Dynamically Reconfigurable Processing Module for Future Space Applications

Giuseppe Montano, Paul Norridge, Wayne Sullivan, Chris Topping, Alex Wishart
Astrium Limited, Gunnels Wood Road, Stevenage, SG1 2AS, England
Tel: +44 1438 77 4105, Fax: +44 1438 77 8910, paul.norridge@astrium.eads.net

Frank Bubenhagen, Björn Fiethe, Harald Michalik, Björn Osterloh
IDA TU Braunschweig, Hans-Sommer-Str.66, D-38106 Braunschweig, Germany
Tel: +49 531 391 3743, Fax: +49 531 391 4587, b.osterloh@tu-bs.de

Jørgen Ilstad
Directorate of Technical & Quality Management, Onboard Payload Data Processing Section (TEC-EDP), ESTEC,
PO Box 299, 2200 AG Noordwijk ZH, The Netherlands
Tel.: +31 (0) 71 565 3275 Jorgen.Ilstad@esa.int

1 INTRODUCTION

This paper describes a Dynamically Reconfigurable Processing Module (DRPM) designed to meet the on-board processing requirements of a wide variety of single or multiple instrument payloads in science and earth observation spacecraft missions. The key feature of the DRPM is its reconfigurable core, which uses the latest radiation tolerant, SRAM-based FPGA technology to support in-flight dynamic partial reconfiguration of application firmware.

The demand for increased on-board instrument processing capacity is being driven by the need to process the large volume of data generated by higher resolution sensors. Another driver is the need for greater payload autonomy in missions where the communications lag precludes direct control from the ground, particularly in the case of a multi-instrument payload with a requirement to coordinate remote sensing and in-situ instruments.

In many instrument applications the nature of the data processing task lends itself naturally to implementation in firmware, and the very high gate capacity of the SRAM based FPGAs offer considerable computing power. Further, the ability to dynamically reconfigure all or part of the FPGA allows multiple applications to share a common hardware resource, which offers considerable advantages, both for the system designer and operationally for the instrument end user.

Astrium Limited and IDA Technical University of Braunschweig are currently engaged on the design, development and validation of a system demonstration model of the DRPM under ESA contract. The primary objective of this development work is to raise the technology readiness level (TRL) of the DRPM and its associated Application Development Environment (ADE) to the point where the benefits of this reconfigurable FPGA technology can be made available to flight programmes.

2 SYSTEM REQUIREMENTS

2.1 Payload Processing

The payload context is shown in Fig. 1. It is a centralised architecture, in which multiple instruments share a common processing module (in the limit a single instrument). The DRPM enforces time and space partitioning (TSP) in cases where more than one application tasks are executing concurrently.

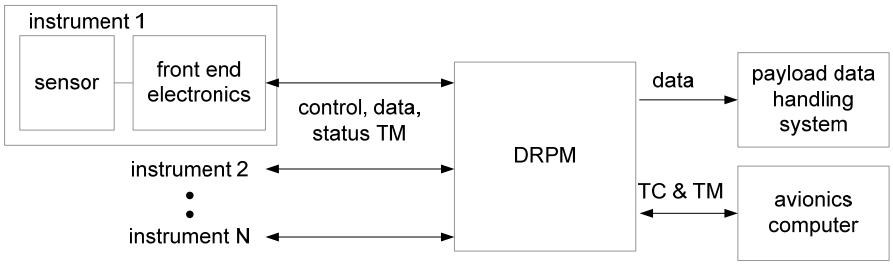

Figure 1. DRPM context

The general instrument model is a sensor and its associated front electronics, linked to a physically separate back end digital processor. The sensor may be passive or active, and comprise a single element or an array. The mixed signal

interface (ADC and DAC) is located with the front end electronics (FEE) for signal integrity reasons, particularly for high rate instruments. In high data rate (HDR) instruments the digital data transfer between the FEE and the DRPM is via dedicated high speed serial (HSS) or SpaceWire links; with low data rate (LDR) instruments, multiple FEEs may be connected to the DRPM in a bus-type architecture such as CAN Bus. Instrument control may be via SpaceWire or CAN Bus. The DRPM interface is linked to the spacecraft avionics through MIL-STD-1553B and to a mass memory module through HSS or SpaceWire.

Digital data processing tasks can be divided into three categories, according to data rate and algorithm complexity, as shown in Table 1.

	Sample level	Frame level	Logic level
Data rates (operations/sec)	high	medium	low
Algorithm complexity/abstraction	low	medium	high

Table 1. Digital processing complexity

The sample and frame level data processing tasks are characterised by highly repetitive, deterministic algorithms operating on large amounts of data, requiring highly accurate timing, and are generally best performed in firmware. The higher level logic functions associated with the control of the instrument are best realised in software running on a general purpose CPU (and would be very difficult to implement in firmware). An important feature of the DRPM and its supporting ADE (see §4) is the flexibility allowed to the developer to partition an application task between firmware and software components.

2.2 Reconfiguration

Reconfigurable firmware is becoming more widespread in terrestrial applications as the technology improves to allow the system level benefits to be realised (e.g. software radio). The DRPM will extend these benefits to space applications. A reconfigurable system offers advantages both for planned mode-dependent functional alterations and for unplanned updates.

In-flight reconfigurable digital hardware allows multiple independent modes to be time multiplexed on the same processing resource. This implies that the processing resource can be sized for the maximum operational load, rather than for the aggregate of every function, with attendant savings in mass, power and design complexity. On the DRPM, application reconfiguration will be a deterministic process in response to the instrument operational mode (i.e. not a fully dynamic system based, for example, on identifiers in the incoming data).

Operationally, there is the major system advantage that the initial firmware loaded at launch may subsequently be updated or complemented with new configurations. This may be due to changing operational requirements, improved processing algorithms, or in response to in-flight calibration or sensor hardware faults. Indeed, the ability to flexibly reconfigure in response to faults offers the possibility to move away from the traditional, mass penalising, prime/redundant paradigm.

A feature of the DRPM is its support for run-time partial reconfiguration of an application's firmware without affecting concurrent execution of other applications. The DRPM can then perform multiple independent tasks (e.g. image compression algorithms, FFTs, etc.) simultaneously and without mutual interruption.

The on-chip communication in the reconfigurable FPGAs which make up the DRPM's reconfigurable core is provided by a System-on-Chip Wire (SoCWire) network. SoCWire is a high speed glitch-free communication architecture, based on the SpaceWire standard. It provides hot-plug network-on-chip functionality with a flexible and high-speed inter-device communication path.

2.3 Modularity

The DRPM is modular in concept. A single DRPM will comprise a reconfigurable core plus a system controller, working memory, configuration memory, avionics and instrument interfaces etc. Multiple DRPMs may be combined to form a unit with more processing capacity or hardware redundancy. A separate power supply module provides all secondary power rails from the spacecraft 28V supply. In a multiple module configuration one of the system controllers would be designated as the master system controller for the unit. The intra- and inter- DRPM communications architecture is network based. The network topology allows an application task to be partitioned across multiple FPGAs and offers unrestricted access to interfaces and memory resources which may be physically located on a separate FPGA, either within the same reconfigurable core or on a separate DRPM.

3 DRPM DESIGN

3.1 Hardware

The DRPM architecture is shown in figure 3-1.

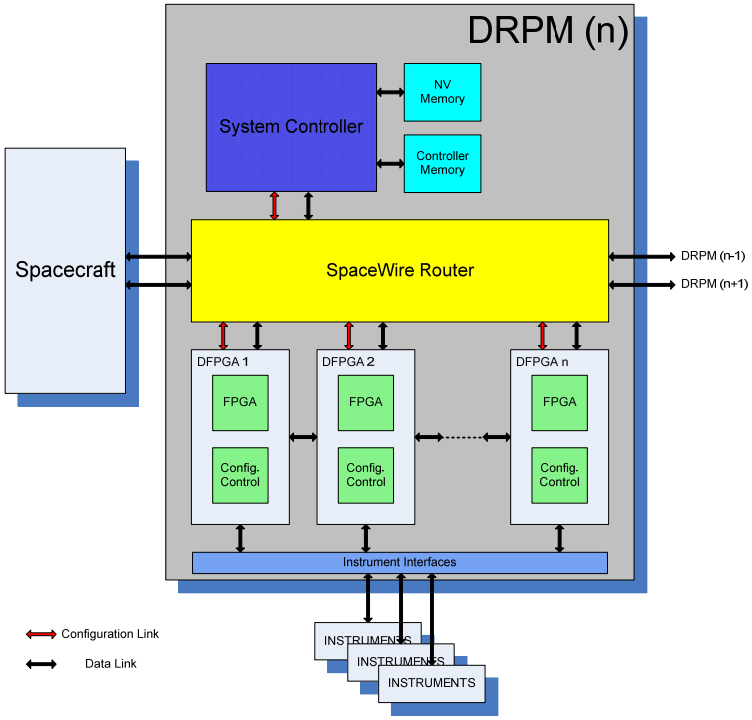

Figure 3-1: The DRPM Architecture Concept

The DRPM concept is built around a number of configurable blocks (reprogrammable FPGAs) to which configurations can be regularly uploaded or changed, either from the central on-board configuration memory or directly via upload (telecommand). A configuration can apply to the whole FPGA or just a part of the device via partial reconfiguration. The latter is possible in several commercial FPGAs such as the Xilinx Virtex4; this is the FPGA chosen for implementing the reconfigurable logic portion of the DRPM demonstrator giving the best balance between reprogrammability, flexibility, device capacity and flight heritage. Partial reconfiguration involves the spatial partitioning of the FPGA device and has the advantages of allowing multiple processes to operate on the FPGA, essentially independently. Reconfiguration in one region will not directly affect the operation of another. Consequently, the FPGA hardware can perform multiple independent tasks (e.g. image compression algorithms, FFTs, etc.) simultaneously and without interruption during a neighbouring reconfiguration.

Interconnect between the various partial FPGA regions, local DFPGA configuration memory and DFPGA working memory (not shown in Figure 3-1) are linked via a SoCWire network. The partial FPGA regions communicate with a SoCWire router switch which will be fixed in the static logic of the Xilinx device. A number of partial reconfigurable regions can access this router and hence the rest of the DRPM.

The reconfigurable core of the DFPGA will be managed by a local configuration controller which loads configurations onto the Xilinx as well as performing tasks to manage the SEU susceptibilities of the Xilinx device (e.g. configuration scrubbing). The configuration controller shall be based on an Actel ProASIC3 FPGA for demonstrator purposes.

The ProASIC3 will house a LEON3 processing core operating at around 15 MHz which will provide localised SoCWire network control functions and could potentially accommodate some user application software tasks. Bridges between the SoCWire network and the SpaceWire network, High Speed links, channel links, analogue interfaces, memory controller, etc. exist as static features of the DRPM unit and are also implemented on the fabric of the ProASIC3.

Overall DRPM management is handled by a second LEON3 acting as the System Controller. This will be implemented in an ASIC, which will incorporate additional interfaces for communication with the Spacecraft (e.g. SpaceWire, MIL-STD-1553B, CANbus) as well as supporting IP blocks (FPU, debug access, etc.)

As noted above, the DRPM architecture allows scalability for processing functions which require more than a single FPGA device – high-speed links allow multiple FPGA interaction on a single DRPM and even several whole DRPM units can be formed into a modular solution for even bigger applications by making use of the SpaceWire routers and linking between modules.

3.2 Software

The bulk of system software will run on the System Controller. This software will be ultimately responsible for carrying out essential DRPM tasks. These include the loading of new FPGA configuration files and software application programs from on-board memory locations, dealing with upload of new configuration/applications from the ground (via telecommand) and ensuring any faults or errors are flagged. New configurations must be verified and, in response to operational modes or ground command, must be selected and loaded as required.

The software design will follow a layered approach (see figure 3-2). This will be based on a real-time operating system (RTOS, e.g. RTEMS) with appropriate drivers for allowing access to the various interfaces of DRPM (SpaceWire, SoCWire, Channel Link, ADC/DAC, GPIO, etc.). Middleware applications will include reconfiguration management tasks and data-handling services. At the top-level, an FPGA management application will monitor FPGA performance, configuration uploading and the status of reconfiguration tasks. User applications are then supported by these software layers. In this kind of architecture, it is vitally important to prevent user application errors from affecting each other and the FPGA management tasks. Ultimately, a time-space partitioning approach (using an enhanced RTEMS) would be used as means of preventing such fault propagation. (An example would be the PRISM project developed by Astrium.)

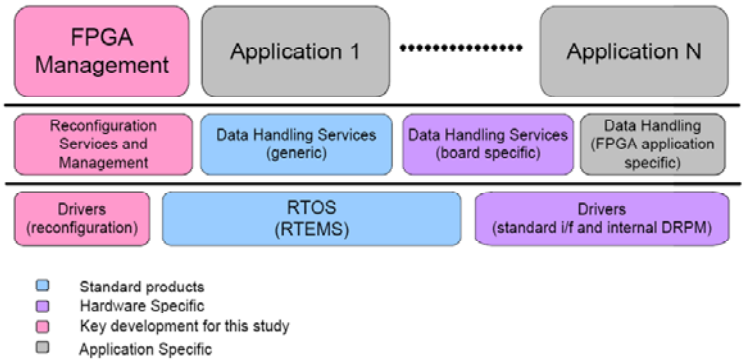

Figure 3-2: System Controller Software – Layered Architecture

4 THE APPLICATION DEVELOPMENT ENVIRONMENT

It is not sufficient to provide system engineers and application developers with new hardware, they also require appropriate infrastructure to allow them to use it effectively. For this reason, the DRPM development work includes a detailed consideration of the associated Application Development Environment. The intention is to overcome reluctance that projects may have over new technology and to ensure that developers can utilise the flexibility efficiently.

The ADE provides a clear process for developing DRPM applications and supports all stages of application production: architecture, simulation, detailed design, verification and test. The architectural-level support provides methodologies for handling partial reconfiguration, for FDIR handling and for partitioning applications. The SoCWire/SpaceWire network underlying the DRPM aids the partitioning process by providing a standard interface for each application component, independent of its implementation details. Architectural tools are provided to ensure that the network can handle the necessary traffic.

To aid simulation, the ADE includes blocks for key components and provides the infrastructure to demonstrate robust reconfiguration. This allows the partial reconfiguration process to be considered in detail, showing in specific cases that the firmware can be altered in a controlled manner. As part of the verification process, the simulations can be re-visited, incorporating the final block designs (both software and firmware).

For the detailed design, the ADE integrates with commonly-used hardware and software development tools. Where necessary, IP blocks and software components are provided for interfacing and implementation of key system functionality.

In addition, the ADE will provide developers with guidelines and strategies that enable efficient use of the DRPM architecture. These will range from architectural patterns for common problems to appropriate methods for detailed implementations. Particular focus is given to the key issues for reconfigurable blocks: switching during partial reconfiguration, block configuration and initialisation, in-situ testing, etc.

5 DRPM SYSTEM DEMONSTRATOR

In the frame of the current ESA study, Astrium Limited and IDA are developing a DRPM demonstration system which will be used to test the main features and operational concepts of both the reconfigurable hardware and the development process. The hardware will be implemented on commercial grade equivalents of the selected flight technologies, ensuring that the demonstrator architecture and functionality is fully representative of a subsequent flight implementation. A set of benchmark applications will be used to quantify the performance of the demonstration system. The benchmarks will be representative of data processing functions typically required in instruments for which the DRPM would be an appropriate back-end processor.

A good example is the on-board processing required for an array detector operating in the thermal infra-red. This is an important spectroscopic band in the 5-15 micron wavelength used for climate and atmospheric studies. Measurements are performed using passive remote sensing from space with nadir and limb sounding Fourier Transform Spectrometer (FTS) instruments. The next generation FTS instruments are expected to exploit the new detector array technology to offer improved spatial resolution with simultaneous spectral acquisition. A high performance on-board processor is required to manage the massive amount of raw data generated by the detector array.

The instrument has three imaging modes, cloud, chemistry and dynamics, characterised by markedly different spatial and spectral resolutions with correspondingly different sample and frame level algorithms implemented in firmware (pixel binning, interpolation, apodisation, FFTs, phase correction, spectrum co-addition etc.). Fig. xx shows how the processing exploits the key features of the DRPM – chemistry and dynamics modes are so computationally intensive that the firmware is partitioned across two reconfigurable FPGAs, and partial reconfiguration is used when switching between chemistry and dynamics modes while cloud mode operates continuously. The instrument control function is implemented in software which performs mode selection, configuration of the detector array etc.

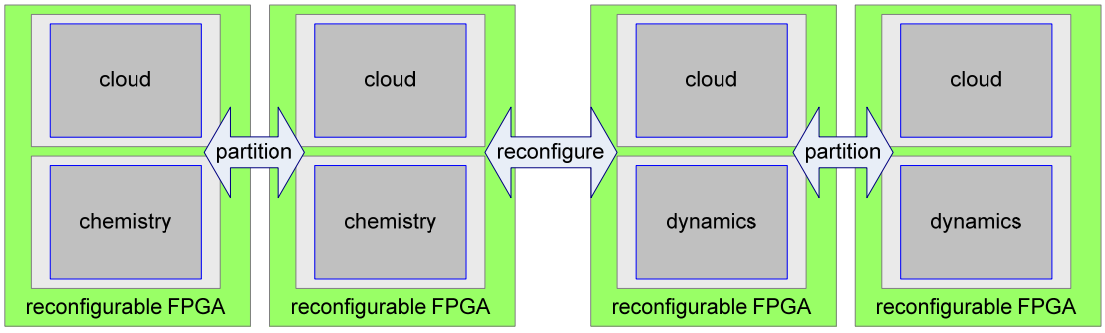

Figure x: benchmark application which exploits the partitioning and reconfiguration features of the DRPM