EPSC Abstracts
Vol. 8, EPSC2013-57, 2013
European Planetary Science Congress 2013

Integrated Payload Data Handling Demonstrator

T. FitzGeorge (1), W. Sullivan (1), A. Wishart (1), S. Fowell (2), M. Hann (2), N. Phan (2), C.M. Carr (3), E. Cupido (3), P. Fox(3), T. Oddy (3), A. McGregor (4), Andy Marshall (4), N. Waltham (4)
(1) Astrium Ltd, (2) SCISYS UK Limited, (3) Imperial College London, (4) RAL Space

1. Introduction

An integrated Payload Data Handling System (I-PDHS) is one in which multiple instruments share a central payload processor for their on-board data processing tasks. This offers a number of advantages over the conventional decentralised architecture. Savings in payload mass and power can be realised because the total processing resource is matched to the requirement, as opposed to the decentralised architecture where the processing resource is in effect the sum of all the applications. Overall development cost can be reduced using a common processor. At individual instrument level the potential benefits include a standardised application development environment, and the opportunity to run the instrument data handling application on a fully redundant and more powerful processor.

This paper describes a joint programme by Astrium Ltd, SCISYS UK Limited, Imperial College London and RAL Space to implement a realistic demonstration of an I-PDHS using engineering models of flight instruments (a magnetometer and a camera) and a laboratory demonstrator of a central payload processor which is functionally representative of a flight design. The objective is to raise the Technology Readiness Level (TRL) of the centralised data processing technique by addressing the key areas of task partitioning to prevent fault propagation and the use of a common development process for the instrument applications. The project is supported by a UK Space Agency grant awarded under the National Space Technology Programme SpaceCITI scheme. The demonstration system is set up at the UK Space Agency's International Space Innovation Centre (ISIC) at Harwell and makes use of the ISIC Concurrent Design Facility (CDF).

2. I-PDHS Demonstration System

The I-PDHS demonstration system is an evolution from the earlier PRISM system [1]. I-PDHS uses the latest Integrated Modular Avionics (IMA) software partitioning technology together with reconfigurable firmware on a Field Programmable Gate Array (FPGA) co-processor to provide a powerful and flexible central payload processing system. Instrument data handling functions are implemented as a combination of application task software (for control functions) and firmware (for intensive data processing functions).

IMA provides separate partitions for software applications to run in, preventing software faults from propagating from one partition to another by enforcing Time-and-Space Partitioning (TSP) [2]. Time partitioning means each partition has a fixed static allocation of time to execute, with each partition scheduled in a cyclic manner.

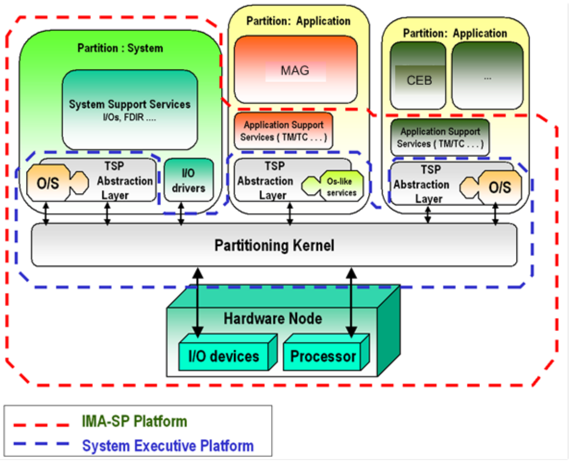

Figure 1: Integrated Modular Avionics.

Space partitioning means each partition has a separated memory through the use of a Memory Management Unit (MMU), preventing accidental corruption of data in other partitions. TSP is provided by a low level hypervisor, such as XtratuM, running on an appropriate processor, such as the SPARC-based LEON3. Within each partition, a tailored form of a real-time operating system, known as a Guest OS, provides standard functions in support of the applications. For I-PDHS, the RTEMS Guest OS is used.

Co-processing firmware is provided on two reconfigurable Xilinx V4 FPGAs [3]. The FPGA fabric is divided into a static area and a dynamic area. The static area includes a network-on-chip centred on the use of SoCWire [4] and is only reconfigured on full device reconfiguration (e.g. after unit power-on). The dynamic area is divided into a number of partitions (PRMs) which contain the application firmware. A PRM is a node in a SoCWire network. Partitions may be dynamically reconfigured and the SoCWire design guarantees that glitches do not propagate to the user sub-system within the PRM, i.e. do not affect the run-time operation of this or other PRMs. FPGA configuration management is performed in control software which runs as an application task in an IMA partition.

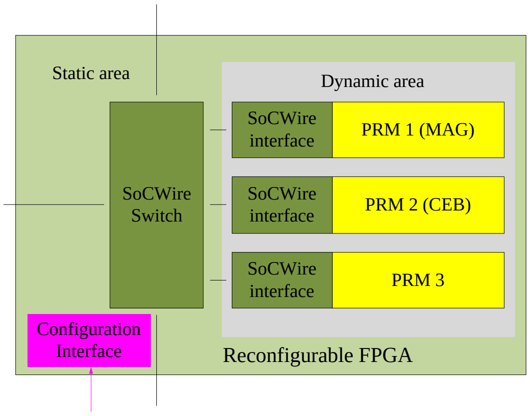

Figure 2: Reconfigurable FPGA structure.

The Magnetometer (MAG) is an in-situ instrument measuring the DC magnetic field vector in the range 0-10 Hz. It is designed to operate continuously at a low data rate, typically < 0.1 kbps, rising to 10 kbps for short bursts In its digital implementation, currently being developed for the Solar Orbiter and JUICE missions [5], the fluxgate front-end produces packetised magnetic field vectors made available over a SpaceWire link to the instrument controller. This approach is particularly suited to be ported on the I-PDHS platform. For this demonstration the instrument controller functions of telemetry and telecommand handling, automatic measurement range switching and overall instrument supervision have been implemented in software, whereas the more computationally-intensive tasks such as raw magnetic vectors filtering, compression and execution of event-triggering algorithms reside in a firmware PRM.

The Camera Electronics Box (CEB) is an engineering model of the unit supplied by RAL Space for NASA’s Solar Dynamics Observatory [6]. The camera uses a 4k x 4k pixel CCD and the CEB transmits each image as a single SpaceWire packet of up to ~ 32 Mbytes. The application firmware written for a PRM reconstructs the image and formats it for direct transmission to the ground segment. The application software handles telecommands and telemetry and provides control and command signals to the CEB.

6. Summary and Conclusions

The project represents a significant step towards realising the advantages of an I-PDHS in a flight mission. The two key areas of partitioning and the instrument application development process have been demonstrated through a challenging practical example with two disparate instruments.

Acknowledgements

The authors thank Peter Allan at ISIC for facilitating this work.

References

[1] Payloads with Resource-Efficient Integration for Scientific Missions (PRISM), Data Systems In Aerospace (DASIA 2009), Istanbul, Turkey, May 2009.

[2] Sharing a Unique Computing Platform between a Payload and a Platform Software using IMA-SP, Data Systems in Aerospace (DASIA 2013), Porto, Portugal, 2013.

[3] Reconfiguration Management and Application Development upon a Dynamically Reconfigurable Processing Module, Data Systems In Aerospace (DASIA 2011), Malta, May 2011.

[4] B. Osterloh, H. Michalik, B. Fiethe and K. Kotarowski, "SoCWire: A Network-on-Chip Approach for Reconfigurable System-on-Chip Designs in Space Applications" in NASA/ESA Conference on Adaptive Hardware and Systems, vol. (AHS-2008), (Noordwijk), pp. 51-56, June 2008

[5] H O’Brien, P Brown, T Beek, C Carr, E Cupido, T Oddy, 2007 Meas. Sci. Technol. 18 3645.

[6] www.sdo.gsfc.nasa.gov, http://aia.lmsal.com/