



European Space Research  
and Technology Centre  
Keplerlaan 1  
2201 AZ Noordwijk  
The Netherlands  
Tel. (31) 71 5656565  
Fax (31) 71 5656040  
[www.esa.int](http://www.esa.int)

# DOCUMENT

## ESA IP Core Technical Requirements

ESA IP Core technical guidelines.docx

**Prepared by** Kostas Marinis, Luca Fossati  
**Reference** TEC-EDM/2010.61/KM  
**Issue** 2  
**Revision** 6  
**Date of Issue** 24/04/2015  
**Status**  
**Document Type** TN  
**Distribution**

European Space Agency  
Agence spatiale européenne



# APPROVAL

<b>Title</b> ESA IP Core Technical Requirements	
<b>Issue</b> 2	<b>Revision</b> 6
<b>Author</b> Kostas Marinis, Luca Fossati	<b>Date</b> 24/04/2015
<b>Approved by</b>	<b>Date</b>

# CHANGE LOG

Reason for change	Issue	Revision	Date



**Table of contents:**

**1 Introduction ..... 4**

**2 Ip Core Technical Requirements ..... 4**

2.1 Generic IP Requirements.....4

2.2 VHDL IP Core Requirements .....5

2.3 SystemC IP Core Requirements .....8

**3 List of deliverables..... 8**

3.1 HDL .....8

3.2 SystemC .....9

**4 Target technologies..... 10**

**5 Reference and Applicable documents..... 10**

## 1 INTRODUCTION

The objective of this document is to describe the technical requirements and the minimum set of deliverables expected, in order to allow a design to be reused as an “Intellectual Property Core”, or IP Core. Furthermore, the main subject of this document are the “soft IP cores”, i.e. IP Cores that are delivered to the Agency as synthesizable RTL HDL code or as SystemC Simulatable code.

## 2 IP CORE TECHNICAL REQUIREMENTS

An IP Core (either HDL or SystemC) should generally be portable to various designs and technologies, and be easily re-usable by third parties, who were not involved into the development of the IP. It must therefore satisfy a number of requirements:

### 2.1 Generic IP Requirements

- R1 **Self-containment:** The IP database (source code, test benches, scripts) must be self-contained. External dependencies (i.e. on elements, design units, packages, etc. which are not part of the same IP Core group of files and/or for which the user does not have the same IP rights) shall be avoided, unless these can be expected to be commonly and freely available to users, like e.g. IEEE packages generally supported by CAD tools, or technology specific simulation models (memory models etc.). Background IPR shall be considered an external dependency in case they are not delivered with the same IP rights of the current development.  
All other files and documents which are necessary to make successful use the IP Core shall be identified in the IP Core documentation, detailing the versions to be used and the way to obtain them.
- R2 **CAD tool independence:** The IP-Core code database shall not depend on any specific CAD tool, and it is required that the compilation order and any library dependency be specified in a tool-independent way. Tool dependent constructs inside the code are only allowed if there is no other way to describe the same functionality, and they require special attention in code comments, documentation and reviews.
- R3 **Testbenches:** Testbenches shall be implemented as pass/fail tests, with automatic checking of the results and allowing batch and regression runs.  
Unlike what specified in [2], testbenches might be written either using the VHDL or SystemC language; appropriate scripts have anyway to be provided to allow using the same testbenches (or a subset of those) with both VHDL and SystemC models.
- R4 **Version Management:** All the code and documents shall be available during the development on an internet-accessible concurrent version management system (such as Subversion or GIT).
- R5 **IP Verification and Validation Plan:** The IP Verification and Validation Plan document shall describe the complete verification environment and procedures in detail. All test cases shall be described with clear indications of the IP configurations on which they have been validated. The correspondence between the testbench code and the test cases described in this document shall be stated. A list of tools and operating systems (and their versions) used during the development/verification of the IP shall also be provided.



The IP Verification and Validation plan shall be based on the outline given in Annex E and F of [4]; in addition a *Requirement Coverage Matrix* must be provided, specifying the correspondence between the requirements and the tests which verify them.

Objective of the tests is to exercise the whole of the IP Core code, checking individually the functionality of its single components and to validate that the implemented functionality actually matches the intended one and the requirements.

- R6 IP Verification and Validation Report:** The verification results shall be documented in a Verification Report, along with explanations for problems, and justifications for non-passed cases. Technology and environment related verification results shall be summarised, and any problems which can be anticipated (e.g. memory initialisation during gate level simulation) shall be highlighted. Overall the tests shall cover 100% of the IP Core source code; in case that would not be possible, careful manual code analysis shall be performed and clearly described in the Verification Report. For VHDL code, coverage shall be measured on, at least, *statement*, *branch*, *FSM*, and *condition*.

## 2.2 VHDL IP Core Requirements

- R7 Coding Guidelines:** The ESA VHDL Modelling Guidelines [2] are applicable for any new VHDL design as a baseline. Any exceptions to the design and coding guidelines outlined in [2] must be noted in the Detailed Design Document. It is especially important to explain any asynchronous circuits, combinational inputs, and combinational outputs. With respect to what indicated in [2], the VHDL code of the IP Core shall have to adhere to the *IEEE Standard for VHDL Register Transfer Level (RTL) Synthesis*, [5]. Documentation shall be written to be readable by an automatic documentation system of choice (e.g. Doxygen, [9]).
- R8 Clock and Reset:** Clock and Reset trees shall contain no combinatorial logic; if really necessary, such logic shall be grouped in a well-defined module. All Flip-Flop shall be designed to have both synchronous and asynchronous reset signals. At IP-Core instantiation the user shall select (for example through *generics*) which of the two reset flavours to use; the selection shall be the same for the whole IP-Core (i.e. it shall not be possible in the same instantiation to mix synchronous and asynchronous reset signals).
- R9 IO pin requirements:** By default, there shall be no bidirectional pins (for external I/O's, the IP must have separate input, output and enable pins). All outputs shall be registered. Deviations from these defaults must be avoided and, if they are really necessary, they must be documented in the IP Datasheet (see R14 below).
- R10 Functional verification:** FPGA prototyping based on a COTS evaluation board is encouraged in order to complement and accelerate functional verification. However, a sufficient set of simulation test benches is always required, allowing the user to carry out an assessment of the essential functionality in simulation. These test benches need to be available in full source code, including auxiliary scripts, pattern files and reference models (e.g. in C, SystemC, or Matlab) used for test pattern generation or analysis. Console output during simulation shall be verbose and shall give clear reference to the test cases defined in the IP Verification Plan (see R5 above). Test benches shall be self-checking pass-fail wherever possible; when not possible, reference ("golden") patterns are to be provided.



- R11 Timing Verification:** Post-synthesis simulations with timing backannotation (SDF) for worst case (for setup checks) and best case (for hold checks) shall be run for FPGA and ASIC technologies; in addition, post-Place and Route simulations shall be run for FPGA technologies. The list of such ASIC and FPGA technologies on which this requirement is applied shall be decided together with the Agency.
- R12 Verification of configurable IPs:** In many cases, the IP Core will have configuration options (hard-coded or software programmable), which make an exhaustive verification of all combinations of these options impossible. A reasonably chosen subset of configurations has to be verified. To avoid leaving the user in doubt, it is crucial that the chosen subset is fully documented and justified in the IP Verification and Validation Plan document (see R5 above).
- R13 Technology mapping:** In general, the IP shall be developed as technology independent, synthesisable VHDL code. However, for certain functionalities (e.g. embedded memories) technology specific macrocells may need to be used. The choice of target technologies is then subject to a mutual agreement between the contractor and the Agency. These macrocells need to be clearly identified in the source code (via proper encapsulation) and the documentation, in order to ensure easy portability to other technologies. Furthermore, a 'generic' version of the IP should also be provided, where the macrocells are replaced by behavioural simulation models. The IP Database description (R17) shall feature an easy way of configuring the target technology to be used, possibly integrated in the build-system.
- R14 IP Datasheet:** The IP Datasheet has a similar function for the SoC designer as the component datasheet has for the board designer: it should provide sufficient information, in sufficient detail, to allow a potential user to evaluate whether the IP Core is appropriate for the target application. It must also provide all the necessary information to allow the designer to integrate the IP Core into the target system. The IP Datasheet shall be distributable to the public.
- The IP Datasheet should follow the outline depicted in [3], Appendix C.5, with the following amendments:
- The hard configuration options (constants to be defined prior to synthesis), which do not exist at component level, have to be documented in a dedicated section, ensuring clear separation between hard (at SoC design time) and soft (programmable registers or input pins) configuration.
  - Any design techniques, implemented in the code, optimising certain electrical and environmental properties (e.g. fault tolerance by design, power saving modes etc.) shall be described along with the functional description.
  - The pin list shall specify the timing relation for every pin, such as 'synchronous with clock X', 'registered output', 'combinational through', etc.
  - The pin list shall indicate (in a typical application), which of the pins are for on-chip signals, and which are for external pads. In the latter case, the pad type shall be indicated (e.g. Schmitt trigger, open-drain, LVDS, pull-up).
  - Logical interfaces (e.g. groups of pins representing an address, or the interface to a data bus) shall be clearly identified, and all the relevant characteristics specified (e.g. endianness, etc.).
  - Mechanical, electrical (DC, AC, power) and environmental (thermal, radiation) specs are not applicable for a technology independent IP. However, the datasheet shall contain an overview of the technology mapping results obtained during the IP development.



- An indication of the area of the design shall be provided, as a number of basic (NAND) gates.
- Timing specifications should include input setup and hold times for all input and I/O pins and clock-to-output delays for all output pins. Timing specifications for any combinational inputs/outputs must be clearly documented. Timing for soft IP Cores should be specified for a representative process.
- ASIC layout and/or FPGA place and route guidelines should be provided for any timing critical signals.
- Recommendations regarding the clocking and reset strategies needed to drive the IP Core should be provided.

Annex G of [4] shall also be taken into account during the preparation of the datasheet.

- R15 IP User Manual:** The VHDL model hierarchy, and the corresponding file and directory structure, shall be described. Furthermore, the correspondence between the VHDL model structure and the design architecture described in the datasheet shall also be stated. In the ideal case, there should be a straightforward correspondence between the block names described in the documentation, and the directory/library structure and file names in the design database.

Instructions on how to setup the design database shall be provided, and any specific paths or variable settings required shall be documented. Furthermore, instructions shall be given on how to initiate “design operations”, e.g. compilation, simulation, synthesis, and timing analysis. An elegant solution is to provide a Makefile which, when called without argument, simply prints a list of all available targets starting up the various test benches, verification steps, etc.

Quickly explanation of the testbenches and detailed instructions on the steps necessary to run them shall also be given.

The IP User Manual shall be distributable to the public.

- R16 Single Event Effects Mitigation:** The design shall contain SEE hardening/mitigation measures implemented at the Architectural Level. Such hardening/mitigation measures shall be implemented with configuration options ensuring that a non-hardened design can be created for implementation of radiation hardened technology or for prototyping purposes.

In particular, Error Detection and Correction (EDAC) and/or parity bits for internal and externally used memory banks shall be included in the source code as options that can be activated or inhibited, depending on the user's needs

In addition, all Finite State Machines in the IP shall be dead-lock free and be able to go back to idle state in the event of entering an illegal state or experimenting an illegal state transition caused by an SEU.

- R17 IP Database description:** The IP database description will be shipped to the customer along with the database and shall effectively enable him to use the database and make the link between code and documentation. In the ideal case, the database structure (directories, filenames, signals names) is sufficiently self-explanatory, so the database description is reduced to a minimum and becomes a simple README text file in the root directory of the database. As a minimum, the following information shall be available to the user:

- Setup of database, any specific path or variable settings required





- How to start “design operations” (simulation, synthesis etc.) – an elegant solution is to provide a Makefile which, when executed without argument, simply prints a list of all available targets starting up the various test benches, verification steps etc.
- Link between VHDL model structure and the architecture described in the datasheet. In ideal case, there is an “easy-to-guess” correspondence between the hierarchy and block names described in the documentation and the directory/library structure and file names in the database.
- Link between test bench code (which test bench/generic parameter) and the test cases described in the verification document. List of tools and operating systems (and their versions) used during development/verification of the IP.

## 2.3 SystemC IP Core Requirements

- R18 Functional Behaviour:** The VHDL RTL version of the IP Core and the SystemC one shall have identical functional behaviour, i.e. given the same inputs, they shall produce the same outputs. This shall include the effects of configuration registers and any corner case if present. Functional verification shall be performed using the developed test-benches as specified at requirement R3 and or any reference model of the functionality.
- R19 Timing Behaviour:** The difference in timing at the interfaces between the VHDL RTL version of the IP Core and the SystemC one should not exceed 20% on any benchmark.
- R20 SystemC IP Core User’s Manual:** The User’s Manual has a similar function for the SoC designer as the component datasheet has for the board designer, as it should allow him to design and integrate the IP/component into the system. It should clearly indicate the SystemC IP Core configuration options which have to be documented in a dedicated section.
- R21 Build System:** The SystemC IP Core shall come with compilation scripts and shall use a scalable, configurable, multi-platform build system.  
*The use of autotools is discouraged due to their difficult use in multi-platform environments.*
- R22 Code Quality:** All the code shall be written according to ESA BSSC(2000)1 standard ([7]). The code elements (classes, attributes, methods, etc.) shall be thoroughly documented with a style compatible with the Doxygen (see [9]) documentation system; in addition documentation shall be added inside the body of methods, functions and procedure in order to ease the understanding of the behaviour of the code.

## 3 LIST OF DELIVERABLES

The following table lists the minimum set of deliverables to be expected as part of the IP core package (design database). The “*Reference*” column indicates the corresponding requirements. The description is split whether it is a SystemC or an HDL core.

### 3.1 HDL

Description	Reference
-------------	-----------



<ul style="list-style-type: none"> <li>- Complete, synthesizable source code for the design, and all its sub-blocks (in VHDL or Verilog).</li> <li>- VHDL/Verilog design examples that instantiate the IP core.</li> <li>- Simulation scripts, compilation scripts and any command files used during the simulations.</li> </ul>	R7, R1, R2, R9, R16
<ul style="list-style-type: none"> <li>- Testbenches, incorporating automated results checking, in the form of PASS/FAIL tests.</li> <li>- Bus functional models/monitors used in the testbenches (if any).</li> <li>- Simulation logs, code coverage reports.</li> </ul>	R10, R12
<ul style="list-style-type: none"> <li>- Synthesis scripts, constraints files, and reports for multiple technologies (at least two FPGA – e.g. Xilinx Virtex/Spartan, and Actel RTAX/ProASIC) and one ASIC technologies (e.g. Atmel 0.18um). Refer to Section 4 for further details.</li> </ul>	Section 3.2, R13
<ul style="list-style-type: none"> <li>- Complete documentation for the design, including: <ul style="list-style-type: none"> <li>o Datasheet</li> <li>o Detailed Design Document</li> <li>o Verification and Validation Plan</li> <li>o Verification and Validation Report</li> <li>o User Manual</li> </ul> </li> </ul> <p>It must be possible to distribute the Datasheet and the User Manual to the public, without the need to sign any license/NDA. The documents must come at least in an editable format.</p>	R14, R15, R5, R6, R17

### 3.2 SystemC

Description	Reference
<ul style="list-style-type: none"> <li>- Complete documentation for the design, including: <ul style="list-style-type: none"> <li>o Detailed Design Document</li> <li>o Verification and Validation Plan</li> <li>o Verification and Validation Report</li> <li>o User Manual</li> </ul> </li> </ul> <p>It must be possible to distribute the the User Manual to the public, without the need to sign any license/NDA. The documents must come at least in an editable format.</p>	R5, R6, R20
<ul style="list-style-type: none"> <li>- Complete, source code for the design.</li> <li>- SystemC design examples that instantiate the IP core.</li> <li>- Simulation scripts, compilation scripts and any command files used during the simulations.</li> </ul>	R21, R22
<ul style="list-style-type: none"> <li>- Testbenches, incorporating automated results checking, in the form of PASS/FAIL tests.</li> <li>- Simulation logs, code coverage reports.</li> </ul>	R3

## 4 TARGET TECHNOLOGIES

The IP-core shall be compatible with the following set of technologies. As a minimum, the implementation flow (synthesis for ASIC and FPGA, layout for FPGA) shall be exercised for one ASIC technology, one reprogrammable FPGA and one one-time programmable FPGA. As an indication, the following are some of the most commonly used technologies for space-related ASIC and FPGA developments:

- ASIC technologies
  - i. Atmel ATC18RHA (0.18 um)
  - ii. DARE rad-hard library on (UMC 0.18 um)
- FPGA
  - i. Actel RTAX (one-time programmable)
  - ii. Actel RTSX (one-time programmable)
  - iii. ATF280K (rad-hard, reprogrammable SRAM based)
  - iv. Xilinx Virtex-4/5 (SRAM based reprogrammable)
  - v. Actel Pro-ASIC flash-based (non-volatile reprogrammable)

The final choice regarding the target technologies shall be discussed and agreed upon between the contractor and the Agency.

## 5 REFERENCE AND APPLICABLE DOCUMENTS

- [1] Keating, M.; Bricaud, P., “Reuse Methodology Manual for System-on-a-Chip Designs”, 3rd edition, Kluwer Academic Publishers, 2002.
- [2] “VHDL Modeling Guidelines”, issue 1, September 1994, ESA/ESTEC,  
[https://amstel.estec.esa.int/tecedm/website/docs\\_generic/ModelGuide.pdf](https://amstel.estec.esa.int/tecedm/website/docs_generic/ModelGuide.pdf)
- [3] “ASIC Design and Manufacturing Requirements”, ESA document WDN/PS/700:  
<http://microelectronics.esa.int/asic/DesignReq.pdf>
- [4] ECSS-Q-ST-60-02C – “ASIC and FPGA development”:  
<http://www.ecss.nl>
- [5] IEEE Standard for VHDL Register Transfer Level (RTL) Synthesis (IEEE Std 1076.6™-2004 or later).
- [6] 1666-2005 - IEEE Standard SystemC® Language Reference Manual:  
<http://standards.ieee.org/findstds/standard/1666-2005.html>  
<http://www.systemc.org/downloads/standards/>



- [7] ESA C and C++ Coding Standard BSSC(2000)1:  
[ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/bssc2000\(1\)i10.PDF](ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/bssc2000(1)i10.PDF)
- [8] Lessons Learned from FPGA Development Version 0.2, September 2002:  
[http://microelectronics.esa.int/asic/fpga\\_001\\_01-0-2.pdf](http://microelectronics.esa.int/asic/fpga_001_01-0-2.pdf)
- [9] Doxygen:  
<http://www.doxygen.org/>