

Towards a Generic and Adaptive System-on-Chip Controller for Space Exploration Instrumentation

Xabier Iturbe^{1,2,3}, Didier Keymeulen², Patrick Yiu¹, Dan Berisford², Kevin Hand², Robert Carlson² and Emre Ozer³

¹ California Institute of Technology, Pasadena CA, USA; Email: xabier.iturbe@jpl.nasa.gov

² NASA Jet Propulsion Laboratory, California Institute of Technology, Pasadena CA, USA

³ ARM R&D, Cambridge, UK

Abstract—This paper introduces one of the first efforts conducted at NASA’s Jet Propulsion Laboratory (JPL) to develop a generic System-on-Chip (SoC) platform to control science instruments that are proposed for future NASA missions. The SoC platform is named APEX-SoC, where APEX stands for Advanced Processor for space Exploration, and is based on a hybrid Xilinx Zynq that combines an FPGA and an ARM Cortex-A9 dual-core processor on a single chip. The Zynq implements a generic and customizable on-chip infrastructure that can be reused with a variety of instruments, and it has been coupled with a set of off-chip components that are necessary to deal with the different instruments. We have taken JPL’s Compositional InfraRed Imaging Spectrometer (CIRIS), which is proposed for NASA icy moons missions, as a use-case scenario to demonstrate that the entire data processing, control and interface of an instrument can be implemented on a single device using the on-chip infrastructure described in this paper. We show that the performance results achieved in this preliminary version of the instrumentation controller are sufficient to fulfil the science requirements demanded to the CIRIS instrument in future NASA missions, such as Europa.

Keywords—System-on-Chip, ARM, FPGA, Data Processing

I. INTRODUCTION

SoC integration permits to create smaller, cheaper and more reliable electronic systems contained in a single device that are capable to operate at higher clock frequencies and consume less energy. Although these features are of utmost importance when developing space applications, the electronic systems currently used in space exploration missions, such as science instrument electronics, are composed of a collection of devices interconnected in a PCB board, including FPGAs, microprocessors, DSPs, memories and other chips that implement peripheral functionality [1]. In addition, the FPGA design and software code involve huge recurrent engineering costs as they have to be built from scratch for each of the instruments needed in a mission. Finally, the space-qualified rad-hard technology available today (e.g., RAD750 [2]) lags commercial devices (e.g., ARM processors) by several generations, and fails to meet the future computation needs of space exploration missions. The SoC emergent technology is to solve all this issues in a single device package and is now available for precursor flight systems [3]. An early SoC prototype of a flight

instrument using a (rad-hard) Xilinx Virtex-5 FPGA, with an on-chip PowerPC processor, is presented in [4].

In this context, new proposals to use state-of-the-art COTS SoC technology are timidly starting to appear for low-cost space missions. This is the case of the CHREC Space Processor (CSP) that is aimed at using a Xilinx Zynq SoC to develop CubeSats [5]. CSP relies to use a combination of COTS and rad-hard components, where the COTS devices perform critical computations and are supervised by the rad-hard devices.

The work described in this paper is also aligned with the new trend for using state-of-the-art COTS SoC technology in space applications. The objective is to build a generic and lower-cost controller that integrates as much of its functionality as possible in a single chip, and that could be easily adapted to a variety of instruments proposed for future NASA missions. We have named this SoC controller as APEX-SoC (Advanced Processor core for Space EXploration). As in the CSP processor, the APEX-SoC is based on a Zynq. In this initial stage, we have designed a proof-of-concept APEX-SoC prototype and tested it with JPL’s Compositional InfraRed Imaging Spectrometer (CIRIS). The next version of the controller is expected to optimize the software/hardware allocation, improve performance, and will include fault-tolerance features.

The remaining of this paper is organized as follows. Section II describes the APEX-SoC-centric instrumentation controller that is being developed at JPL, making special emphasis on the APEX-SoC infrastructure implemented on the Zynq to support instrument-dependent acquisition and processing logic. Section III describes the CIRIS instrument, and section IV describes the instrument-dependent electronics added to the base APEX-SoC infrastructure. Finally, section V concludes the paper and points out to future work.

II. JPL’S APEX-SOC FOR SCIENCE INSTRUMENTATION

The instrumentation controller is designed to be an independent and stand-alone subsystem in the spacecraft. Its general architecture is shown in Fig. 1 and consists of: (1) digital GPIOs to deal with the digital signals of the instruments, (2) ADCs and DACs to deal with the analog signals, (3) FPGA logic to process the instrument data (using fixed-point representation), (4) an ARM processor to

compute floating-point operations, drive the FPGA logic and implement the interfaces with the spacecraft computer, (5) two DDR memories to store the program executed by the ARM processor (PS-DDR) and intermediate data managed by the FPGA processing logic (PL-DDR), respectively, and (6) a SATA Solid State Driver (SSD) to buffer instrument data until it is transferred to the spacecraft's main memory to be downlinked to Earth, when a spacecraft-earth downlink communication window is available.

A significant part of the controller electronics is integrated in a single Xilinx Zynq chip (the APEX-SoC), namely the FPGA logic, the ARM processor and the interfaces with the other components: ADC/DACs, DDR memories and SSD, as well as with the spacecraft main computer through Ethernet.

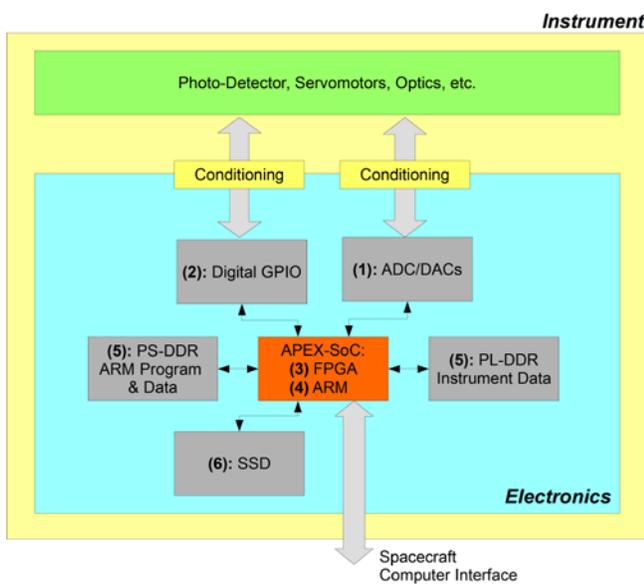


Figure 1. JPL's instrumentation controller general architecture

The instrumentation controller is currently being prototyped on a ZedBoard mini-ITX development board, populated with a Xilinx 7Z100 Zynq SoC device [6], to which an FMC board containing the ADCs and DACs is to be attached. It is important to note here that this development board is planned to be used only in the early stages of the design; all the components necessary in the controller can actually be integrated in a board of small dimensions, such as the one presented in [5], or in its flight equivalent for low radiation dose up to 25krad [3]. The ZedBoard is equipped with digital isolators to electrically isolate the Zynq I/O pins from the instrument it controls, and to translate the voltage levels between both devices. The PL-DDR memory is organized in different data segments, each of which is read and written by a specific data processing stage implemented on the APEX-SoC. The number and size of the segments can be customized to the specific data processing needed by each instrument. When dealing with instruments that generate small amounts of data, the data can be stored using solely on-chip memory resources, such as Block-RAMs (BRAMs).

As depicted in Fig. 1, the APEX-SoC is the central component in the instrumentation controller. It includes a dual core ARM-based processing system (shown in green and white color in Fig. 2) running the flight software, and a SoC infrastructure (shown in gray color) that gives support to the instrument-dependent custom acquisition and processing logic (shown in yellow color). While the latter infrastructure is exclusively implemented on the FPGA fabric of the Zynq, the ARM-based processing system includes components that are implemented using both programmable (i.e., FPGA) and fixed logic in the chip. The fixed logic components, shown in white color, include: (1) the ARM Cortex-A9 dual-core processor, (2) a controller for the PS-DDR memory and, (3) a set of communication controllers, such as Ethernet. On the other hand, the components of the ARM-based system that are implemented on the FPGA fabric, shown in green color, include: (4) the AMBA AXI bus interconnect, (5) a DMA controller, (6) an AXI Data Mover (which acts as a DMA controller for the FPGA custom processing logic), (7) GPIOs, (8) a SATA SSD controller, and (9) a controller for the PL-DDR. The latter PL-DDR controller is accessed by both the DMA controller, to carry out data transfers requested by the ARM processor, and by the AXI Data Mover, to carry out data transfers requested by the FPGA custom logic.

The components that give support to the custom instrument acquisition and processing logic are: (10) ADC/DAC SPI interface logic, (11) a buffer to temporarily store the digitized instrument data samples prior to be written in burst to the PL-DDR memory, (12) a crossbar to connect the instrument-dependent processing cores to the PL-DDR memory, (13) a data-flow controller that dictates the order of execution of the processing stages and controls data transfers to/from the PL-DDR memory and, (14) a register bank connected to the ARM-based system's AXI bus to set the functioning and configuration of the acquisition and processing logic as well as to retrieve its status from the software running on the ARM processor.

We note that in this preliminary version of the instrumentation controller, we are using the existing DDR3 memory in the ZedBoard as a standard SDRAM memory with limited bandwidth (i.e., single 32-bit access per clock).

The instrument data processing occurs in bursts (commanded either by the DMA or by the AXI Data Mover) and the acquisition and processing cores are connected each other by means of standard AXI-Stream buses that support data bursts at the highest transfer rate, one data per clock cycle [7]. The buffer used to temporarily store the instrument data samples is precisely intended to optimize the burst data transfers carried out by the AXI Data Mover. Being the central interconnection component in the system, the crossbar has AXI-Stream ports to which the instrument-dependent cores are attached. The data-flow controller indicates to the AXI Data Mover the start memory address and the number of data bytes to read/write through the crossbar, and the latter signals the data-flow controller when the data transfer has been completed. Therefore, the data-

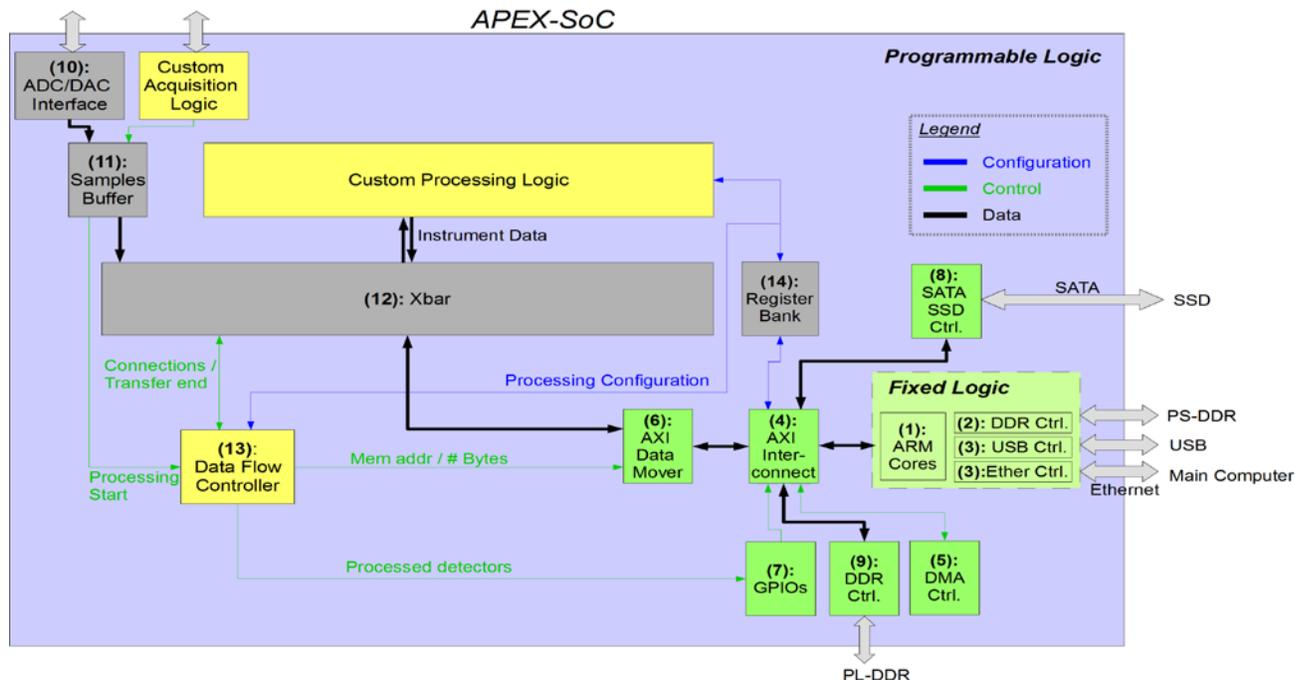


Figure 2. JPL's APEX-SoC architecture block diagram

flow controller needs to be aware of the system configuration at every time and keeps track of the PL-DDR memory data segments dedicated to each processing stage.

All the APEX-SoC infrastructure components described above are designed in a generic way (i.e., using HDL generics) to meet the needs of a wide range of instruments. The list of parameters that can be adjusted at synthesis time include: (1) the sampling frequency and (2) number of ADC/DACs in the system, (3) the size of the samples buffer, (4) the number of AXI-Streamports in the crossbar, and (5) the amount of registers in the configuration register bank. The only infrastructure component that is not generic enough and must be considered instrument-dependent is the data-flow controller. Although this has a quite generic architecture that can be adapted to potentially any data-flow application and, even though the designer can use a HDL template and rely on a set of very useful signals to invoke the data transfers with the PL-DDR memory (e.g., data transfer ending signal provided by the crossbar), large parts of application-dependent code still need to be written from scratch, thus involving a considerable design effort.

In order to ease the development of instrumentation applications, the ARM processor in the APEX-SoC runs a Linux-based operating system, which provides Ethernet protocol to communicate with the spacecraft's main computer, a file system to ease the management of the instrument data stored in the SSD, as well as driver routines for other communication interfaces present in the Zynq. Other software routines to abstract the APEX-SoC

infrastructure are also available, such as to help programming DMA transfers.

The generic instrumentation controller architecture described in this section (i.e., ADCs/DACs + DDR Memory + SSD + ARM processor + FPGA logic, along with the platform SoC infrastructure implemented on the APEX-SoC and the Linux-based operating system running on the ARM cores) will be reused for implementing SoC control electronics of several different JPL instruments, where only the instrument-dependent data acquisition and processing logic needs to be modified.

III. JPL'S CIRIS INSTRUMENT

This paper uses the CIRIS instrument as a case-study for the APEX-SoC-based generic instrumentation controller described in the previous section. This instrument is described in detail in [8].

CIRIS is a compact and robust Fourier Transform Spectrometer (FTS) that is suitable for airborne and spaceborne measurements in the near-IR to thermal-IR band (2-12 μm) typically required in space exploration missions. As shown in Fig. 3, CIRIS replaces the linearly moving mirrors used in related spectrometers to vary the Optical Path Difference (OPD) of incoming light rays with a constant-velocity rotating refractor that is made of Zinc Selenide (ZnSe). Besides the refractor DC servomotor, for in-situ applications, CIRIS is equipped with three additional servomotors to move the focusing lens on the collected material samples to analyze (see Fig. 3).

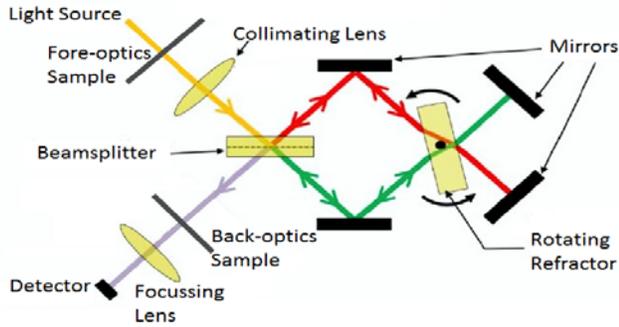


Figure 3. Diagram of CIRIS [10]

Based on Snell's law, the OPD in CIRIS is zero whenever the rotating refractor is parallel or perpendicular to the beam splitter as each of the split beams are incident on the refractor at 45° and thus travel the same distance through the instrument. As the rotating refractor is either parallel or perpendicular to the beamsplitter four times over the course of a revolution, this configuration allows the refractor to induce Zero OPD (ZPD) for four positions per revolution. CIRIS measures the optical interference at the photo-detector due to the optical difference between the two rays generated by the beam splitter. A nice optical property of the instrument is that the OPD scales approximately linearly in relation to the angle/position of the refractor around the ZPD. The regions where the interference can be measured occur four times per revolution of the refractor and are located at approximately 16° arcs around each of the ZPD positions. A Faulhaber E2-360I optical incremental encoder [9] mounted on the DC servomotor that drives the refractor's rotation is used to identify the latter regions. The optical interference measured in the photo-detector while the refractor sweeps through these areas is called interferogram. As the refractor performs 6.5 revolutions per second in CIRIS, interferograms span over a period of 13.6 ms every 24.8 ms. The interferograms are processed via a Fast Fourier Transform (FFT) to produce a spectrum that illustrates the intensity of the different wavelengths present in the light beam. This in turn permits to know the chemical composition of the sample by looking at the absorption lines in the spectrum. However, spectral leakage (e.g., "picket-fence" effect) and noise are also present in the spectrum due to the limited discretization of the interferograms through time limited digital sampling, and need to be properly handled by the instrument electronics to produce meaningful results [11].

The next generation of CIRIS will be equipped with multiple photo-detectors to increase the spectral and spatial resolution of the instrument. The photo-detectors will be sensitive to either near, mid and far-infrared wavelengths. Besides, the spatial distribution of the detectors in an array will allow to map the chemical composition of the sample or body under study.

In the NASA Europa mission, CIRIS is proposed to analyze the chemical composition of the ice surface of this Jovian moon [12]-[14], as a first approach to understand the

chemical dynamics of its internal sea. In this mission, the CIRIS controller will have to process interferogram data collected by as many as 25 photo-detectors and will have to tolerate the severe radiation conditions prevailing in Jupiter's magnetosphere, which are estimated to be seven times greater than in any previous NASA mission [15].

IV. CIRIS-DEPENDENT FUNCTIONALITY

This section details how the APEX-SoC in section II has been customized to work with the CIRIS instrument described in section III. Much of the data-processing stages described here are based on [16] and have been previously discussed in [17]-[19].

The signals delivered by the optical encoder mounted on the CIRIS rotating refractor are connected to the controller's digital input, while the interferogram analog signals are to be digitized by the ADCs populated in the FMC board attached to the mini-ITX ZedBoard. As there is a single photo-detector in the current version of CIRIS, only one ADC is used, namely an Analog Devices AD7984 [20]. This ADC is configured to sample the interferogram signal at 1 MSPS and with 18-bit resolution, capturing approximately 13,600 samples over each interferogram sampling region. The interferogram signal is conditioned, filtered and amplified to $\pm 5V$ range prior to being digitized. Despite all of the interferogram data detected by the single photo-detector existing in the current version of the CIRIS instrument can be stored on-chip, the external PL-DDR memory needs to be used when dealing with the new version of CIRIS equipped with multiple photo-detectors. No DAC is needed to control the CIRIS instrument and the servomotors that move the input lens are commanded through the USB interface.

The CIRIS controller implemented on the APEX-SoC distinguishes two different functioning phases that are alternated over time: (1) the interferogram acquisition phase, which occurs four times per revolution of CIRIS refractor and spans over a period of 13.6 ms, and (2) the interferogram processing phase, during which the acquired interferogram data are processed and spans over a period of 24.8 ms.

The instrument-specific acquisition logic in this application is the logic that interfaces with the optical encoder mounted on the CIRIS rotating refractor to identify the samples that are associated to valid interferogram areas and, more specifically, to which of the four ZPD positions those samples are associated with. This interfacing logic has been designed in a generic way and can be easily adapted to work with different rotation speeds of the refractor and different widths of the valid interferogram regions. The interferogram samples corresponding to each of the four ZPD positions are stored in separate memory segments and processed independently as they require different spectral corrections due to variations in the CIRIS refractor optics between the ZPD positions.

The instrument-specific processing logic in this application consists on the interferogram processing stages that are shown in Fig. 4. These have been previously

prototyped at JPL using LabVIEW in a cRIO platform that includes a separate PowerPC processor and a Xilinx Virtex-5 FPGA [21]. In general, the processing cores receive data from the PL-DDR memory, process it and write the results to dedicated segments within the memory, from where they are delivered to the next processing stage or read by the ARM processor. Some processing cores, however, bulk their results directly to the next processing stage, without buffering them on the PL-DDR memory. Such cores include FIFO memories on their interfaces.

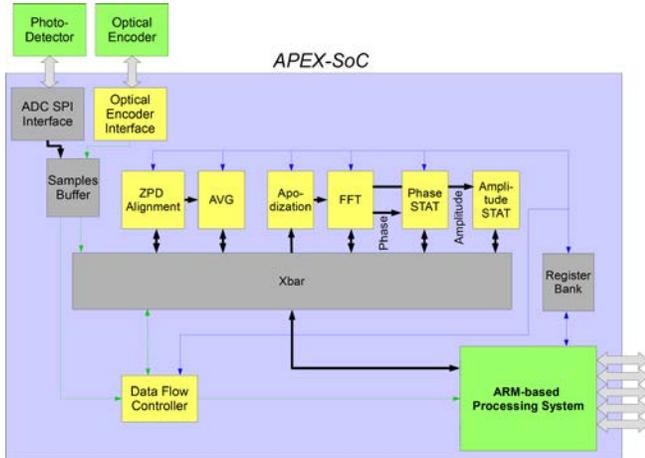


Figure 4. CIRIS-specific logic (yellow) added to the APEX-SoC

A. Interferogram Processing Stages

The interferogram processing stages in the CIRIS controller include: (1) a ZPD alignment core to correct any temporal shifts that might have occurred while sampling the interferogram, (2) a core to apodize the interferogram at the edges of the sampled region to minimize the effects of spectral leakage, (3) an FFT core to translate the interferogram data into the spectrum domain, and (4) some cores that implement Knuth’s algorithm [22] to perform a cumulative moving average on successive data streams with the objective of increasing the SNR by eliminating the effect of high frequency and random noise in the instrument. At the price of slightly increasing the computation burden and time, these cores can also compute the variance of the data streams to enable the CIRIS controller for estimating the SNR of the instrument at each time. There are three of these cores in the system, named as STAT in Fig. 4. One of the STAT cores operates on the ZPD aligned interferograms samples from the same rotational refractor position across several rotations, while the other two operate on the phase and amplitude (i.e., spectrum) values resulting from applying the FFT on the interferograms. The reason to choose Knuth’s algorithm to compute the cumulative average and deviation is that it is much less prone to loss of precision due to massive cancellation than related algorithms. This is very important in space exploration missions, where the averaging processes can span for a period of time as long as several minutes in some cases. As shown in Fig. 5, Knuth’s algorithm uses only

two variables (delta and mean) to compute the cumulative average and an additional variable (M2) is needed when also computing the variance. These variables are stored in the data PL-DDR memory between successive iterations of the algorithm.

```

n = 0;
mean = 0;
M2 = 0;

for x in data:
    n = n + 1;
    delta = x - mean;
    mean = mean + delta/n;
    M2 = M2 + delta*(x - mean);
end for;

variance = M2 / (n - 1);

```

Figure 5. Knuth algorithm pseudocode

The first interferogram processing stage is implemented by the ZPD alignment core, which identifies the sample corresponding with the ZPD position and selects 8,192 samples centered around it, discarding the remaining data. The ZPD aligned 8,192 interferogram samples are either passed to the interferogram STAT core, or written back to the PL-DDR memory. The interferogram data are next passed to the apodization core, which multiplies them by an apodizing function, and the resulting values are delivered to the FFT core.

In order to increase the spectral resolution, the FFT core can add 4,096 zeros to each of the tails of the input interferogram to obtain 8,192 additional interpolated spectrum data in-between the original nonzero-filled spectrum points, that is, 16,384 total spectrum points. The zero padding allows to reduce the erroneous signal due to the “picket-fence” effect, which can be as large as 36% [16]. If zero-filling is not enabled, the core computes the 8,192-point FFT. Due to the Hermitian symmetry of the spectrum, only half of its points are meaningful in CIRIS spectrometry. In order to obtain the best performance, the FFT core pipelines several Radix-2 butterfly processing engines, where each engine has its own memory banks to store input and intermediate data. Likewise, with the objective of improving the accuracy of the computation, the FFT core uses full-precision unscaled arithmetic. This means that the width of the data path increases to accommodate the bit growth through the butterfly stages, reaching 48 bits in the last stage. This pipelined implementation allows the FFT core to deliver one spectrum data per clock cycle, with a latency of 16,630 clock cycles when zero-filling is disabled and 33,013 clock cycles when zero-filling is enabled. 36 additional clock cycles are needed to translate the real and imaginary representation of the spectrum data used in the internal butterfly processing engines into polar representation, which is more suitable for scientific analysis. This translation is accomplished using CORDIC logic and the resulting phase and amplitude values are represented using a single 32-bit word, thus reducing the amount of data transfers to/from the PL-DDR memory.

TABLE I FPGA RESOURCES CONSUMED BY EACH CORE

Core	Slice Flip-Flops	Slice LUTs (Logic)	Slice LUTs (Memory)	DSP Blocks (DSP48s)	BRAMs (36 Kb)	BRAMs (18 Kb)
SPI ADC Interface	36	25	-	-	-	-
Optical Encoder Interface	23	48	-	-	-	-
Interferogram Buffer (1KB)	62	87	-	-	-	1
Crossbar (7x1 ports)	-	181	-	-	-	-
ZPD Alignment	64	186	-	-	18	-
STAT (1,024 max. trials)	451	608	43	10	31	3
Apodization	111	123	17	4	-	1
FFT (16,384-point)	20,304	11,193	3,244	130	6	68
Data-flow Controller	177	324	-	-	-	-
ARM-based System	13,178	13,470	2,024	-	11	18
Totals	35,334	31,123	5,414	164	128	97
Available in Zynq 7Z100	554,800	277,400	108,200	2,020	755	1,510
% used	6.4%	11.2%	5%	8.1%	17%	6.4%

The long latency introduced by the FFT core is used to carry out delta, mean and M2 data transfers for the STAT cores. Namely, two 8,192 data sets can be transferred when computing 8,192-point FFT (i.e., zero-filling disabled) and up to four data sets can be transferred when computing 16,384-point FFT (i.e., zero-filling enabled).

When dealing with the next generation CIRIS instrument, the data-flow controller repeats all the processing steps for each of the interferograms measured in the photo-detector array.

All the processing cores described above are parameterizable, using HDL generics, and run-time adjustable to adapt to unexpected data observed while exploring unknown space environments. The list of the parameters that can be configured include: (1) in the ZPD alignment core, the method to detect the position of the ZPD sample (e.g., most-negative, most-positive or most-magnitude), (2) in the STAT cores, the number of trials to be averaged and the requirement to compute or not the variance, (3) in the apodization core, the apodizing function, and (4) in the FFT core, the requirement for zero-filling (16,384 spectrum points) or not (8,192 spectrum points). With regard to run-time adaptivity, it is important to note that the coefficients of the apodizing function are stored in a BRAM that is writable by the ARM processor.

When all the interferograms have been processed by the processing FPGA logic, the data-flow controller signals an interruption to the ARM processor. This performs the last interferogram processing stages that involve floating-point operations, including those related to the spectral corrections due to variations in the CIRIS refractor optics between the four ZPD positions, and to the compensation for the OPD variations due to the CIRIS ZnSe refractor's refractive index dependency with wavelength. In addition, the ARM processor uses the computed phase values as feed-back to adjust the focusing of the CIRIS optics. The computed spectrum points are finally stored in the SSD until they are requested by the spacecraft main computer.

Table II shows the internal organization of the PL-DDR memory, which is divided into data segments dedicated to each processing stage. These segments are in turn divided into four regions that store the data corresponding to the four ZPD positions where valid interferograms are detected. Note that this data structure is replicated in the PL-DDR memory as many times as photo-detectors are in the CIRIS array. Therefore, up to 34.6 MB are needed when dealing with the 25 photo-detectors in the next generation of CIRIS, approximately 6.75% of the total amount of available memory.

TABLE II DATA SEGMENTS IN THE PL-DDR MEMORY

Data Segment: Content	Size (max.)	Write Core	Read Core
Detected raw interferogram	256 KB	Buffer	ZPD
ZPD aligned inter.	128 KB	ZPD, STAT	STAT, Apo.
ZPD aligned inter. (M2)	128 KB	STAT	STAT, ARM
ZPD aligned inter. (delta)	128 KB	STAT	STAT
Phase	128 KB	FFT, STAT	STAT, ARM
Phase (M2)	128 KB	STAT	STAT, ARM
Phase (delta)	128 KB	STAT	STAT
Amplitude	128 KB	FFT, STAT	STAT, ARM
Amplitude (M2)	128 KB	STAT	STAT, ARM
Amplitude (delta)	128 KB	STAT	STAT
Total per detector	1.384 MB		
Available memory	1024 MB		
% used (25 detectors)	3.4%		

B. Implementation and Timing Results

The processing stages used in this case-study have been implemented in a very efficient way on the Zynq's FPGA fabric and can run at 200 MHz. As shown in Table II, the complete CIRIS controller, including the ARM-based system and the SoC infrastructure consume 176 BRAMs, 164 DSP blocks and 36,537 LUTs. This represents approximately 23% of the total BRAMs, 8% of the DSP blocks and 13% of the LUTs on the used Zynq 7Z100 part. On its part, the 28-nm ARM dual-core processor on the Zynq runs at 800 MHz, delivering a maximum performance of 2.5 DMIPs per MHz per core, and more importantly, consuming less than 0.25 mW per MHz per core [23].

Table III shows the processing time taken by the different stages implemented in the CIRIS controller. Note that the processing time of the STAT cores is different for the first trial and others. This is because the cores initialize all of their internal variables to zero in the first iteration, circumventing the need for the data transfers from the PL-DDR memory. In the worst-case configuration (i.e., when variance computation is required in all STAT cores and zero-filling is enabled in the FFT core), the total amount of time needed to process a single interferogram is 821.6 μ s. This is roughly 5x performance improvement compared to the prototypic cRIO-based implementation. When using the current CIRIS configuration, in which the interferogram processing phase spans approximately 24.8 ms, it is estimated that the APEX-SoC-based CIRIS controller can successfully manage about 30 photo-detectors, which is enough to fulfill the science requirements demanded for this instrument in future NASA missions, such as Europa.

TABLE III REQUIRED TIME BY EACH INTERFEROGRAM PROCESSING STAGE

Processing Core	Configuration	Time
ZPD Alignment	-	218.4 μ s
STAT	Variance required (<i>1st trial</i>)	245.8 μ s
	Variance required (<i>Others</i>)	409.6 μ s
	No Variance required (<i>1st trial</i>)	163.8 μ s
	No Variance required (<i>Others</i>)	245.8 μ s
Apodization	-	81.9 μ s
FFT	Zero-filling enabled	412.0 μ s
	Zero-filling disabled	207.7 μ s
Total per detector	<i>Worst-case Configuration</i>	821.6 μs
Total 25 detectors	<i>Worst-case Configuration</i>	20.5 ms

Another key benefit of the APEX-SoC-based implementation of the CIRIS controller is the achieved 2x saving in power consumption compared to the cRIO-based implementation. In the cRIO, only the Virtex-5 FPGA needs to dissipate more than 5 W, while the worst-case estimation given by Xilinx design tools for the Zynq design, which also includes the power delivered to the DDR memories, is only 2.7 W.

V. CONCLUSIONS AND FUTURE WORK

This paper has described the ongoing work at JPL to develop a SoC Zynq-based instrumentation controller (APEX-SoC) that could be used with a wide variety of instruments in future NASA space exploration missions. Integrating most of the functionality necessary to control the science instruments (i.e., hardware logic + software routines) into a single chip will lead to lighter, cheaper and more reliable electronic systems that consume less power and deliver higher performance. This paper has used JPL's CIRIS instrument as a case-study to demonstrate the feasibility of the APEX-SoC-based instrumentation controller. A SoC implementation of the controller electronics will allow for integrating science instruments, such as CIRIS, in low-cost space missions such as CubeSats.

Future work will be oriented in two directions: increase the performance and reliability of the instrumentation controller. In order to increase performance and in advance of the readiness of space-qualified DDR memories [24], we will explore the use of a Multi-Port Memory Controller (MPMC) for the PL-DDR memory that stores the instrument data in the controller [25]. In the case of CIRIS, where the interferograms detected in the photo-detector array are independent of each other, this will allow us for taking advantage of the high bandwidth delivered by DDR technology (i.e., DDR3 operating at 800 MHz and offering 6.4 GB/s [26]) as well as the parallel computation delivered by the FPGA fabric. As a result, the instrumentation controller will be able to process more interferograms in the same time, enabling a higher spectral resolution in the sample analysis. In order to make the APEX-SoC more reliable, we will integrate fault-tolerance features on the FPGA fabric already available for the Zynq in the Xilinx design tools, such as ECC codes in memories, "safe implementation" of Finite State Machines (FSMs) and Soft Error Mitigation Controller (SEM IP) based "scrubbing" technique [27], as well as implement other fault-tolerant techniques such as redundancy [28], dynamic partial reconfiguration based techniques [29] and fault-resilient mapping on the FPGA fabric [30]. In the case of CIRIS, interferogram processing techniques that minimize the degradation provoked by radiation are also available to be implemented [31].

ACKNOWLEDGMENT

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (NASA). Xabier Iturbe is funded by the European Commission's FP7 Marie-Curie International Outgoing Fellowship Program with "Project No. 627579".

REFERENCES

- [1] R. Key, S. Sander, A. Eldering, J. F. Blavier, D. Bekker, K. Manatt, D. Rider and Y. H. J. Wu, "The Geostationary Fourier Transform Spectrometer", in Proc. of the IEEE Aerospace Conference, 2012.
- [2] BAE Systems Inc., "RAD750 Radiation-Hardened PowerPC Microprocessor", 2008.
- [3] CubeSat Flight Computer (CFC-300) Innoflight Inc. (<http://www.innoflight.com>), 2015.
- [4] D. L. Bekker, J. Blavier, R. W. Key, D. M. Rider and S. P. Sander, "An FPGA-based Focal Plane Array Interface for the Panchromatic Fourier Transform Spectrometer", in Proc. of the IEEE Aerospace Conference, 2011.
- [5] D. Rudolph, C. Wilson, J. Stewart, P. Gauvin, A. George, H. Lam, G. Crum, M. Wirthlin, A. Wilson and A. Stoddard, "CHREC Space Processor: a Multifaceted Hybrid Architecture for Space Computing", in Proc. of the AIAA/USU Conference on Small Satellites, 2014.
- [6] Avnet Inc., "Xilinx Zynq Mini-ITX Development Kit User Guide".
- [7] Xilinx Inc., "AXI Reference Guide", UG761, 2012.
- [8] D. F. Berisford, K. P. Hand, P. J. Younse, D. Keymeulen, and R. W. Carlson, "Thermal Testing of the Compositional Infrared Imaging

- Spectrometer (CIRIS)", in Proc. of the International Conference on Environmental Systems, 2012.
- [9] Faulhaber, "Optical Incremental Encoders Datasheet", 2013.
- [10] D. Tamas-Selicean, D. Keymeulen, D. F. Berisford, R. W. Carlson, K. P. Hand, P. Pop, W. Wadsworth, and R. Levy, "Fourier Transform Spectrometer Controller for Partitioned Architectures", in Proc. of the IEEE Aerospace Conference, 2013.
- [11] V. Saptari, "Fourier Transform Spectroscopy Instrumentation Engineering", SPIE Press, vol. 61, 2004.
- [12] R. W. Carlson, K. P. Hand, D. F. Berisford and D. Keymeulen, "The Compositional InfraRed Interferometric Spectrometer (CIRIS) for Assessing the Habitability of Europa", in Proc. of the American Geophysical Union Fall Meeting, 2013.
- [13] R. W. Carlson, W. Calvin, J. Dalton, G. Hansen, R. Hudson, R. Johnson, T. McCord and M. Moore, "Europa's Surface Composition", Europa, pp. 283-327, 2009.
- [14] K. P. Hand, C. Chyba, J. Priscu, R. W. Carlson and K. Neelson, "Astrobiology and the Potential for Life on Europa", Europa, pp. 589-629, 2009.
- [15] J. D. Cressler and H. A. Mantooth, "Extreme Environment Electronics", CRC Press, 2012.
- [16] J. Gronholz and W. Herres, "Understanding FT-IT Data Processing", Instruments & Computers, vol. 3, No. 10, pp. 1-23, 1985.
- [17] W. Wadsworth and J. P. Dybwad, "Rugged High-Speed Rotary Imaging Fourier Transform Spectrometer for Industrial use", in Proc. SPIE 4577, Vibrational Spectroscopy-based Sensor Systems, 2002.
- [18] J. O. Jensen, A. I. Iñarraguerri, W. R. Loerop, W. Wadsworth and J. P. Dybwad, "Chemical Imaging System", in Proc. SPIE 4574, Instrumentation for Air Pollution and Global Atmospheric Monitoring, 52, 2002.
- [19] W. Wadsworth, J. P. Dybwad and D. Stone, "Mosaic imaging Fourier transform spectrometer applications", in Proc. of the International Society for Optics and Photonics Defense and Security, pp. 255-266, 2005.
- [20] Analog Devices Inc., "Evaluation Board User Guide: Evaluation Board for the 8-/10-Lead Family of 14-/16-/18-Bit PulSAR ADCs", 2014.
- [21] P. Yiu, D. Keymeulen, D. F. Berisford, K. P. Hand, R. W. Carlson, W. Wadsworth, J. P. Dybwad and R. Levy, "Design and Integration of an Adaptive Controller for a Fourier Transform Spectrometer", in Proc. of the NASA/ESA Conference on Adaptive Hardware and Systems, 2014.
- [22] D. E. Knuth, "The Art of Computer Programming", vol. 2: Seminumerical Algorithms, Addison-Wesley, 1998.
- [23] Xilinx Inc., "Zynq-7000 All Programmable SoC Overview", 2014.
- [24] J. Marshall, "DDR Memory Solutions for Next Generation Space Systems", in Proc. of the Microelectronics, Reliability and Qualification Working Meeting, 2015.
- [25] K. Dao and D. Buli, "AXI Multi-Ported Memory Controller", Xilinx Inc., XAPP739, 2011.
- [26] Micron Inc., "4Gb: x4, x8, x16 DDR3L SDRAM Description", 2011.
- [27] Soft Error Mitigation Controller – How to use SEM IP with Vivado Lab Tools (<http://www.xilinx.com/support/answers/54460.html>)
- [28] TMRTool: Triple Module Redundancy (TMR) for space grade re-programmable FPGAs
http://www.xilinx.com/ise/optional_prod/tmrtool.htm
- [29] X. Iturbe, A. Ebrahim, K. Benkrid, C. Hong, T. Arslan, J. Perez, D. Keymeulen, M. D. Santambrogio, "R3TOS-based Autonomous Fault-Tolerant Systems", IEEE Micro, vol. 34, no. 6, pp. 20-30, 2014.
- [30] J. D. Corbett, "The Xilinx Isolation Design Flow for Fault-Tolerant Systems", Xilinx Inc., WP412, 2013,
(http://www.xilinx.com/support/documentation/white_papers/wp412_IDF_for_Fault_Tolerant_Sys.pdf)
- [31] P. Yiu, D. Keymeulen, D. F. Berisford, K. P. Hand, R. W. Carlson, X. Iturbe, W. Wadsworth, J. P. Dybwad and R. Levy, "Adaptive Controller for a Fourier Transform Spectrometer with Space Applications", in Proc. of the IEEE Aerospace Conference, 2015.