

---

## CLP PROGRAM

# CLP Executive Summary Report

---

Category: 4

Class: 2

Prepared by:

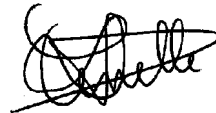
M. Ruiz



08/09/16

Approved by:


P. Lenelle



08/09/16

Authorised by:

M. Ruiz



08/09/16

Issued by : SABCA	Contractual : Yes (X) No ( )	Date : 01/09/2016
Customer : ESA	Contract Number: 4000107720/12/NL/LvH	

**TITLE :** CLP Executive Summary Report

**AUTHOR:** ruiz

**AUTHOR'S ABSTRACT:** This document presents the CLP Executive summary report of the Phase 1 of the CLP program

**TOTAL NUMBER OF PAGES:** 17 - **Number of Annexes :** n/a

**File Identification :** CLP-TN-G-011-SABCLAR1r0.docx

Distribution						Key Words
Internal			External			
Dept.	N° Cop.	Names	Dept./Company	N° Cop.	Names	
		Document's signatories	ESA/ESTEC	1	R. Weigand	CLP, ASIC, processor, real-time
		CLP team	ESA/ESTEC	1	L. Van Hilten	
		K. Vekemans				

## ISSUE / REVISION STATUS

Date	Iss./Rev. N°	Brief Description of Change	Mod. N°	ECP/PM N°	Config. Control Approval
01/09/2016	1/0	First Issue			

**TABLE OF CONTENTS**

<b><u>1</u></b>	<b><u>INTRODUCTION AND ABSTRACT .....</u></b>	<b><u>5</u></b>
<b><u>2</u></b>	<b><u>DOCUMENTS AND ACRONYMS .....</u></b>	<b><u>5</u></b>
<b>2.1</b>	<b>APPLICABLE DOCUMENTS .....</b>	<b>5</b>
<b>2.2</b>	<b>REFERENCES.....</b>	<b>5</b>
<b>2.3</b>	<b>ACRONYMS.....</b>	<b>6</b>
<b><u>3</u></b>	<b><u>OVERVIEW OF THE ACTIVITY.....</u></b>	<b><u>7</u></b>
<b><u>4</u></b>	<b><u>OUTCOME.....</u></b>	<b><u>8</u></b>
<b>4.1</b>	<b>CONTROL LOOP PROCESSOR .....</b>	<b>8</b>
<b>4.2</b>	<b>SOFTWARE DEVELOPMENT ENVIRONEMENT.....</b>	<b>10</b>
<b>4.3</b>	<b>VALIDATION CAMPAIGN.....</b>	<b>12</b>
<b>4.4</b>	<b>ASIC TECHNOLGY .....</b>	<b>13</b>
<b><u>5</u></b>	<b><u>RETURN OF EXPERIENCE .....</u></b>	<b><u>14</u></b>
<b>5.1</b>	<b>SUBCONSTRUCTORS.....</b>	<b>14</b>
<b>5.2</b>	<b>LESSONS LEARNED .....</b>	<b>14</b>
<b><u>6</u></b>	<b><u>DELIVERABLES.....</u></b>	<b><u>15</u></b>
<b>6.1</b>	<b>DOCUMENTS.....</b>	<b>15</b>
<b>6.2</b>	<b>HARDWARE .....</b>	<b>16</b>
<b>6.3</b>	<b>SOFTWARE.....</b>	<b>16</b>
<b><u>7</u></b>	<b><u>CONCLUSIONS.....</u></b>	<b><u>17</u></b>

## 1 INTRODUCTION AND ABSTRACT

Nowadays, the field of tight control loops, characterized by hard real-time constraints (loop frequency > 1 kHz) in conjunction with complex algorithmic needs, currently lacks a microprocessor allowing to make a software programmable approach economically and technically viable. The control of electro-mechanical actuators is an example of target application.

A dedicated space-hardened microprocessor, called Control Loop Processor (CLP), has been designed in this activity and integrates several key features ensuring a fully deterministic behaviour as well as an embedded robustness/anomaly management in conjunction with vectorial IEEE-754 floating-points operations. A wide range of interfaces have also been selected to cover current and future space-oriented interfaces.

A software development environment has been defined during the activity to deal with the need of future users. In particular, the development of a C compiler and a code generator taking Simulink as code source is envisaged. This activity allowed designing a first version of the assembler, the Simulink macro library and the ISS, based on a cycle-accurate SystemC model

## 2 DOCUMENTS AND ACRONYMS

### 2.1 APPLICABLE DOCUMENTS

- [AD1] 4000107720-13-NL-LvH GSTP Contract  
GSTP Contract – Control Loop Processor (Phase 1)
- [AD2] ECSS-Q-ST-60-02C  
Space Product Assurance, ASIC and FPGA development

### 2.2 REFERENCES

- [RD1] CLP-DVP-C-001-SABC  
CLP Development Plan
- [RD2] CLP-DJ-C-001-SABC  
CLP Feasibility and Risk analysis
- [RD3] CLP-TN-C-001-SABC  
CLP Requirements Specification
- [RD4] CLP-VP-C-001-SABC  
CLP Verification Plan
- [RD5] CLP-COP-C-001-SABC  
CLP Control Plan

### 2.3 ACRONYMS

A5ME	Ariane 5 Midlife Evolution
ADC	Analog to Digital Converter
AWG	Arbitrary Waveform Generator
BIST	Built In Self Test
CLP	Control Loop Processor
CPU	Central Processing Unit
DRD	Document Requirements Description
DSP	Digital Signal Processor
EDAC	Error Detection And Correction
EEPROM	Electrically Erasable Programmable Read Only Memory
EGPIO	Extended GPIO
FPGA	Field Programmable Gate Array
GPIO	General Purpose Inputs Outputs
GSTP	General Support Technology Program
HBRISC	Hardened Bi Reduced Instruction Set Computer
HW	HardWare
I2C	Inter Integrated Circuit
ISS	Instruction Set Simulator
ITAR	International Traffic in Arms Regulation
JTAG	Joint Test Action Group
KLOC	Kilo Lines Of Code
LSB	Least Significant Bit
LVDS	Low Voltage Differential Signalling
LVDT	Linear Variable Differential Transformer
MAC	Media Access Controller
MSB	Most Significant Bit
PWM	Pulse Width Modulation
RT	Remote Terminal
SEL	Signal Event Latch-up
SEU	Single Event Upset
SIMD	Single Instruction Multiple Data
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SW	SoftWare
TVC	Thrust Vector Control

### 3 OVERVIEW OF THE ACTIVITY

This activity has covered the phase 1 of the whole CLP development. A number of trade-offs and studies have been performed to analyze and propose:

- A processor architecture that is in line with the targeted needs, derived from the HBRISC2 device and the work made during HBRISC3 activities. The selection of the IPs has also been performed
- A list of SDE tools to support the full V-cycle SW development, based on well-known tools
- An analysis of the feasibility of the DARE180 ASIC technology with respect to the CLP characteristics, including analog functions

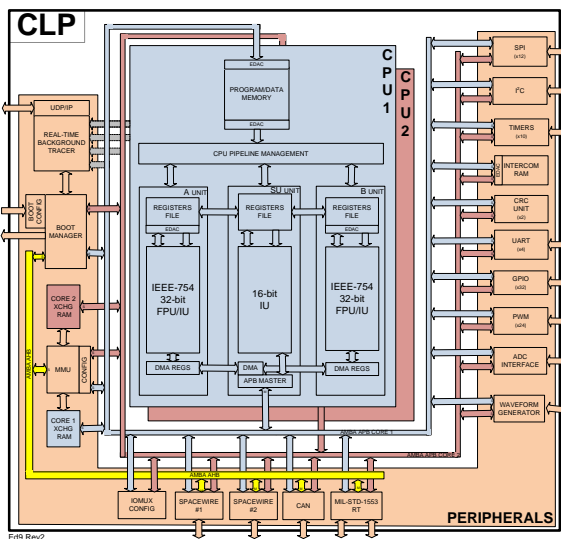
The activity has been closed with PDR acknowledging:

- The development of the CLP RTL code, based on its specification
- Development of a cycle accurate SystemC model allowing the co-validation with the VHDL code and a solid basis for the ISS
- A complete suite of automated testbenches allowing the validation of the CLP RTL code by simulation and via FPGA prototyping (based on Cobham Gaisler RASTA system).
- A preliminary version of the CLP assembler and unit test manager. A first version of the macro library has also been developed
- A preliminary business plan

## 4 OUTCOME

This section provides a detailed overview of the phase 1 main outputs constituted of the CLP, its SDE items and the ASIC feasibility.

### 4.1 CONTROL LOOP PROCESSOR



The Control Loop Processor is made of **two CPUs** (drawn in blue and read) and a number of **peripherals** (surrounded by an orange box).

Each CPU is based on a RISC architecture and cache-free topology to provide a fully deterministic behaviour. No interruptions is foreseen. It is made of three arithmetic units. Two of them are **floating-point units** and the last one is a 16-bit **integer unit**. The floating-point units integrate the IEEE-754 format in single precision. Various SIMD instructions are available to parallelize computations. The CLP target speed is 50 MHz thus providing up to 200 MFLOPs peak

performance when the 4 FPU's are fully occupied. Each CPU contains a 32kx39 program memory, called **PRAM**, which is also available for data through the APB bus controlled by the CPU. Each CPU integrated 2x512 **general-purpose registers** in floating-point units and 32 general-purpose registers in the SU unit. A **stack pointer** is also available with an on-chip integration of stacks (using general-purpose registers). Each CPU contains a **DMA**. This unit allows to off-load the software from transfers occurring between the CPU registers and any APB peripherals. The communication between the two CPUs is possible through an 1kx39-bit wide **INTERCOM RAM**.

Two on-chip busses, based on the APB AMBA standard, are available. Each CPU has its own bus to interact with peripherals and has the master role. Each peripheral is thus exclusively slave. An **allocation and write protection** mechanism is integrated to program, at boot time, which APB register or segment is allocated to which CPU. When the software is running, this configuration is static thus ensuring a deterministic communication between the CPUs and the peripherals. Any abnormal access is detected and reported in a dedicated counter.

10 on-chip **Timers** are available for use by the CPUs or most of the various on-chip peripherals. These timers constitute the nerve of the CLP as they allow to synchronise the software running of the CPUs with the used peripherals. Two **SPI** interfaces are included and can be programmed in either master or slave mode. The data acquisition can be either sequential or made in parallel. This interface is particularly interesting when working with serial off-chip ADCs. A **PWM** interface is integrated allowing to drive up to 2x24 signals. Each output has its complement thus easing the



generation of signals intended for power electronics. Two **AWG** are available to automatically generate any arbitrary waveform. The typical use of these AWG are either performing sensors excitation or some sort of analog reporting. A parallel **ADC interface** is included to allow interacting with any off-chip parallel interface. The polarity and timings are fully programmable. Up to 4 off-chip devices can be addressed and 16 samples transferred per acquisition (which is triggered by one of the CLP timers). The ADC interface is autonomous. Four **UARTs** are included to interface any off-chip device with the CLP. The baud rate is programmable and either 8-bit or 32-bit mode available. The CLP also integrates one **I2C** interface to interact with upcoming space devices which make use of this standard. Two **CRC** units are available to perform any data integrity check. The polynomial is fully programmable. 96 **GPIOs** are available either in the conventional mode (input, output or input/outputs) or in a MEM8 mode allowing to automatically interact with an external 8-bit memory. These GPIOs are multiplexed with other interfaces in the IOMUX pins.

A **Spacewire-RMAP**, **MIL-STD-1553 RT** and **CAN** interfaces are included to support most space applications needs. All these interfaces comply with their respective standards are interact with the software through two exchange RAMS. One CPU has its dedicated XCHG\_RAM which is 8Kx39 bits wide. An **MMU** performs the routing from/to these XCHG\_RAMs from/to the mentioned interfaces. The MMU can be programmed by the software through descriptors. An allocation mechanism is also foreseen to filter data handled by a given interface.

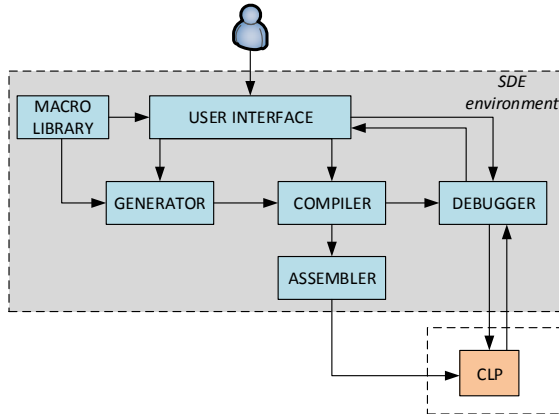
The software is debugged and validated through the **RTBT** link. This interface is based on the 100 Mbits/s Ethernet/UDP standard and allows to perform real-time tracing of CPUs registers, APBs traffic, MMU interactions and PRAMs activity. The RTBT is not intrusive and does not need instrumentation code thus making it suitable for applications forbidding useless code.

The CLP is **fault-tolerant** against any SEU occurring in its whole memory space. EDAC management is integrated to perform “on-the-fly” correction in case of read and an on-chip write-back correction to avoid errors accumulation. A scrubbing unit is also foreseen with a programmable refresh rate and correction address range. The fault tolerance also applies to any abnormal event occurring inside the CLP such as a deadlock condition in the ADC interface, an allocation violation detected by the MMU, a saturation condition occurring in an arithmetic unit or an illegal operand detection.

Finally, a **boot manager** is included which provides a robust boot and restart mechanism through 4 boot descriptors. 5 boot sources can be programmed (RMAPx2, CAN, MEM8 or RTBT) to automatically upload these boot descriptors and then boot the whole CLP according to the content of these ones.

In the scope of phase 1, the CLP VHDL code has been fully developed.

**4.2 SOFTWARE DEVELOPMENT ENVIRENEMENT**



The **SDE user interface** is based on two standard tools which are commonly used in space applications:

- Matlab/Simulink to graphically design the application with built-in entry methods and perform simulation
- Eclipse which is an open-source framework giving a user-friendly interface to all the SDE items that will be developed

The **SDE generator** is intended to produce an intermediate level model from the high-level model that has been designed by the front-end tool. The generated model is compliant with format defined by LLVM library and can thus be used as an entry by the SDE compiler. The SDE generator will make use of the off-the-shelf Qgen tool, provided by ADACORE company.

The **SDE compiler** is intended to produce a target independent low-level model from a model described with C standard. It can also read an intermediate-level model which is compliant with LLVM format. The generated code can be as an entry for the SDE assembler. The choice of the LLVM library is motivated by the various trade-offs that have been made during the phase 1 of the project. No license or fee is needed to access this library.

The **SDE assembler** is intended to produce a binary code from a low-level representation model. A first prototype has been designed for this tool and is available on request. It is intended to update and enhance the development with a number of additional utilities such as a configurator allowing a user to initialise the CLP peripherals and CPU(s) in a friendly fashion, a loader allowing the user to download the binary code into the CLP board, an assembly editor based on Eclipse.

The **SDE debugging tool** is intended to support the user to test, debug and validate an application running on the CLP. Four items are foreseen: the unit test manager giving a basic infrastructure to the user to develop unit test campaigns on CLP assembly code, the tracer allowing to program and track all reporting information to/from the RTBT unit inside the CLP, the ISS based on the cycle-accurate CLP SystemC model, the debugger allowing the user to interactively debug a given CLP code by means of various intrusive functions. The debugger will make use of the SystemC model whose behaviour is ensured to be equivalent to the CLP because of the all the co-validation and co-simulations made during phase 1.

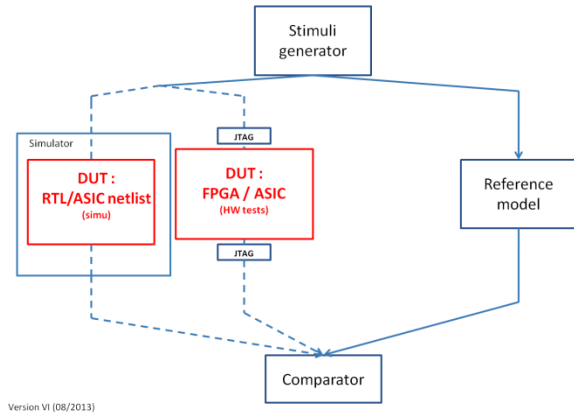
When a CLP application is designed by mean of the Matlab-Simulink tool, two approaches are possible. The user may use the standard library available within the environment (i.e. S-function, and

so on...) or a dedicated library made of custom blocks coupled to CLP assembler code or a mix of both approaches. To support the user in this second alternative, a **library** with a number of predefined **CLP macros** including commonly used high-level functions will be developed and cover, for instance, more complex mathematical functions such as filters and basic drivers functions allowing to simplify the communication with the CLP interfaces (PWM, MIL-STD-1553, etc ...).

In the scope of phase 1, the SDE assembler and the CLP macro libraries have been developed and are available in preliminary version. The de-risking activities ensuring the successful development of remaining SDE items (during phase 2A and phase 3) has also been performed.

**4.3 VALIDATION CAMPAIGN**

Eight different testbenches are or have been developed to extensively verify the CLP RTL code through simulations and FPGA prototyping. The validation methodology is given on figure and aims to generate a number of stimuli which are applied to a reference model and the DUT in an automated fashion. The generated data are then compared and a report is created to provide the pass/fail status and reproduce the problem if required



For most CLP functions, the reference model is a **cycle-accurate SystemC model**. It has been developed in parallel with the VHDL code. The SystemC model is foreseen to be integrated in the ISS as well as the various SDE tools such as the debugger of the unit test manager to take advantage of its full representativity.



Stimuli are created by a tool generating directed stimuli and /or pseudo-random vectors according to a predefined test scenario. The stimulus then feeds the DUT and its corresponding reference model. The DUT may either be the CLP RTL code or an FPGA prototype – based on the

RASTA system from Cobham Gaisler - integrating the full CLP design in a Virtex-4 device.

In the scope of phase 1, More than 600 hundreds test campaigns based on Perl and Python scripts have been developed to extensively cover the CLP design and ensure the maturity level requested by ECSS standard. Whereas a number of interfaces (SpaceWire, CAN,...) still need additional validation, the large majority of functions have been successfully validated thus improving the global TRL of the product.

This RASTA system has also been used to successfully make up to 3 “real-system” demonstration integrating a fault-tolerant CLP software driving a permanent-magnet synchronous motor with a vectorial control strategy.

For user evaluation, the SystemC model and the CLP FPGA prototype are available.

#### 4.4 ASIC TECHNOLOGY

The feasibility performed on the DARE180 ASIC technology has been based on a trial synthesis made on the full CLP VHDL design. The work led to the following results:

- Maximal frequency: 58 MHz in pre-layout conditions (hardened HIT scan cells have been selected)
- Die area: around 14 mm x14mm (routing, pads, JTAG and scan logic excluded), a large part of the design being made of RAM cells.

These figures led to consider other alternatives to the DARE180 since yield and consumption issues were considered as likely thus impacting the CLP potential. Note also that significant work has been made to assess the possibility of integrating analog functions (ADCs, DAC,...) but no relevant solution was found.

It is today foreseen to target the CLP design on the DARE65 ASIC technology which is currently under development by IMEC. Further analysis will have to be made in the future.

Note also that FPGA variants are available with the current definition of the CLP.

## 5 RETURN OF EXPERIENCE

### 5.1 SUBCONTRACTORS

The SDE has been led by **Spacebel** (Belgium). The implication and availability of the technical team has been appreciated. Their knowledge and involvement allowed preparing and de-risking the development activities that are part of the phase 2A.

The ASIC feasibility has been by **IMEC**. The contact was excellent and technical work has been appreciated. Their honesty has also being valued despite the fact the DARE180 has been considered as not optimal.

The selected IPs (SpaceWire/RMAP, MIL-STD-1553 RT, CAN, SPI, I2C and Ethernet) were provided by **Gaisler**. The helpdesk was very reactive and the documentation very good despite a few uncertainties on details. The IP integration inside the CLP design was made with no particular problem.

### 5.2 LESSONS LEARNED

The validation methodology proposed by SABCA was challenging and led to a product with a very good level of maturity. The co-validation with the SystemC was the good approach even though the fact that that man-effort needed to align the VHDL, the SystemC and the specification (up to the cycle level for some parts) should not be neglected in a future project.

The automated testbenches, in addition to their key role in the validation campaign, also allowed to monitor the maturity of the product thanks to the non-regression tests settled early in the development and executed every night. This point allowed to make early trapping of VHDL and SystemC regressions during bug resolution made during the day.

The use of pseudo-code to pre-design and document the extensive number of tests that were foreseen during this activity was a good approach. However, the use of either textual descriptions or standard languages (perl, python,...) is preferable.

The use of Eclipse and open-source tools (Git and Mantis) to perform design and bug management on the VHDL, SystemC, validation scripts and testbenches has clearly provided an added value to the overall project quality.

## 6 DELIVERABLES

The following deliverables have been produced within this contract:

### 6.1 DOCUMENTS

- CLP Feasibility and Risk Analysis
- CLP Requirements Specification
- CLP Delta Specification
- CLP Development Plan
- CLP Verification Plan
- CLP Verification Control Document
- CLP Control Plan
- SDE Requirements Baseline
- SDE Requirements Specification
- SDE Development Plan
- SDE Validation Plan
- SDE ECSS-E-ST-40C Statement of Compliance
- SDE ECSS-Q-ST-80C Statement of Compliance
- Business Plan + website
- CLP Validation Plan
- CLP SystemC Design Document
- SDE Preliminary User Manual
- Software Product Assurance Plan
- SDE - SW Metrics
- SW Library Preliminary User Manual
- SW Configuration Management Plan
- CLP Architectural Design
- CLP Datasheet
- CLP Validation Report
- Summary Report

## 6.2 *HARDWARE*

- CLP VHDL code (+IPs)
- Functional RASTA Board with fully tested CLP bitstream
- Automated testbenches (using COTS boards)

## 6.3 *SOFTWARE*

- SDE assembler
- CLP macro library
- SystemC model
- Eclipse GUI interface
- SDE unit test manager



## 7 CONCLUSIONS

This GSTP activity, called phase 1 in the overall CLP development, led to fully design and validate the CLP based on its specification, to develop a cycle-accurate SystemC model, to make a first version of the CLP assembler and to define the SDE tools that need to be developed during phase 2A and phase 3. The activity has been successfully performed thus now allowing the space industry to benefit from mature HW and SW material for early evaluation.

The phase 2A will conduct the development of the SDE tools while phase 2B will consolidate the CLP validation activities and lead to an ASIC prototype on a technology which still needs to be identified.