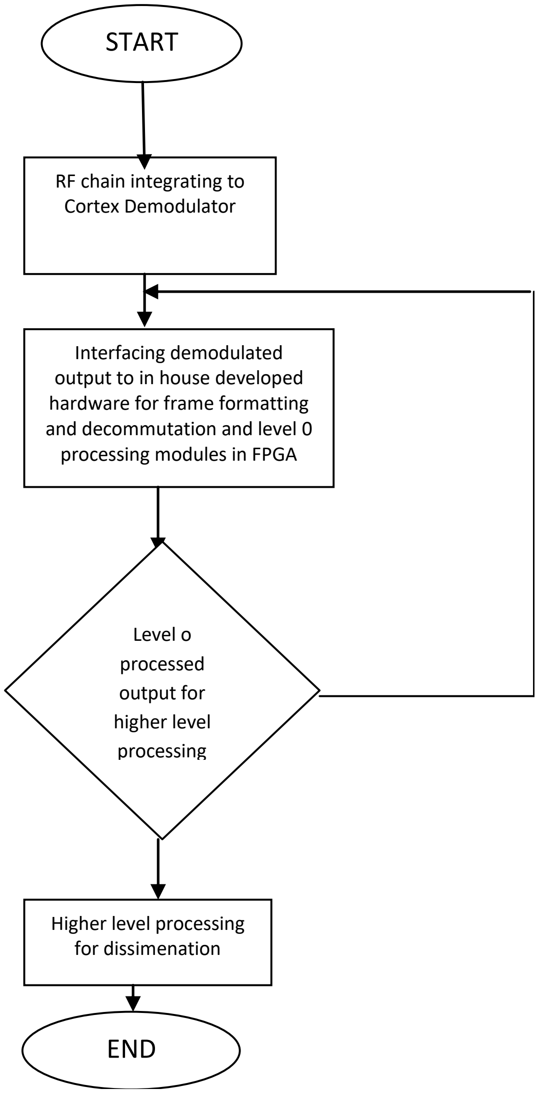
START
RF chain integrating to Cortex Demodulator
Interfacing demodulated output to in house developed hardware for frame formatting and decommutation and level 0 processing modules in FPGA
Level o processed output for higher level processing
Higher level processing for dissimenation
END

Figure 4: Data Acquisition & processing done in FPGA and computer system.

With advances in FPGA's, compute resources and with increasing data rates the Type III data acquisition and processing system has been developed to meet the requirements of the upcoming missions. A comparison the resources utilized, indicating the hardware and software resource utilization and the time taken for completion of processing with all the three types of the data acquisition system is shown in Table 1. The system configuration in all the three cases are four CPU machines with change in the number of cores per CPU and system RAM. The CPU has been offloaded by 60% enabling the same system to be used for higher level product generation with the FPGA based data acquisition system.

Table 1: Comparison of three types of data acquisition and processing.

System Type	Hardware Processing	Software Processing	Compute resources	Hardware resources
Type I	NIL	100%	85% usage	NIL
Type II	10%	90%	75% usage	10% usage
FPGA based	70%	30%	40% usage	60% usage

5. CONCLUSION

The significant increase in the amount of satellite data, observed at the present time, leads to the need of developing methods and technologies for the systems of receiving, storing and processing satellite information. FPGAs have been traditionally been used for various specialized tasks. Their low power consumption and extremely fast processing make them particularly suited for applications in satellite ground station for data processing. Their advantages, however, are not exclusive to these domains and can be leveraged for many other high performance computing tasks. The FPGAs-a-Service which is presented can assist in the implementation of FPGAs as a service in a variety of computing work flows across many domains.

REFERENCES

[1] Artem Perepelitsyn, Inna Zarizenko, Vitaliy Kulanov**"FPGA as a service solutions development strategy".**The 11th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2020, 14-18 May, 2020, Kyiv, Ukraine

[2] Dylan Rankin, Jeffrey Krupa, Philip Harris, Maria Acosta Flechas, Burt Holzman, Scott Hauck, Shih-Chieh Hsu, Matthew Trahms, Javier Duarte, Mia Liu. **"FPGAs-as-a-Service Toolkit (FaaST)".**

[3] NitinPundir, FahimRahman, FariamhFarahmandi, Mark Tehranipoor, University of Florida, USA **"What is All the FaaS About?"** - Remote Exploitation of FPGA-as-a-Service Platforms" Vol. 0, No. 0, Article 000. Publication date: June 2021.

[4] Oleg Yakubailik1, Victor Romas'ko1, and Evgeny Pavlichenko1 **"Complex for reception and real time processing of remote sensing data".**E3S Web of Conferences 75, 03003 (2019).

[5] G.Prasad and N. Vasantha **"DESIGN AND IMPLEMENATION OF PCM DECOMMUTATOR ON A SINGLE FPGA".** Published in the International Journal of Advanced Research in Computer and Communication Engineering. Vol. 1, Issue 7, September 2012.

[6] On-Chip FIFO Memory Core in Volume 5: Embedded Peripherals of the Quartus II

[7] Memory Interfaces made easy with Xilinx FPGAs and memory Interface Generator, WP260(v1.0) February 16,2007.

[8] Developing high speed memory interfaces. The Lattice SCM FPGA advantage White paper February 2006.

[9] Memory System Design from Altera Corporation February 2010.

[10] On-Chip FIFO Memory Core in Volume 5: Embedded Peripherals of the Quartus II.

[11] J. Toledo, H.Muller, J. Buytaert, F.Bal, A.David, A.Guirao and F.J.Mora (2002), **"A plug and play approach to data acquisition",** Network architecture and performance digital equipment corporation, Littleton

[12] Jim McManus, PCI Applications Engineer, Xilinx Inc"Using **FPGAs as a flexible PCI Interface Solution".**

[13] David Robinson, Patrick Lysaght, Gordon McGregor and Hugh Dick **"Performance Evaluation of a Full Speed PCI Initiator and Target Subsystem using FPGAs".**

[14] G.Prasad and N. Vasantha **" INTEGRATING A PCI IP core to FPGA-** Design and Implementation". Published in the International Journal of Engineering and Advanced Technology. (IJEAT) Volume 3, Issue 1, October 2013, pp 36 – 40, ISSN:2249-8958.

[15] G.Prasad and N. Vasantha **"FPGA Based Hardware as a Coprocessor".** Published in the International Journal of Scientific and Research Publications, Volume 4, Issue 12, December 2014. ISSN: 2250-3153.