TRP activity
Building Blocks for System on a Chip

Call-off order No.3

Development of a Spacecraft Controller on a Chip

EXECUTIVE SUMMARY

EUROPEAN SPACE AGENCY CONTRACT REPORT

The work described in this report was done under ESA contract. Responsibility for the content resides in the author or the organisation that prepared it.

ESTEC Contract No. 13345/98/NL/FM

Prepared by : Marc LEFEBVRE
EADS Astrium France
37, avenue Louis Bréguet - B.P. 1
F-78146 Vélizy Villacoublay Cedex

ESA STUDY CONTRACT REPORT

No ESA study Contract Report will be accepted unless this sheet is inserted at the beginning of each volume of the Report

* ESA Contract N° 13345/98/NL/FM	SUBJECT **Building Blocks for System on a Chip - COO3** **Executive Summary**	NAME OF CONTRACTOR **EADS Astrium**

* ESA CR ( ) N°	* STAR CODE	No of Volumes : 1 This is volume n°1	CONTRACTOR'S REFERENCE : R&D-SOC-NT-338-V-ASTR Issue 0 Rev 0

ABSTRACT:

The work of COO3 was necessary to evaluate very high integration levels of microelectronics coming with the increasing demand of performance in terms of processing, power and mass.

Increasing amount of functionalities integrated into the same ASIC implies that the development of a whole system relies on the use of pre-developed and pre-tested IP functions. The end target is to be able to construct a whole system by “simply” integrating IP blocks

The goal of the Call of Order 3 “Development of a Spacecraft Controller on a Chip” (SCoC) of the Frame Contract “Building Blocks for System on a Chip” is to demonstrate the feasibility and to evaluate the methodology and development complexity - of a System-On-Chip (SOC) device composed of IP blocks coming from different sources.

The requirements and functionalities of the SCoC come from the previous Call of Order 2. It was then decided to implement a demonstrator of this SCoC into a commercial reprogrammable FPGA mounted on a Compact-PCI board, making the design available for a validation of the performances and functionalities.

Gate level design methodology currently used for ASIC development has to shift one step over to reach a state of block level design methodology, focusing on the validation of the whole system design.

This work described in this report was done under ESA contract. Responsibility for the contents resides in author or organisation that prepared it.

Names of authors:

Marc LEFEBVRE (EADS Astrium)

ESA STUDY MANAGER Mr Roland WEIGAND DIV: TOS-ESM	ESA BUDGET HEADING	

*Sectors to be completed by ESA

Title : ESA 13345/#3 : Building blocks for system on a chip

SCoC

Executive Summary

	Name and Function	Date	Signature
Prepared by	**Marc LEFEBVRE** ***SCoC Design Manager***		
Verified by	**Jean-François COLDEFY** ***Digital Electronic Expert***		
Approved by			
Authorized by	**Marc SOUYRI** ***ASIC/FPGA Design Expert***		
Application authorized by			

Document type	Nb WBS	Keywords

DOCUMENT CHANGE LOG

Issue/ Revision	Date	Modification Nb	Modified pages	Observations
0/0	30/09/03			First Issue
0/1	30/06/04			Corrections of Final Report integrated in the executive summary

PAGE ISSUE RECORD

Issue of this document comprises the following pages at the issue shown

Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.
all	**0/1**										

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 CONTEXT AND RELEVANCE OF THE STUDY

The Frame Contract « Building Blocks for System on a Chip » has been established with ESTEC Microelectronics section to anticipate the necessary evolution of the density and complexity of microelectronics designs.

There are three Call off Orders (COO) within the SoC frame contract. This document reports the work performed for COO3: “Development of a Spacecraft Controller on a Chip”. It is based on requirements and functionalities specified in COO2 (refer to AD4).

A final report is also available for this study.

The objective of the Call of Order 3 (COO3) study has been then established to :

- define a complex SCoC functional architecture
- develop a complex SoC considering the use of existing IP cores and a new methodology for design and integration,
- develop a hardware demonstrator of the SCoC to provide a platform for integration and performances evaluations.

2 DESIGN OBJECTIVES AND CONSTRAINTS

2.1 BACKGROUND

The telecommunication satellite constellation market and the small satellite market drive to dramatically decrease spacecraft equipment weight, consumption budget and recurrent price. The emerging micro satellites and nano satellites generation generates also a demand for much more integrated equipment. ESA and industrials developed in the years 80 and 90 sets of ASICs to reduce the size of electronics. Functions such as the VCA, VCM and PTD used to build a CCSDS TM/TC system, bus interfaces such as MACS, OBDH, 1553, or error correction chips (EDAC, Reed-Solomon, Viterbi…) were manufactured in ASICs.

In the past years, the ASICs developed under ESA contracts were offered to European industrials as ASSP. Most of these chips were designed by using the VHDL language creating software macros that can be reused under certain conditions. The idea by now is to merge all these available blocks in a single ASIC called “System-on-a-Chip” that would be able to perform a large number of the Data Management System functions of a platform.

The System-On-a-Chip is the beginning of a new approach for Space system, leading to the ASIM (Application Specific Integrated Microsytem) emerging concept. The ASIM will also contain:

- micro mechanical (micro-motor, micro intelligent captor, etc…)

- micro optical (e.g. micro-camera 1024 x 1024 Pixels CCD for Space observation)

2.2 DESIGN OBJECTIVES

The system-on-a-chip approach is a necessary innovation for the European industry. It is obvious that the number of gates per ASIC will continue to grow in the future as it can be seen for commercial applications. There are at least two main domains of applications that require ASICs in Space, which are:

- The digital signal processing for which the available technologies are far from the requirement in terms of gate and in term of limitation of the power dissipation. These functions are by now far from the System-On-a-Chip concept since they integrate for some of them hundreds of ASICs.
- The Data Management Systems composed of elementary ASICs designed by the industrials or bought as ASSP after their development under an ESA contract. These functions can be integrated together with the microprocessor, and this is the next step that is faced the European Space industry to reduce power and mass budget.

By now Data Management Systems are built around a microprocessor, backplane bus interface, external serial bus interfaces, positioning functions such as GPS, TM/TC functions... It would be profitable to integrate most of these functions on a single die. The objective is to integrate most of the peripheral ASICs and the controller in a single chip, keeping in mind the Failure Detection, Isolation and Recovery (FDIR) scheme in the implementation.

The developed System-On-a-Chip can be used in most of the satellite platforms, and especially it would be convenient to use for small satellites.

The System-On-a-Chip developed in this activity is a one chip Spacecraft Control System FPGA comprising the following functions:

- LEON SPARC CPU with FPU
- Backplane bus controller (PCI)
- Spacewire Bus Interfaces
- CCSDS packet telecommand and telemetry system
- TM Housekeeping generation
- 1553 System bus controller/bus monitor/remote terminal
- Time management system

The subject of this study is at first to validate the concept of a System-On-a-Chip approach. The work to carry out in this study is to practically try out the problems posed by the System-On-a-Chip approach for a Space application. It follows as a basis the classical development plan of ASIC that is defined by ESA. Since an ASIC foundry of such size is very expensive, a first step using large FPGA for prototyping of the System-On-a-Chip is made in this phase of the contract. The System-On-a-Chip is implemented on a demonstration board, and the ASIC is integrated in a XILINX VIRTEX-E FPGA.

Page intentionally left blank

3 MAIN TOPICS

3.1 SCOC ARCHITECTURE AND PERFORMANCES

The architecture of the SCoC is developed with modularity to facilitate the insertion of new blocks or the suppression of others. This modularity is based on the internal bus structure and on standardized services provided to the blocks, like interrupts management and time distribution, independently of their intrinsic functionality.

Internal bus performances (Figure 1)

The performances of each SoC interface function taken individually can be estimated and controlled. The architecture choices for the internal block implementation have only local repercussion. Definition of the SoC internal bus architecture has a drastic impact on the performance of each block taken individually, and also on the overall performance of the system.

The analysis of a standard system requirement drives the SCoC bus architecture selection and leads to a choice of 2 AMBA AHB busses. The performances of this architecture for the targeted application are good. Hardware evaluation platform provided with the BLADE board is an excellent tool for architecture implementation and evaluation.

Data Transfer

The integration of complex IO interfaces within the same chip as the CPU needs a study of data exchanges mechanisms between the peripherals. In the SCoC, DMA mechanisms have been spread over each function. This allows having high and efficient transfer rates with a little use of the CPU.

SCoC Control (Figure 2)

The SCoC is mainly controlled through the APB bus. All IP cores are connected to this bus for register access. This structure allows the control of the SCoC by any AHB master connected to the CPU AHB bus. Basically the LEON CPU core acts as the main system controller, but the SCoC can also be configured as a PCI agent and be controlled the PCI system controller (or any master agent).

Time Management (Figure 3)

The Time management and time distribution is not a specific issue of the SCoC. But the SCoC has to anticipate the need of synchronisation between peripherals in its architecture, as the design can be frozen in an ASIC. A synchronisation switch matrix is implemented into the SCoC to adapt the function to specific system requirements.

Architecture evolution (Figure 4)

The basic architecture of SCoC has been evaluated in terms of performances. Bottlenecks have been identified and are subject of recommendations for a future implementation of a SCoC in hardware. Evolutions mainly consist in an upgrade of LEON CPU and a modification of the internal bus architecture to offer simultaneously higher performances in term of computation and data transfer rates.

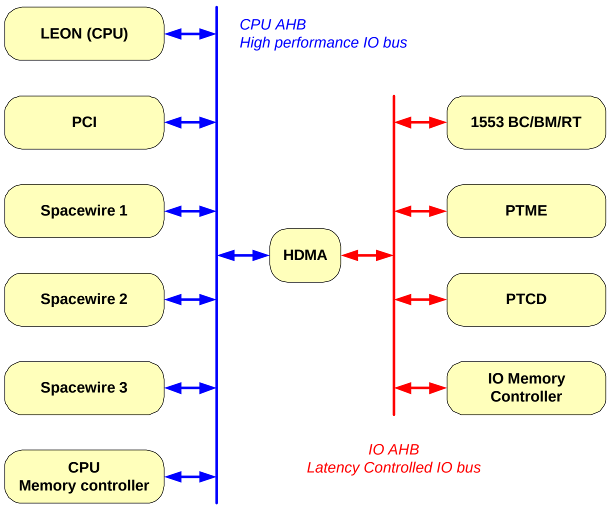


Figure 1 : SCoC internal bus structure

Two AHB busses, the CPU AHB for high performance, flow controlled interfaces, and the IO AHB for medium data rate interfaces, form the basics of the SCoC internal bus structure. LEON CPU is the highest priority master on the CPU bus, and then the available bandwidth for PCI and Spacewire shall be allocated by software control of the CPU bus activity.

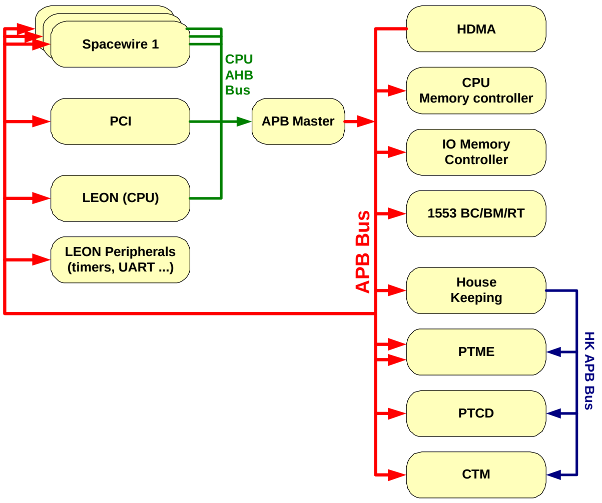


Figure 2 : Control of the APB bus

The APB Bus allows accessing to all registers implemented in the different IP cores. Any master on the CPU AHB Bus can access to the APB bus as as Master through the APB Master interface. It is then possible to control the SCoC through the PCI interface for example

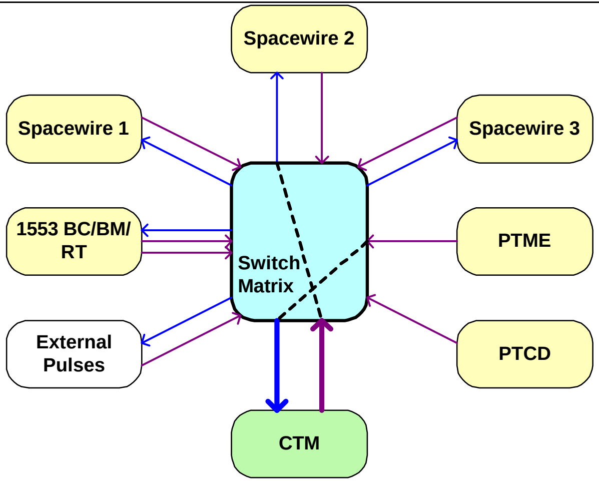


Figure 3 : Switch Matrix for internal / external synchronisation

Integrated IO IP cores within the same chip require adding flexibility for synchronisation and time distribution. The figure indicate how the switch matrix can be used to connect for example the Spacewire time code generation input to the CTM PPS output and the PTME time code output to the CTM event tagging input

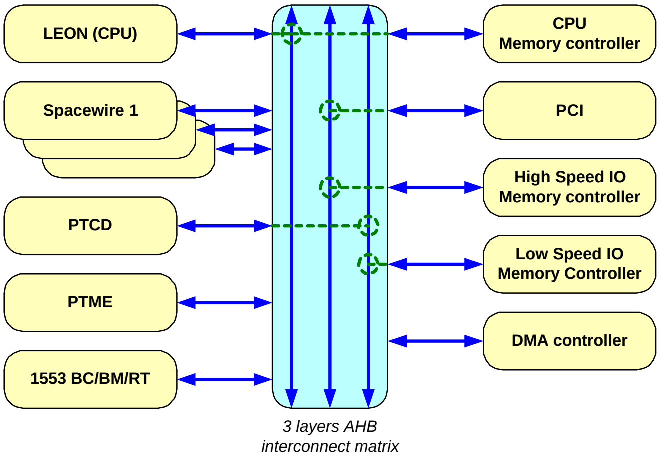

LEON (CPU)
Spacewire 1
PTCD
PTME
1553 BC/BM/RT
CPU Memory controller
PCI
High Speed IO Memory controller
Low Speed IO Memory Controller
DMA controller
3 layers AHB interconnect matrix

Figure 4 : Bus evolution 2 – use of Multi-layer AHB

The 3-layer AHB arbiter allows at the same time LEON reading instruction to its CPU memory, PCI writing to the High-Speed IO(HSIO) memory and PTCD accessing to the Low-Speed IO (LSIO) memory. At the next cycle, the interconnect matrix can for example connect the CPU to the HSIO memory to retrieve data previously written through the PCI.

3.2 DISCUSSION ON THE IP CORES

We can distinguish 4 main sources for the IP cores used within the Spacecraft Controller on a Chip : IP provided by ESA, commercial IPs, Astrium ASIC design modified as IP cores, and IP fully developed in the frame of this activity.

LEON : CPU core (Figure 5 and Figure 6)

LEON-1 SPARC V8 core developed by ESA and Gaisler Research provides a powerful highly configurable CPU core available for integration within System on a Chip. Nevertheless, while the configuration is efficient, it is not adapted to the use within a SoC, as LEON core can be seen itself as a SoC.

CTM : CCSDS Time Management

The CTM provides all the functionalities for the on board CCSDS Time Management by integrating time keeping, alarms and pulse per second generation, event time tagging, time correlation and re-synchronisation. Its use within a complex test structure in A3M study confirmed the effectiveness of this system.

PTME : Packet Telemetry Encoder

The PTME provides all the functionalities of the ground telemetry system, integrating the previous design of ASSP VCA, VCM, Rescue and Turbo Encoder. The functionalities have been wrapped into a single IP core interfacing with a single external memory. Integration of PTME into SCoC has been performed without any major problem, but the PTME internal design contains the worst case combinatorial path limiting the overall performances of the design.

PCI bridge (Figure 7)

The PCI bridge integrated into the SCoC is made of 2 parts : a PCI core handling PCI protocol corresponding to the InSilicon (now Synopsys) PCI core and a ESA developed PCI wrapper providing AMBA AHB and APB interfaces to this commercial core. The PCI interface has been evaluated by performing exchanges between 2 BLADE boards, the PCI being monitored by a commercial PCI analyser from VMETRO. The PCI core has been implemented as a black box into the SoC.

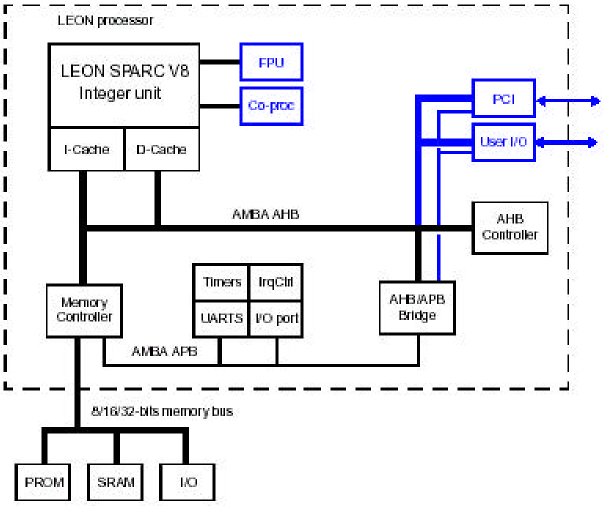


Figure 5 : Internal structure of LEON1

LEON processor embeds all functionalities of a CPU core. The main difficulty is that LEON can be considered as a SoC and is hardly integrated within another SoC design such as SCoC.

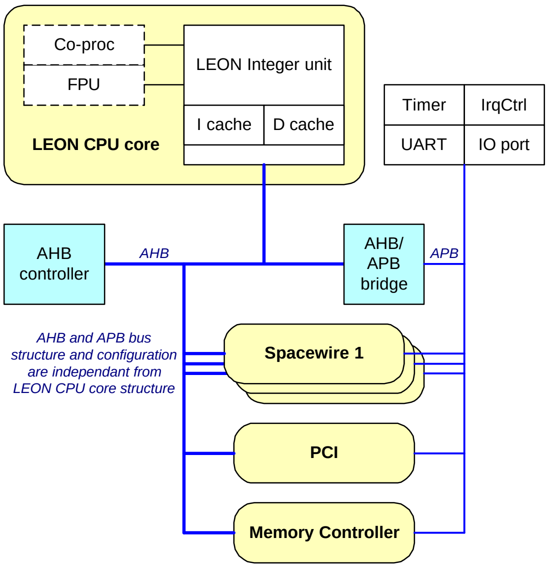
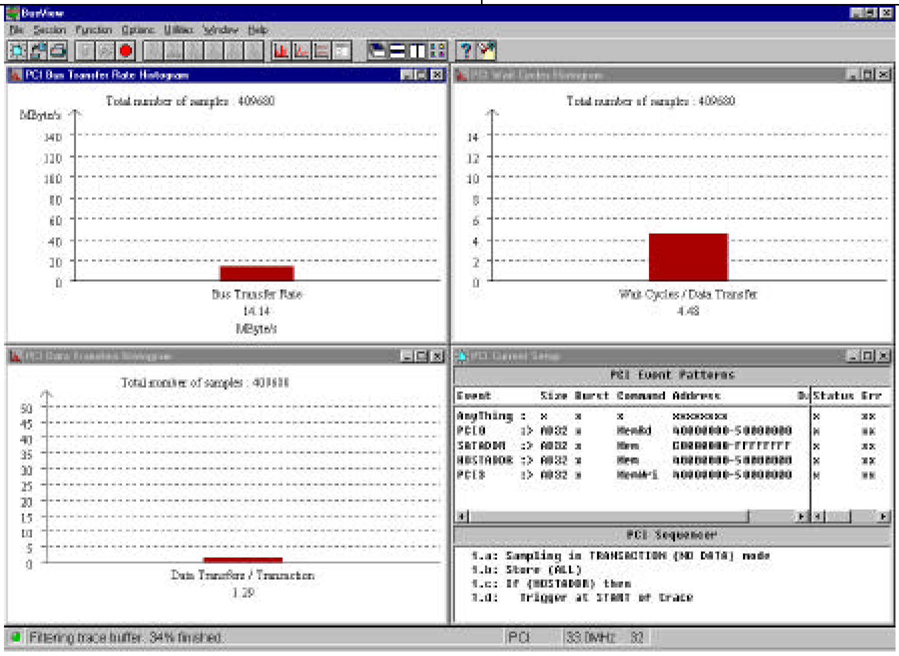


Figure 6 : Evolution of LEON structure

Extracting the configuration of AMBA bus structure from the LEON CPU core configuration eases the development of the SoC structure, as well as the possibilities for architecture exploration and evolution. A potential LEON CPU core functionalities evolution can be integrated more easily.


BusView
File Session Function Options Utilities Window Help
PCI Bus Transfer Rate Histogram
Total number of samples : 409680
MBytes/s
140
120
100
80
60
40
20
0
Bus Transfer Rate
14.14
MBytes/s
PCI Wait Cycles Histogram
Total number of samples : 409680
14
12
10
8
6
4
2
0
Wait Cycles / Data Transfer
4.48
PCI Data Transfers Histogram
Total number of samples : 409680
50
45
40
35
30
25
20
15
10
5
0
Data Transfers / Transaction
1.29
PCI Current Setup
Filtering trace buffer: 34% finished
PCI
33.0MHz
32

```
PCI Event Patterns
Event          Size Burst Command Address            D: Status Err
AnyThing :     x    x     x       xxxxxxxx              x      xx
PCI0      :>   A032 x     MemRd   40000000-50000000     x      xx
SRTADDR   :>   A032 x     Mem     C0000000-FFFFFFFF     x      xx
HOSTADDR  :>   A032 x     Mem     40000000-50000000     x      xx
PCI3      :>   A032 x     MemWr1  40000000-50000000     x      xx

PCI Sequencer
  1.a: Sampling in TRANSACTION (NO DATA) mode
  1.b: Store (ALL)
  1.c: If (HOSTADDR) then
  1.d:    Trigger at START of trace
```

Figure 7 : Performance on the PCI bus for exchange between 2 BLADE boards

Bus performance analysis for 2 board performing symmetric DMA transfers on the PCI bus are relatively average, in terms of data rate (14.14 Mbytes/s) burst length (most of transfers are 2 words length) and wait states.

PTCDIP : Packet Telecommand Decoder (Figure 8)

The PTCDIP has been developed from the PTCD ASIC design (which is a transfer from the GPS PTD ASIC to the MITEL PTCD ASIC). The VHDL code of this ASIC has been modified to integrate AMBA AHB and APB interfaces, making it suitable for integration in a SoC. In order to preserve a synchronous design, the ASIC code has been modified instead of developing a wrapper.

IP1553 : Mil-Std-1553 controller

The IP1553 design is based on the former MAT53 Astrium ASIC, which integrates the Bus Controller, Remote Terminal and Bus Monitor capabilities on the Mil-Std-1553 bus. The IP1553 design is based on the reuse of core functionalities of the ASIC and a modification of the user interface for compatibility with the AMBA AHB and APB busses

Spacewire (Figure 9)

The Spacewire IP core allows achieving transfer rates at up to four times the system frequency. The VHDL structure of the IP core clearly separates the Spacewire protocol management from the host interface, allow re-using either the full IP or only the core. The host interface provides intelligent DMA services that can be tailored to be adapted to specific system requirements.

HKPF House Keeping “Packetizer” function

The HKPF function automatically generates telemetry packets transferred to the virtual channel 0 of the PTME. The telemetry sources are internal – telecommand report, time packet, context RAM dumping - for the SCoC design study, but the design can be upgraded to integrate automatic analog and digital status acquisition for example.

AMBA small IP

Small IP specified from the SCoC functional specification has been developed to enhance the SCoC internal architecture. These blocks are the A**H**B **D**irect **M**emory **A**ccess (HDMA) function and the IO memory controller function (IOMCTRL).

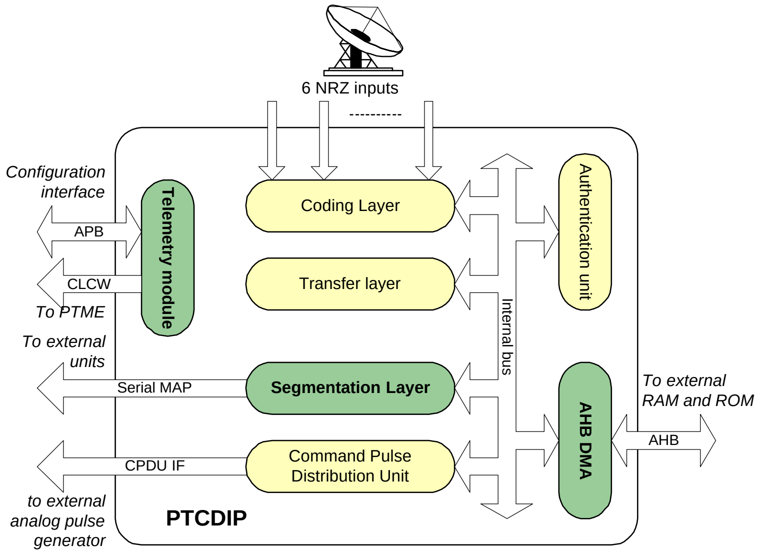


Figure 8 : Modifications from PTCD to PTCDIP

The modifications of the ASIC design were concentrated into 3 blocks (dashed on the figure). The core blocks handling TC protocol are not modified, thus avoiding insertion of bugs in the protocol management functionality.

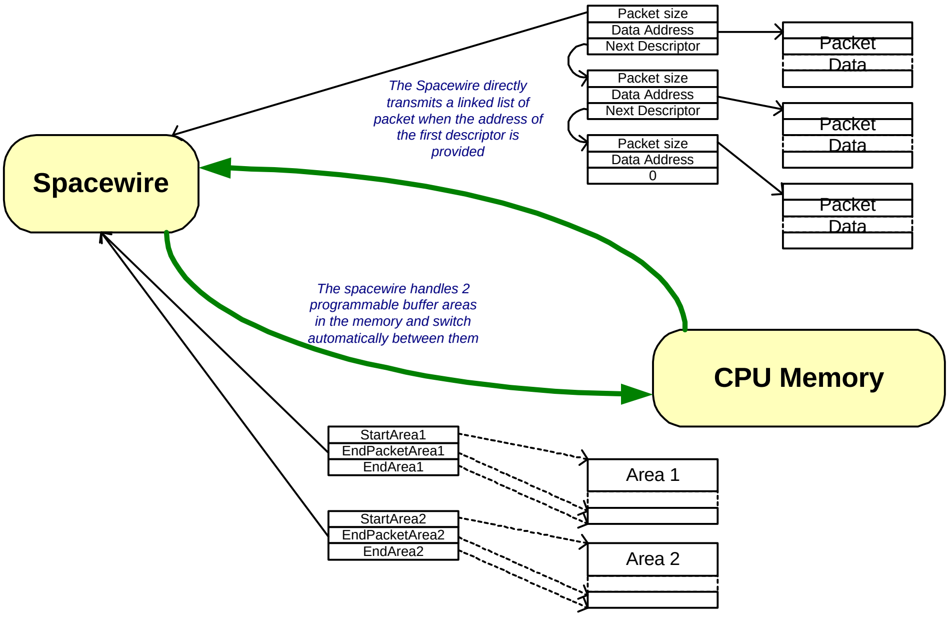


Figure 9 : Example of Spacewire intelligent DMA

The linked list of packet for transmission can be directly generated as a standard C-like linked list. The double area with flip-flop management in reception allows using the Spacewire at full performances

4 METHODOLOGY AND TOOLS

4.1 METHODOLOGY

The particularity of the development of a SoC comes principally from the use or re-use of blocks coming from different sources. Each block has its own particularities. The different phases of the development of an ASIC have been performed for the SCoC development. Limitations of the design flow as described in AD3 for the development of the SCoC are established and an adaptation of the methodology to the SoC design is proposed.

Initial Analysis

The functional specification of a SoC design cannot be a self-contained document as it is based on the use or reuse of existing IP cores. Initial analysis should follow the same concept of reuse by the document and information. That means that the information necessary for feasibility study (complexity …) are presented for each IP core in a re-usable form. Feasibility study is limited here to the analysis of the design complexity as the foundry selection was out of the scope of this activity.

Architectural Design

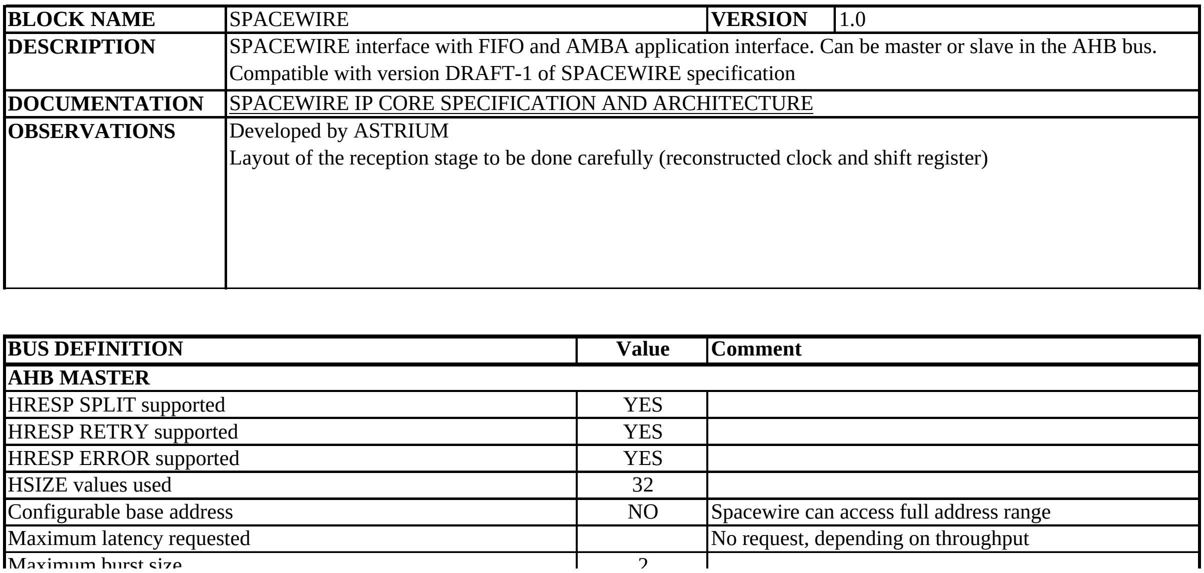
The architectural design phase integrates the architecture of the SCoC along with the detailed specification of the IP cores to be developed. The detailed architecture of each block is provided in block specific documents. Then the SCoC architectural design document references documentation of each block. For the simulations, the main point is the re-use (and eventually porting) of simulation environments from the IP core up to the system level.

Development of an IP core

The Spacewire IP Core has been developed to provide an AMBA Spacewire bridge suitable for use in SoC applications. It is also an example to study the development methodology of a re-usable IP core.

Gate Level Design

The gate level design has been performed targeting a commercial VIRTEX-E FPGA intended for function prototyping. Nevertheless, this step allows to reveal potential problems linked to synthesis of the system. As the main limitation was timing performance, only global synthesis has been performed, allowing the tool to perform global optimisations.



BLOCK NAME	SPACEWIRE	VERSION	1.0
DESCRIPTION	SPACEWIRE interface with FIFO and AMBA application interface. Can be master or slave in the AHB bus. Compatible with version DRAFT-1 of SPACEWIRE specification		
DOCUMENTATION	SPACEWIRE IP CORE SPECIFICATION AND ARCHITECTURE		
OBSERVATIONS	Developed by ASTRIUM Layout of the reception stage to be done carefully (reconstructed clock and shift register)		

BUS DEFINITION	Value	Comment
AHB MASTER		
HRESP SPLIT supported	YES	
HRESP RETRY supported	YES	
HRESP ERROR supported	YES	
HSIZE values used	32	
Configurable base address	NO	Spacewire can access full address range
Maximum latency requested		No request, depending on throughput
Maximum burst size	2	

Figure 10 : Extract of Spacewire information sheet

For each block, an EXCEL sheet provides all information necessary for the feasibility study : characteristics on the AMBA AHB and APB busses, number of external I/Os and gate count for a particular technology.

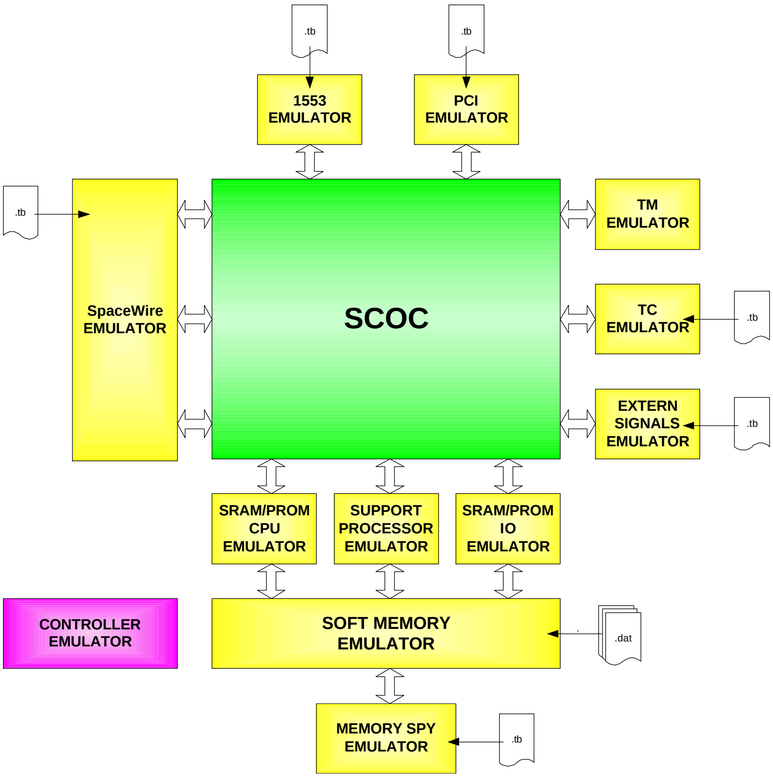


Figure 11 : Overview of the SCoC testbench structure

Most of the emulator composing the SCoC testbench are issued and re-used from the IP core testbench. The standardization of emulators allows inter-operability at upper level. The TM emulator is in fact an instantiation of functions provided with the PTME IP. The controller emulator allows synchronisation of the emulators altogether.

Hardware Evaluation

The evaluation/demonstration board BLADE has been used to test an implementation of the SCoC into a commercial FPGA – XILINX XCV2000E. Bugs discovered with this implementation, mainly located at asynchronous boundaries, have been rapidly corrected and re-validated. The evaluation also allowed measuring performances of the high speed I/Os implemented within the SCoC

4.2 TOOLS

The SCoC development has been using software tools for classical ASIC development. This section aims to discuss on each of these tools, mainly VHDL simulator and synthesis tool when used to develop SoC.

Simulation tool

A commonly used VHDL simulation tool used is Mentor Modelsim. This tool is well suited for verification of the SoC at RTL level. A standardization of the methodology used for the development of testbench as well as basic rules for the VHDL RTL description eases the simulation of complex SOC.

Synthesis tool

Synplify is a powerful synthesis tool for quick FPGA development. The TCL scripting possibilities ease the use of this tool for complex SoC designs.

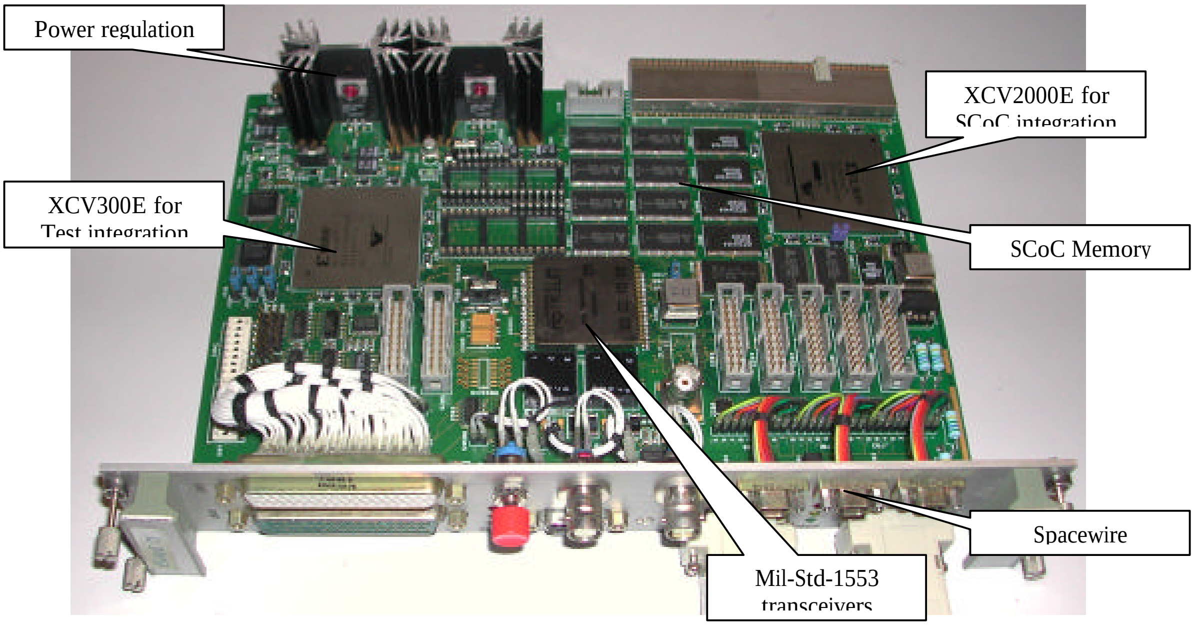

Power regulation
XCV2000E for SCoC integration
XCV300E for Test integration
SCoC Memory
Spacewire
Mil-Std-1553 transceivers

Figure 12 : BLADE board

The Compact PCI BLADE board integrates all environment necessary to evaluate the SCoC with a very limited test environment.

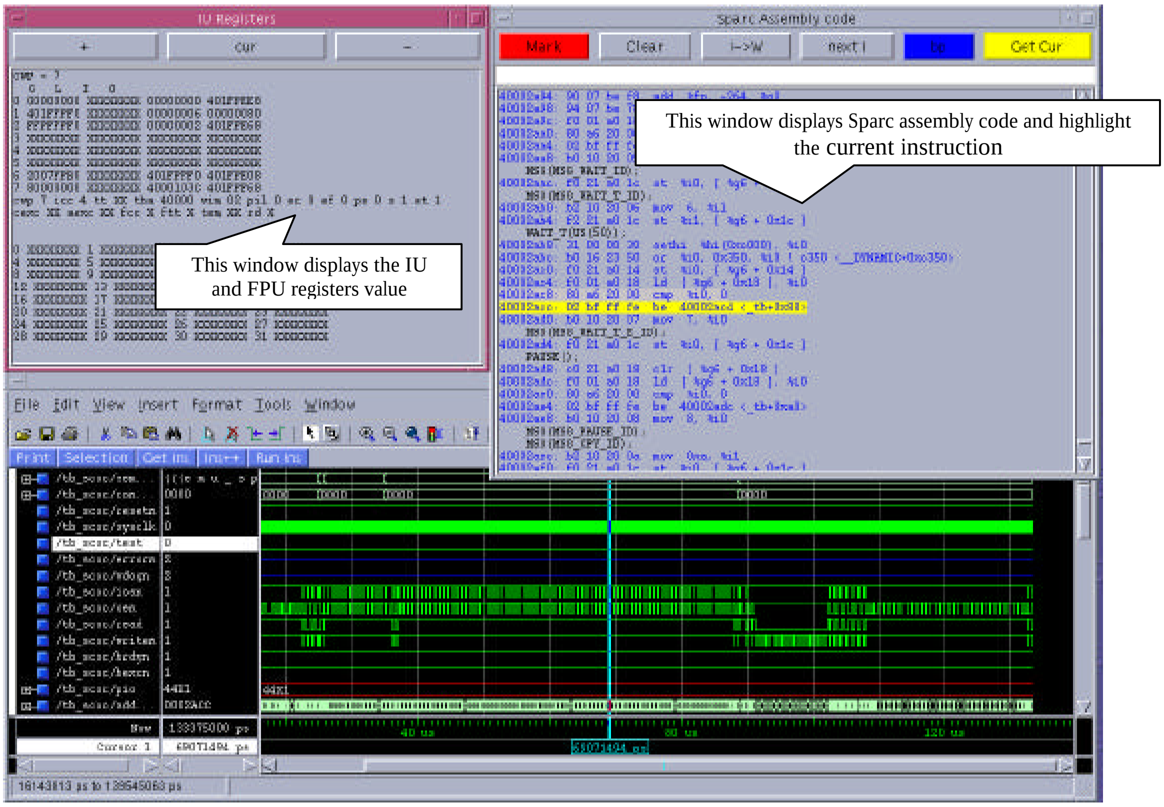

IU Registers
+
cur
-
Sparc Assembly code
Mark
Clear
i->W
next I
bp
Get Cur
This window displays Sparc assembly code and highlight the current instruction
This window displays the IU and FPU registers value
File Edit View Insert Format Tools Window
Print Selection Get ins Ins++ Run ins
40 us
80 us
120 us

Figure 13 : Overview of the Modelsim simulation

The Modelsim scripting capabilities allow highlighting the line of code currently executed at position of the cursor in the wave window, easing the debugging of hardware/software interactions.

5 EVOLUTION OF STUDY

Possible evolutions of the design, of the IP and of the method have been pointed out in the different sections the Final Report. They start from the acquisition of detailed system requirements for performances, to the evolution of the LEON structure, the study of different bus architecture and the development of corresponding VHDL resources, the evolution of the existing IP cores. Following the first evaluation of the performances of the SCoC on the BLADE board, it is also necessary to push forward the full verification of the system, to address the testability issue (including the LEON2 evolution) prior to reach a state were we can envisage to put the system in a real ASIC.

6 CONCLUSION

The study performed for this contract put in evidence the absolute necessity of a reliable and performant design methodology of SoC development. It allows speeding up the architectural development phase of a large design as shown for the development of the SCoC.

This methodology is mainly based on the availability for the European space community of a library of validated IP cores. This is now well supported by ESA, as indicated by the development of the LEON-FT CPU core, the PTME, the Spacewire IP, the HurriCAN controller, the AMBA to PCI wrapper and a collection of small but indispensable IP.

The methodology is also based on the use of standards coming from commercial applications, military developments or space industry, such as AMBA busses for internal connections of the IP cores, and also external interface standards, that allows the reuse of the developed IP core, such as PCI, Mil-Std-1553, Spacewire, CCSDS TM/TC. This strategy was beneficial for the SCoC development as it allowed reusing already developed validation tools (emulators for the simulations and standard hardware testbenches for the board evaluation).

The promotion for using these standards for ESA programmes is absolutely necessary to enhance the catalogue of available IP cores and the availability of verification/validation tools.

The methodology is also based on the availability of large commercial PLD that allow rapidly prototyping the system and functionally validating it deeper than with only simulations. FPGA prototyping also provides a breadboard platform for early software development, well before the availability of the System-On-ASIC.

The development of a demonstration/evaluation hardware platform does not suppress the essential simulation phase. Nevertheless, due to the complexity of the SoC, the simulation time dramatically increases (while partially compensated by the increase of the simulation platform computation performances). The simulations are then used to fully verify the IP core functionality in a standalone mode, while the system simulations only partially verify the core functionality of each IP, focusing on the correct integration of the SoC.

The main conclusions of the SCoC activity are then :

- The development of SoC including the CPU core such as the designed SCoC is crucial for the development of future small space platforms and probes
- There is a great interest of the SoC methodology for the future development of highly integrated ASIC, including or not the CPU core.
- The methodology of IP reuse shall be promoted by the Agency in order to complete the already available catalogue of IP cores, and to enhance the verification level of these IPs.
- The software tools already used for the ASIC development (for VHDL simulation and synthesis) are still suitable to the development of SoC, provided proper scripting is used.
- The hardware prototyping of the SoC using large commercial PLD greatly improves the system verification, compensating for the reduction of verification coverage of post layout simulations.

For the study continuation, potential user's of the system shall validate the requirement and functional specification of the developed SCoC. Then a full verification of the functionalities and performances of the current or enhanced design can be performed. Finally, manufacturing a prototype ASIC and its integration in a complete Data Handling System evaluation platform will validate all the concepts of these developments, from the interest in the product to the global methodology.

---%---%---%---%---