# Spacecraft Controller On-a Chip with LEON3 (SCOC3)


**AUTHORS**

| Name | Phone | Email | Address/Fax |
|---|---|---|---|
| Roland Weigand | +31 71 565 3298 | Roland.Weigand [at] esa.int | European Space Agency, Microelectronics Section Postbus 299, NL-2200 AG Noordwijk, The Netherlands Fax +31 71 565 6791 |
| Jean-François Coldefy | +33 (0)1 39 45 26 23 | jean-francois.coldefy [at] astrium.eads.net | EADS Astrium 37, avenue Louis Breguet, BP n°1, 78146 Vélizy Villacoublay Cedex, France Fax +33 (0)1 39 45 24 83 |
| Marc Souyri | +33 (0)1 39 45 29 39 | marc.souyri [at] astrium.eads.net | |
| Franck Koebel | +33 (0)1 39 45 27 74 | franck.koebel [at] astrium.eads.net | |


## INTRODUCTION

In commercial electronics with large order quantities, where the IC unit cost is often driven by the silicon area, customer or even product specific ASIC are developed, containing only those building blocks really necessary for the respective application.

In space applications with usually low order quantities, the unit cost is dominated by the non-recurrent (NRE) cost (layout, mask manufacturing), and it is not affordable to manufacture a dedicated ASIC for each system configuration. Diversification for different missions can rather be obtained by developing versatile System-On-Chip devices integrating a multitude of functionalities, which can then be scaled in different operating modes. The European Space Agency supports a (limited) number of chip developments [1], which shall become available as Application Specific Standard Products (ASSP, [2]).

One of them is the Spacecraft Controller On-a Chip (SCOC) which is subject of this paper. After two design studies implemented on reprogrammable FPGA, SCOC1 [3] [4] [5] and SCOC2, using LEON1 and LEON2 respectively, SCOC3 is a System-On-Chip for Spacecraft Control and Data Handling, featuring the LEON3 processor with GRFPU, TM/TC interfaces, various on-board interfaces (Spacewire - SpW, CAN, 1553, UART), time management and event routing. Internal communication is distributed over a double AMBA-AHB bus structure with separate external memory ports, and the two buses are linked by inter-bus bridges. SCOC3 is developed by EADS Astrium under ESA contract 20167/06/NL/FM.

The following aspects will be discussed in this paper:

- Architecture of SCOC3, main subsystems and operational modes
- Internal AHB/APB bus architecture
- Redundancy concept (cross-strapping scenarios)

## ARCHITECTURE

The block diagram of the SCOC3 is shown in Figure 2. There are three main areas of the chip:

1. The processor subsystem (top-right, blue) contains the LEON3 processor with GRFPU and related peripherals: a memory controller, UART and Spacewire interfaces for debug purposes and inter-processor communication tasks, communicating via the CPU-AHB bus.
2. The on-board I/O subsystem (bottom right, yellow), which provides various I/O interfaces (UART, CAN, SpW, MIL1553 BC/RT). The interfaces and the TM/TC part exchange data via a separate IO AHB bus. Data exchanges are essentially of DMA type, the only AHB slave (besides the APB bridges for status, configuration and low data rate interfaces) is the IO memory controller.
3. The TM/TC subsystem (top-left and left, green) provides one packet telemetry encoder (PTME) and one telecommand decoder (TCDA) unit, with MAP interface and all external pins necessary to build a redundant, cross-strapped system. A Housekeeping function (HKPF) allows automatic generation of housekeeping telemetry.

4. Other functions comprise a CCSDS time generator, an event switch matrix allowing a configurable distribution of discrete event inputs and outputs (e.g. time ticks) between the various functional blocks or general infrastructure of the chip, clock and reset distribution or test and debug support.

Spacecraft designer tends to conserve heritage from former missions. Systems, once validated or even flight proven, are kept for future designs as long as possible. However, when components become obsolete, or new requirements arise, individual subsystems may have to be replaced, while keeping other parts unchanged. To respond to this need, the SCOC3 design is scalable to work in four different operating modes. Depending on the system environment, part of the chip can be disabled by bootstrap pins. Keeping the disabled part in reset state and gating its clock, (functional) interference and unnecessary power consumption are avoided.

The operating modes are:

1. Full mode: all parts of the chip are available. SCOC3 acts as spacecraft main computer, does TM/TC and payload data handling and thanks to the powerful processor can take over (part of) payload data processing.
2. Processor and I/O: the green part of Figure 2 is disabled. SCOC3 can be used for payload data handling and processing, next to a separate TM/TC subsystem.
3. TM/TC only: In Figure 2, the processor (blue part) and the yellow I/O interfaces (UART, CAN, SPW, 1553, bottom right) are disabled. Note that the IO memory interface has to stay alive, because TM/TC needs buffer memory. SCOC3 can replace a conventional TM/TC subsystem using 'discrete' components such as the VCM, VCA and PTCD ASICs. Special care must be taken to ensure a proper configuration and management of all the active subsystems through the APB buses at the absence of the embedded processor.
4. Processor and TM/TC: In Figure 2, the yellow I/O interfaces (UART, CAN, SPW, 1553, bottom right) are disabled. SCOC3 can be used as main computer next to a separate on-board communication structure, which may consist e.g. of a Spacewire network with routers and RTC bridges to local CAN, or a wireless network etc.

**ON-CHIP BUS ARCHITECTURE**

The performance offered by a single on-chip bus in a system with processor and extensive IO traffic does not fulfil the requirements for latency and throughput. Backbone of on-chip data transfer is therefore a system of two cross-coupled AHB buses (CPU AHB and IO AHB). This concept also leads to a reduced number of bus agents on each bus and therefore improves timing (maximum clock frequency), and it allows operating both buses (and the subsystems attached to them) at a different clock frequency. The CPU AHB bus can be operated at an integer (1, 2, 3, 4) multiple of the IO AHB clock, allowing the optimisation and trade-off of performance versus power consumption depending on the application requirements.

Target maximum clock frequencies are 120 MHz for the CPU AHB and 50 MHz for IO AHB allowing a performance of up to 100 MIPs for LEON3-FT and a bandwidth of 1.6 Gbit/s on the IO AHB bus.

There are two bridges coupling the two AHB. The HDMA is a self-standing DMA engine, being a master on both buses, shuffling data back and forth in background, once it has been configured by the CPU with start/end address and block length. This is obviously a too heavy procedure for random accesses (e.g. the processor accessing configuration registers of the peripherals located on the IO bus), and a write-through bridge AHBR has been created. AHBR, slave on CPU AHB and master on IO AHB, maps the complete IO AHB address space into the CPU AHB address space. As a result, seen from the CPU perspective, the complete chip is accessible, and the dual bus structure is transparent (apart from possibly different wait-states, which are handled in hardware).

Arbitration of the CPU AHB is prioritising, where the CPU has highest priority (TBC overridden only by masters used in debug mode). For the IO AHB bus, a round robin algorithm has been selected. Together with a known maximum burst length of all masters, minimum throughput and maximum latency can be guaranteed for each master. Since the arbitration scheme is programmable, allocating multiple grant slices (per roundtrip) to individual masters, the bus performance can be scaled according to the system requirements.

Both AHB buses have their own memory controller (LEON3 MCTL, IOMCTL) allowing simultaneous data transfer with external memory. Both support SDRAM, including hardware Reed-Solomon block error protection. As the IOMCTL will be mapped starting at address 0 in the CPU address space, the PROM is located in the IOMCTL, and the CPU boots from that PROM, across the AHBR bridge. The LEON3 MCTL has no PROM functionality and its address space starts only at 0x40000000.

For configuration, control and status operations, three APB buses exist according to the three subsystems of the chip, each of them is reached via an AHB/APB bridge (APBR). The CPU can reach all devices on the three buses by simple load/store instructions. A fourth small APB bus is not shown in Figure 2: the House Keeping Function (HKPF) as APB master retrieves in background timing and TC information and directly inserts the status packets to the PTME module.

This structure in principle had been selected already for the SCOC1 design [3]. However, a few modifications should be noted.

1. The CPU AHB is discharged from possibly high data rate IO interfaces (SpW, PCI) to improve performance (throughput and timing) of the CPU bus.
2. Spacewire interfaces are redesigned to have only one single master instead of two masters and one slave on the AHB bus to improve timing. Arbitration between RX and TX data transfer is now done inside the SpW interfaces.
3. Addition of the AHBR write-through bridge as mentioned above.

**REDUNDANCY CONCEPT**

SCOC3 in its different operating modes may be used in various system architectures. We can consider a simplex architecture, or hot and cold redundant architectures, using SCOC3 in full mode (CPU, TM/TC and IO). There are also architectures using SCOC3 in a classical 2 chips configuration, including one SCOC3 chip as processor function and one SCOC3 chip as TM/TC function.

The use of the interfaces of SCOC3 for data communication or cross-strapping in different architectures is summarised in the table hereafter.

| Architectures | Simplex | Duplex (hot) | Duplex (cold) | Duplex (two CPU/IO chips and two TM/TC chips) |
|---|---|---|---|---|
| CPU SpW | Any use (ground debug link,…) | Any use but mainly Inter Processor Link between the CPU SpW of the two SCOC3 | Any use (ground debug link, context memory,…) | Any use as direct connection between SCOC3 (CPU/IO) to SCOC3 in TM/TC operating mode |
| MAP Interface | Connection to MAP internal on boards | Cross-strapping of the received TC between the MAP interface of the two SCOC3 | No cross-strap and data pass over IO SpW for context storage | Cross-strapping of the received TC for the two SCOC3 in TM/TC operating mode |
| CLCW | N/A | Cross-strapping of CLCW word between the two SCOC3 | No cross-strap and data pass over IO SpW for context storage | Cross-strapping of CLCW word for the two SCOC3 in TM/TC operating mode |
| IO SpW | Any use (IO connection link,…) | Cross-strapping between the two SCOC3 | Cross-strapping to a Context memory | Connection between SCOC3 (CPU/IO) to SCOC3 in TM/TC operating mode |
| TM/TC SpW | User TM external access | User TM external access | User TM external access | User TM external access |

An example of hot redundant architecture is given on Figure 1.

The Inter Processor Link (IPL) may be used in different ways depending on the application and the software complexity. The main purpose of this link is usually to allow the patch of the EEPROM containing the flight software of the redundant CPU and the dump of the memory of the faulty CPU after reconfiguration in order to help finding the cause of the reconfiguration. In the case of CPU active redundancy, it can allow the transfer of periodic context by the nominal CPU to the redundant one.

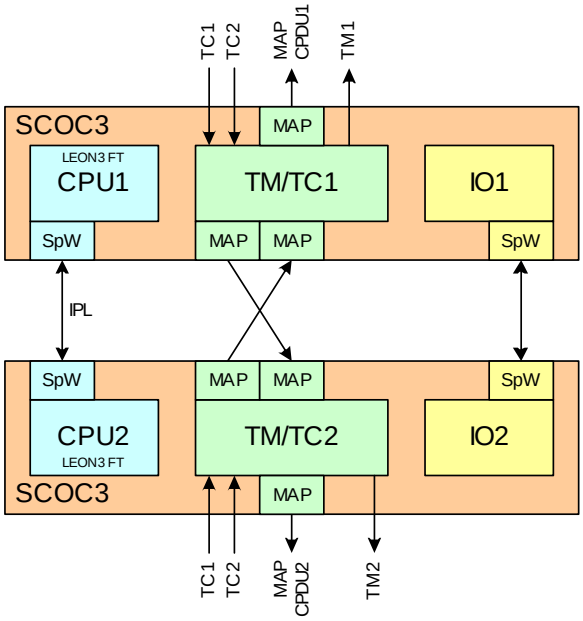


**Figure 1: Example of hot redundant architecture**

## METHODOLOGY

- Co-design hardware and software using a classical (paper) specification, architecture and VHDL ASIC development in interaction with low level software.
- VHDL simulation of the whole ASIC with emulation of the ASIC environment (memories, events, links,…) and elementary software.
- Breadboard on XILINX FPGA of the whole ASIC with test-bench environment and low level software in order to have a functional validation for hardware and software.

## CONCLUSION

SCOC3 is being developed to become a standard component, allowing a high degree of integration covering central spacecraft control/management, telecommand /telemetry and payload control/data handling/processing. It is also scalable allowing its usage in legacy system designs, where only subsystems shall be replaced by a new component.

SCOC3 is currently (1. half of 2007) in the architectural (VHDL) design phase, and if funding for ASIC manufacturing is available, chips will be available by mid 2009.

For future designs, integration of other subsystems can be considered, in particular AOCS functionalities (e.g. interfaces to and dedicated processing blocks for star trackers or GNSS receivers). In addition, further base-band processing and analog interfaces (ADC/DAC) for the RF transponder chains (TM/TC) could be included.

## REFERENCES


[1] http://microelectronics.esa.int/components/comppage.html
Space components developed by/under contract of ESA

[2] http://en.wikipedia.org/wiki/Application_specific_standard_product
Definition of Application Specific Standard Product (ASSP)

[3] http://microelectronics.esa.int/soc/scoc/scoc_synop_07.pdf
SCOC1 block diagram

[4] Tests and Development results of a System-on-a-Chip for Satellite Data Handling, Marc Souyri, Marc Lefebvre, Roland Weigand, DASIA 2003

[5] http://conferences.esa.int/03C20/s1a-01.pdf SCOC - A Spacecraft Controller On-a-Chip, R. Weigand, M. Souyri, M. Lefebvre, Workshop on Spacecraft Data Systems (SDS) 2003

**Figure 2: Block diagram of SCOC3**