Scalable Sensor Data Processor: Testing and Validation

R. Pinto[a] , L. Berrojo, L. Gomez, F. Piarrette, P. Sanchez, E. Garcia, R. Trautner[b], G. Rauwerda[c], K. Sunesen, S. Redant[d], S. Habinc[e], J. Andersson, J. López[f]

[a]Thales Alenia Space Spain (TAS-E), 28760 Tres Cantos, Spain
[b]ESA, 2200 AG Noordwijk, The Netherlands
[c]Recore Systems B.V., 7500 AB Enschede, The Netherlands
[d]IMEC, B-3001 Leuven, Belgium
[e]Cobham Gaisler AB, SE-411 19 Göteborg, Sweden
[f]Arquimea Ingeniería, S.L.U., 28919 Leganés, Madrid, Spain

ricardo.pinto@thalesaleniaspace.com

Abstract

The Scalable Sensor Data Processor (SSDP) is a next-generation heterogeneous multicore mixed-signal ASIC for on-board data processing, embedding in the same chip resources for high-performance data processing and control. These resources are organized as a System-on-a-Chip (SoC) together with generic and specialized interfaces for Input/Output (I/O), as well as interfaces for data acquisition.

Test and validation of such diversity requires an adequate prototyping platform connected to flexible Electrical Ground Support Equipment (EGSE), which are exploited with representative use-cases and applications. This paper details the test and validation activities of the SSDP, ranging from low-level interface testing up to benchmarking.

I. Introduction

Heterogeneous computing architectures are poised to be part of next-generation on-board processing systems, due to their appealing properties, such as flexibility and power efficiency. The flexibility conferred by mixing different computing architectures is undeniable, allowing the co-existence of processing and control in the same package. These are further enriched by a complete set of input/output (I/O) peripherals, in a System-on-a-Chip (SoC) fashion. The Scalable Sensor Data Processor (SSDP) is an example of such devices, having resources for processing, control and data acquisition in the same package. Furthermore, it has local and networked I/O, and the capability of being connected to other SSDP devices to scale a system towards higher performances.

Testing and validation of such devices encompasses many different tasks, stemming from their very SoC nature. For example, there are several I/O interfaces which require testing, and at the same time, the interaction between these and the processing elements must be validated. Such test and validation requires specialized hardware in the form of Electrical Ground Support Equipment (EGSE), with the appropriate interfaces and test execution support.

This paper is organized in the following manner: Section II broadly presents the SSDP architecture, its main blocks and I/O interfaces; Section III and IV describe the prototyping support required by the SSDP and the planned test and validation work; Section V explains the support needed by the testing and validation activities, as well how these are being carried out, both at hardware and software level; and Section VI concludes this paper.

II. SSDP Architecture

The SSDP is a next-generation mixed-signal ASIC for on-board data processing, with a heterogeneous multicore SoC architecture. It embeds specialized Digital Signal Processors (DSPs) together with a General-Purpose Processor (GPP), being capable of delivering high-performance processing together with reliable control. The SSDP architecture can be divided in two major subsystems, based on their main scope:

- **Processing**, with the multicore DSP, large memory and data acquisition interfaces;
- **Control**, with the GPP and I/O interfaces, both local and networked

These subsystems are connected via bidirectional bridges, translating signalling and data between them. A block diagram depicting the subsystems and their modules is shown in Figure 1.

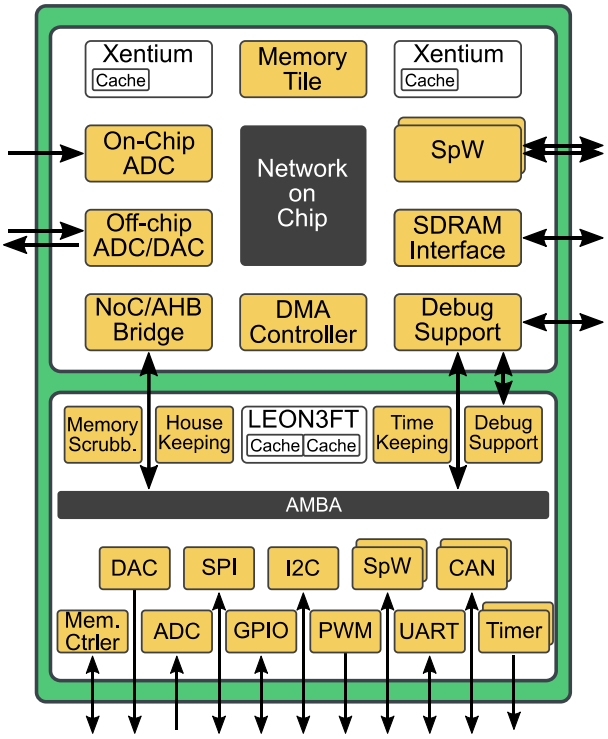

Figure 1: SSDP Architecture Block Diagram

The Processing Subsystem is based on Recore Systems multicore DSP IP, containing two Xentium® fixed-point DSP cores [1] connected to I/O interfaces and SDRAM memory via a high-speed Network-on-Chip (NoC) interconnect. This subsystem is oriented to data processing and contains an internal 64 kB SRAM (Memory Tile) as well as a DMA Controller which can be exploited to efficiently move data between the several components. On- and Off-chip data acquisition is possible, via dedicated bridges. Furthermore, the ADC/DAC Bridge can double as a high-throughput 16-bit Chip-to-Chip interface, capable of reaching 800 Mbps and supporting flow-control mechanisms.

The Control Subsystem is based on the well-known Cobham Gaisler LEON3 System-on-a-Chip (SoC) [2], with a LEON3FT fault-tolerant SPARC V8 GPP connected to SRAM and EEPROM memories and several I/O interfaces via an AMBA bus interconnect. The I/O interfaces provided by this subsystem are more oriented towards control, with local I/O like SPI, I2C, PWM and GPIO among others, and networked I/O like SpaceWire (SpW) and CAN. Furthermore, it provides many advanced functions: house-keeping data acquisition (HK ADC); time-keeping and distribution and memory scrubbing, to name a few.

III. Prototyping Support

The major challenge regarding prototyping the SSDP stems from its sophisticated nature, and is related to the amount of FPGA resources needed to integrate the two subsystems. In order to tackle this issue, a state-of-the-art Xilinx Kintex Ultrascale XCKU060 FPGA [3] is used, which offers enough fabric for the SSDP machinery. The FPGA is mounted together with all the I/O interfaces on a custom board whose block diagram is shown in Figure 2.

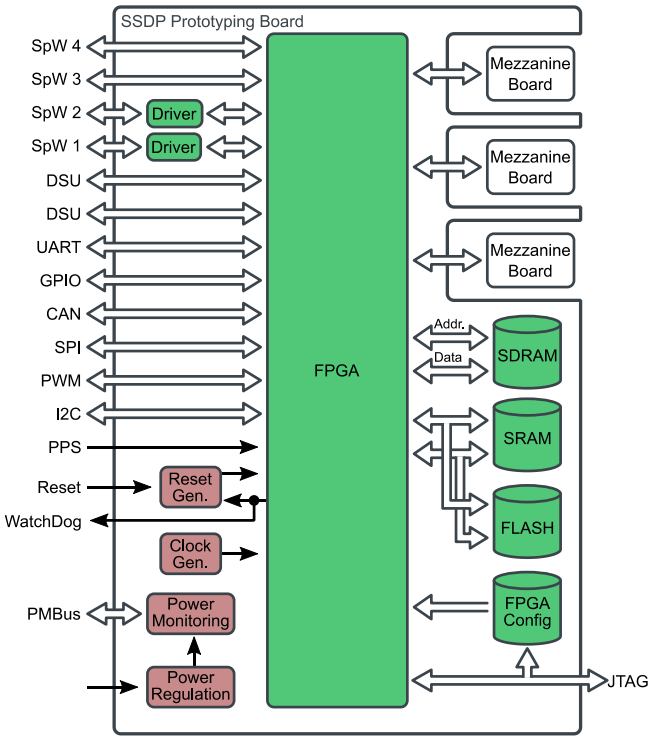

Figure 2: SSDP Prototyping Board Block Diagram

The prototyping board supports all the peripherals and I/O interfaces envisaged in the SSDP architecture. Additionally, connectors based on the FMC standard were added, enabling the expansion of the board functions with modules such as an ADC or DAC devices, as well as allowing the probing of internal signals. The architecture presented in Figure 2 was mapped into a printed-circuit board named SSDP Prototyping Board (SSDP-PROB), and shown in Figure 3.

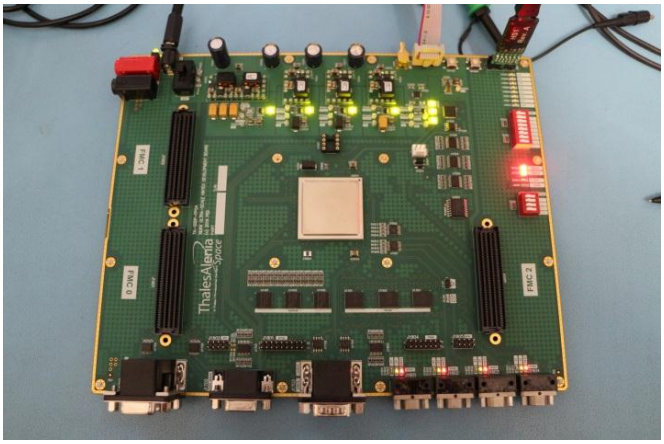

Figure 3: SSDP-PROB - SSDP Prototyping Board

The specification and schematic capture of the SSDP-PROB was performed by TAS-E. The fabrication, assembly and test were performed by Pender Electronics.

IV. Testing and Validation Activities

Testing activities of the SSDP can be divided in three classes[1], based on the objective of the activities:

- *Interface testing*, where one or more interfaces are tested, in order to assess their status of compliance to the (individual) specification;
- *Validation testing*, where an application is used to validate a system or subsystem, usually using several interfaces;
- *Benchmark testing*, where an application or procedure is used to assess the performance of a specific component or set of components (function).

Each of these classes requires different approaches to the testing, including different abstraction levels when designing and implementing the test itself. However, all have a common denominator: the need of some sort of testing support, both at hardware and software level.

1) Interface

The testing of interfaces is a task requiring a very low level of abstraction, for it usually deals with the hardware itself directly. Such activities are usually characterized by activities including configuration and status registers (read and write operations).

Appropriate software support is crucial for this particular activity, for it is the key to increase the level of abstraction of

[1] Radiation testing was left out on purpose, although it can be seen as a particular case of validation.

testing activities. For example, having a software routine which may perform several operations, such as a configuration routine, may enable the design of more powerful tests, and at the same time decrease the amount of test steps needed.

2) Validation

A software application is executed in the SSDP, e.g., filtering, for validation testing activities. The resulting output is then verified to be compliant with a reference model, e.g. output of the same application in a modelling tool like Matlab. Some of the envisaged validation tests are:

- Image processing, with edge-detection algorithms;
- Compression, with algorithms such as CCSDS 122;
- Operating System support, like RTEMS.

These shall be compared against known reference models or golden results, coming from widely accepted reference implementations or standards.

3) Benchmarking

Assessing the performance of a specific component or function is achieved by performing *benchmarking*. A benchmark is a procedure or application which can be used across several different platforms or systems, yet allowing having a common basis for result comparison. In benchmark testing, an application is executed and the time it takes to complete is evaluated. The results can be used to assess the performance of the tested system, and compare it with others systems. An example of a benchmark is the amount of time needed to perform a given operation on a set of data, e.g. the FFT[2] on a set of 1024 samples. In the SSDP scope, the set of benchmarks used for the NGDSP [4] will be used, in order to assess the performance figures of the processing block.

V. Testing and Validation Support

Testing and validation is usually performed by having a *test bench* driving the testing activities, providing stimuli to a Unit Under Test (UUT) and then observing the outputs. Correctness is assessed by comparison with a given reference, which can be based either on specifications of I/O interfaces, or output of reference applications and algorithms.

With the SSDP prototyped on hardware and being the UUT, some sort of Electrical Ground Support Equipment (EGSE) is needed as the Test Bench, in order to provide the necessary stimuli (I/O activities), and capture the outputs for verification. This architecture is depicted in the block diagram of Figure 4.

From the test bench perspective, such architecture requires the provision of both hardware and software components, to (electrically) interface with the UUT and at the same time to (logically) drive the execution of the tests. From the UUT perspective, both hardware and software support is needed: the former is embodied by the SSDP-PROB; the latter in the form of routines to support testing activities, which are described later in this section, or fully-fledged validation applications, described in the following section.

[2] **F**ast **F**ourier **T**ransform

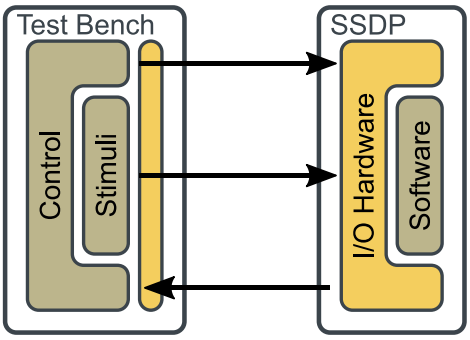

Figure 4: SSDP Testing Architecture Block Diagram

A. Test Bench Hardware

The test setup for the SSDP requires an integrated and flexible EGSE platform, given the diversity of I/O interfaces (*see* Figure 1). A suitable candidate is the PXI platform from National Instruments (NI) [5], which offers the possibility to embed in a single chassis several modular I/O interfaces, together with the computational resources needed to support the execution of the testing activities. A photo of the current setup is shown in Figure 5.

Figure 5: SSDP EGSE Setup for Testing

All the (digital) I/O interfaces of the SSDP architecture are connected to the test bench. Mixed-signal interfaces, such as data acquisition, are *emulated* by resorting either to on-chip mechanism such as a ROM memory, or to an external digital reconfigurable I/O NI PXI module with an FPGA device. Such module allows the emulation of mixed-signal component's digital interface, e.g. the digital word of an ADC, together with the control signals. Furthermore, this module is also used to control the UUT, by issuing signals such as reset.

B. Software

Software support is crucial for the execution of testing and validation activities, as can be inferred from the architecture depicted by Figure 4. Support is required from two different sides: *test bench*, with the logic driving test execution (stimuli and UUT control); *UUT*, with the logic that responds to the stimuli and control signalling, and generates output or actions accordingly.

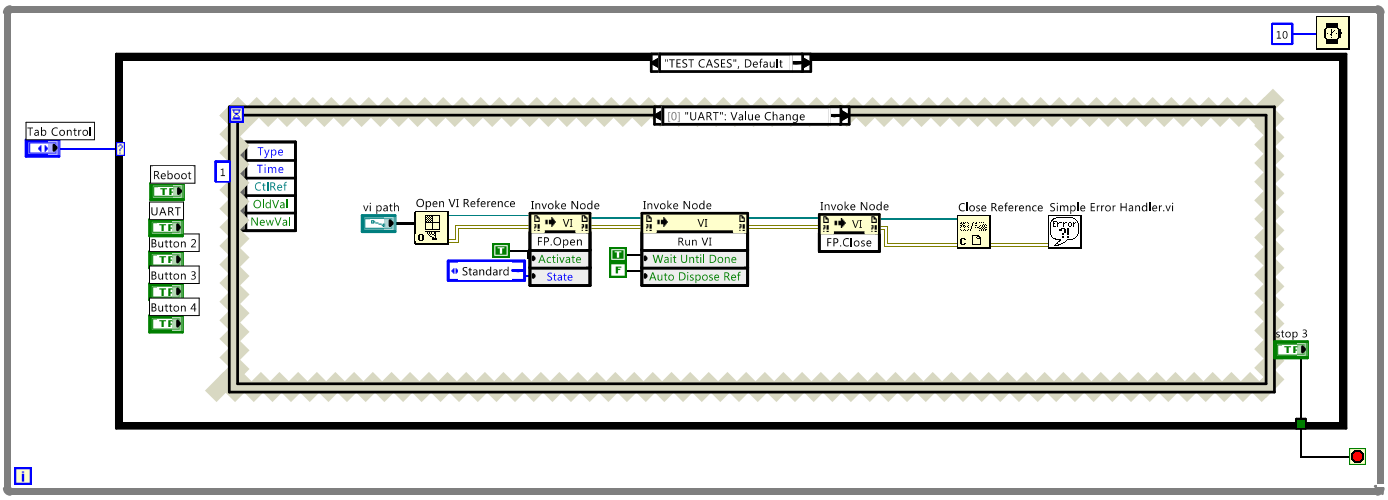

Figure 6: LabView Software Test Design and Entry

1) Test bench

The software driving the tests must provide several levels of abstraction for the design and implementation of the tests. A platform that provides such feature and at the same time is capable of fully exploiting the chosen EGSE hardware platform is NI LabView, an industry standard software w.r.t. test design and execution.

With LabView, tests can be modelled as applications at a high(er) abstraction level and provide support for advanced validation scenarios, based on high-level descriptions, e.g. emulation of a system component, like a mass memory. Such abstraction, however, is based on low-level interfacing with the test components, following a component-based approach, with functions modelled as boxes being instantiated inside other boxes (functions), as shown in **Error! Reference source not found.**.

2) UUT

As pointed out earlier, the UUT software will mainly provide support for test execution, i.e. control of the UUT hardware and stimuli response. Such support comes in the form of the ability to process and exchange telecommands (TCs), which are provided by the Test Bench. A diagram depicting the modelling of a sequence of actions triggered by a TC from the test bench down to the UUT software is shown in Figure 7.

Despite having a seemingly simple function, the UUT software is also capable of performing sophisticated functions, such as data manipulation and peripheral initialization and configuration, needed by the higher-level functions required by the Test Bench application.

C. Resulting Architecture

The resulting test and validation architecture, including hardware and software, is depicted in Figure 8, with some of I/O interfaces represented. Although not depicted, the reconfigurable I/O is also responsible for the (hardware) control of the test activities, e.g. reset.

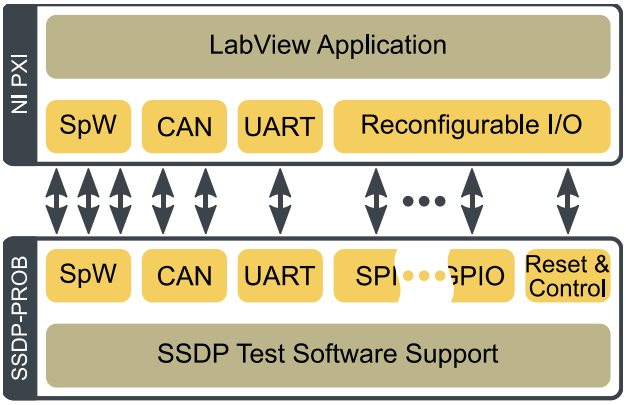

Figure 8: SSDP Testing and Validation Architecture

Such architecture enables the effective test of all the interfaces of a system. Furthermore, this architecture can be reused for Engineering Model (EM) and Flight Model (FM) testing and validation activities, including radiation tests.

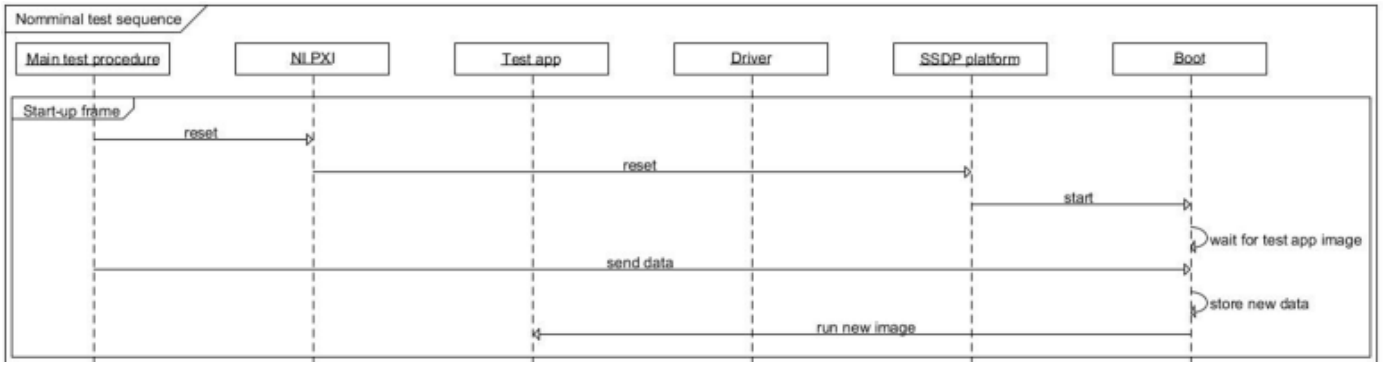

Figure 7: UUT Software Sequence Chart

VI. Concluding Remarks

The Scalable Sensor Data Processor (SSDP) is a next-generation mixed-signal on-board processing ASIC, with a heterogeneous multicore architecture for processing and control activities, having local & networked Input/Output (I/O) and data acquisition and conversion capabilities.

Testing of sophisticated devices like the SSDP requires an appropriate test setup and environment, capable of providing flexibility for the several types of testing activities. Test activities have to be performed at several levels of abstraction, ranging from the hardware low-level modules up to validation as a system, and including also benchmarking activities.

SSDP prototyping is supported by a custom FPGA-based board, with all the needed I/O interfaces, emulation of the digital end of mixed-signal components, like ADCs. Testing and validation activities of the SSDP are supported by a Test Bench architecture based on National Instruments PXI hardware and LabView software.

The same setup is used for all testing, validation and benchmarking activities, with varying software support at the SSDP level, thus encompassing all the required levels of abstraction. Furthermore, Engineering and Flight Model testing and validation can reuse the same architecture for their activities, including radiation testing.

VII. References

[1] Recore Systems, "Xentium® VLIW DSP IP Core - Product Brief," 2016. [Online]. Available: http://www.recoresystems.com/fileadmin/downloads/Product_briefs/2016-1.0_Xentium_Product_Brief.pdf.

[2] Cobham Gaisler, *GRLIB IP Core User's Manual*, 2016.

[3] Xilinx Inc., *UltraScale Architecture and Product Overview*, 2016.

[4] TEC-EDP/2008.18/RT, "Next Generation Space Digital Signal Processor Software Benchmark," ESA, 2008.

[5] National Instruments, "PXI: The Industry Standard Platform for Instrumentation," 2014.