The GENCOD project : Automated generation of Hardware code for safety critical applications on FPGA targets

Pascal Pampagnin, Ludovic Letellier

The GENCOD project : Automated generation of Hardware code for safety critical applications on FPGA targets

Pascal Pampagnin[1], Ludovic Letellier[2],

1: Airbus Operations SAS, 316 route de Bayonne, 31060 Toulouse, France
2: Alyotech Innovations, 22, bd Déodat de Séverac, 31770 Colomiers, France

Abstract: GENCOD is a research project for solutions to automated generation of safe code for Field Programmable Gate Arrays (FPGA) targets.
The paper will describe typical ASIC/FPGA workflow, and current implementation for airborne electronic hardware design.
Major stakes in certification for airborne electronic hardware will be discussed.
The next part will detail the project, the proposed workflow and the associated tools.
We will present the current experimentations.

Finally, the conclusion will expose advantages and drawbacks of such approach.

Keywords: SCADE, VHDL, automated generation, avionics certification, tool qualification.

1. Introduction

Today Avionics manufacturers shall follow the rules given by non-airborne markets (telecom, personal computer, multimedia, home electronics). These strong leaders are driving the whole electronics domains, including components procurement, computer aided design tools usage, and methodology implementation.

These leading markets have very short life cycle, compared to airborne (for instance, life cycle for a memory is around 18 months, but life cycle for aircraft is 40-50 years).

Another trend is the increasing of complexity and integration, which follows Moore's law.

The following table shows the increasing complexity of microprocessors.

year	'80	'90	'00	'05	'10
µP	286	Intel 486 ™	Intel ® Pentium ® 4	Intel ® Itanium ® 2	Intel ® Itanium ® i7
Transistors	10^4	10^6	42.10^6	41.10^7	2.10^9
A/C	A300 A310	A330 A340	A340 500 /600	A380	A350
Digital units	77	115	200	300	>100
On board SW (Mb.)	4	20	40	80	>150
CPU freq. (MHz)	4	16	32	66	>166
(*) (gates / chip)	10	1k	32k	600k	>1M
Errors found per 100 kbytes	A few 100	Less than 10	N/A	N/A	N/A

Table 1 : Moore's law applied to avionics

(*) Flight computer Integration

Taking into account the technology changes, Avionics designers have also to cope with the main following trends :

- new and novel technology issues,
- merging formerly separate and independent functions on same hardware,
- multifunction components, displaying critical and non-critical functional paths in same systems/components,
- replacing mechanical with electronic parts (example relays and switches),
- Using complex electronic hardware in roles "traditionally" targeted at software.

2. What is Certification ?

To operate for the purpose of commercial air transportation, airplanes need a certificate delivered by an Authority that acknowledges that it meets all the applicable airworthiness requirements.

The certification is the legal recognition by the Certification Authority that the aircraft and its systems and equipments comply with the requirements. In particular, certification involves the assessment process of the design to ensure that it complies with a set of standards applicable to that type of product so as to demonstrate an acceptable level of safety.

Certification for civil aircraft is a process shared with the Aircraft manufacturer, the System designer, the LRU (or equipment), including hardware and software) Supplier (or applicant) and the Certification Authority. Cross certification is possible between EASA [1] and FAA [2].

Digital Devices (IP intellectual properties, Integrated circuits, ASIC and PLD components) are significantly used in electronic equipment, due to the increasing computation and integration needs. As these devices become more and more complex, aircraft functions may be increasingly vulnerable to the adverse effect of hardware design errors.

3. What is DO254 ?

DO254-ED80 [3] is the result from joined RTCA [4] & EUROCAE [5] Special Committee SC180. This Committee has been created in 1993 to address design assurance guidance for electronics hardware used in airborne systems.

The purpose of RTCA-DO-254 (referred to herein as “DO-254”) is to provide guidance for the development of airborne electronic hardware. US (FAA), and European (EASA) aviation safety authorities require this standard to ensure that complex electronic hardware used in aircraft systems works as specified under all foreseeable conditions, avoiding faulty operation and potential air disasters.
DO-254 compliance is becoming increasingly common on commercial and military aviation projects. Companies often struggle with the requirements and costs of DO-254 compliance.

The FAA began enforcing DO-254 in 2005, through Advisory Circular AC 20-152 [6].
DO-254 defines a set of objectives that airborne applicants and integrators must meet for their hardware to be certified for use in airborne systems

DO-254 defines both a design process life cycle and supporting processes that must be followed throughout the design development process.

Note that software is also required to apply certification requirements, such as DO178B [7].

Because of the nature and complexity of systems containing digital logic, adherence to a structured approach may be used to show compliance to certification objectives.

There are 5 system Design Assurance Levels, or DAL, level A through Level E, respectively corresponding to the 5 classes of failure conditions : catastrophic, hazardous/severe-major, major, minor, and no (safety) effect.
High safety critical system, such as flight control, cockpit displays, are “DAL A classified”.

4. Hardware design language representations

Major Aircraft manufacturers, systems designers and avionics designers use Hardware Description Languages, or HDL (such as Verilog, or VHDL) to describe complex electronics embedded in ASIC or PLD. These devices become key elements in recent aircraft certification programs. Today this HDL code is manually generated from textual specification. So this classical development process may introduce design errors, and it is difficult to verify. The introduction of automated or semi automated techniques will reduce the time to design and the time to verification. The interest of the project is also to analyse carefully the tool qualification stakes, in avionics domain.
Only few tools are existing in this domain (hardware language HDL generator) and are not used in avionics domain : the HDL code is still handwritten.
Few tool vendors are aware of qualification stakes in avionics domain.

5. Typical PLD flow for airborne applications

The state of the art within the avionic domain is to express the HW requirements with text using (Word, Framemaker,..) or dedicated tools such as (Doors or Requisite Pro) and "translate" manually them writing the VHDL code. The GENCOD project is willing to provide tools to work progressively at an higher level of abstraction using also automatic HDL code generation. All this approach has to be compliant with DO254 and the related safety issue.

The following figure gives an overview of PLD flow for airborne applications.

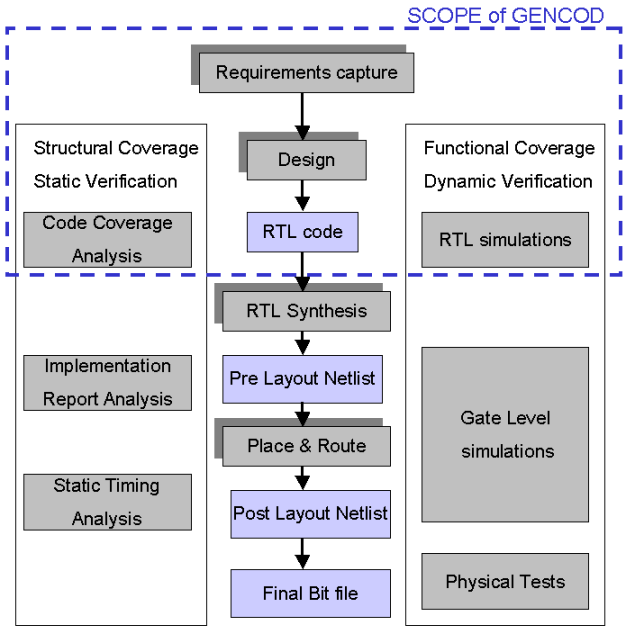


Figure 1 : PLD design flow

Note that requirements capture and conceptual design are done manually, with the help of some tools.

After the design step, we obtain a HDL RTL code.

This figure shows also the number of verification activities for the design, based on RTL level.

6. Major stakes in airborne certification

Both US and European Airworthiness Authorities focus mainly on complex electronic hardware (PLD, FPGA, ASIC) and complex COTS, and consider that classical industrial answers (definition and follow-up of an Electronic Component Management Plan, follow-up of the COTS Manufacturer, capture of non airborne service experience) are not enough as means of compliance.

We find now on recent or work-in-progress aircraft programs the following requirements, for instance :

- Independence for validation processes at safety critical levels (DAL A and B),
- Conformance to code standards for HDL,
- Requirements based testing to cover normal and abnormal operating conditions (robustness defects detection : clock frequency variations, power supply levels voltage variations, temperature variations,...),
- Definition of a target level of verification coverage of design requirements at critical levels (DAL A and B), the target based on the device internal structure is an acceptable means of compliance,
- Justification of non verification of specific detailed design requirements,
- Traceability between the specification requirements, the conceptual design, the detailed design (i.e. HDL), and the implementation, and between the requirements and the corresponding verification or validation activities,
- Justification to the authority of the claim for credit of relevant tool history, and assurance compliance for development and verification tools,
- Augmentation of life cycle data for Intellectual Properties used in ASICs or PLDs design.

Another difficulty for the Aerospace Industry is the time to market constraints.

Due to the increase of the complexity, complex PLD or ASIC designs become more and more difficult to be handled by individuals, especially for verification activities.

7. The GENCOD project

The GENCOD project is a 2-years (started in september 2008) multi partners research project, involving European Aerospace companies (Airbus, Dassault Aviation, SAFRAN, Thales Communications), ESL and software tools vendors (Esterel Technologies [8], Geensoft [9]), and a university laboratory (LRI). Note that Aeroconseil is involved as a subcontractor to perform independent assessment of compliance with DO254 aspects.

This project has been labelled in the French competitiveness cluster, System@tic [10].

The aim of the project is to study an HW design environment able to fulfil the strong contraints linked to the safety of the critical embedded system. The project is mainly focused on HDL code generation and its "certificability" for FPGA target based upon high level requirements but also the verification and validation activity. The application domain is mainly avionics but the approach could be also adopted for automotive, industrial safety and railway. Indeed, the avionics standard is often use as a reference and best pratices in the other domains.

The developed tools in the GENCOD project could be certified or qualified (tools defined as verification tools), in order to increase productivity, manage growing complexity, and support the certification process in FPGA development.

The output toolset is expected to provide the following features:

- Formal specification of textual requirements using high level modelling language (SCADE)
- Model simulation and formal verification
- Automated RTL code generation (VHDL) from SCADE models
- Traceability between textual requirements, formal specifications (model), and RTL code
- Automated documentation generation to help tool qualification and product certification in avionics domain.

Tools are developed by SME companies based on requirements provided by industrial partners which afterwards validate the tools on use case. Certification impact is also investigated.

Another goal of the project is to define an associated design flow, to ensure proper use of the tools in a DO254 certification context.

The initial target market is the aerospace FPGA design domain, with control-intensive features, like bus or interface controllers (ARINC, USB, MIL-STD-1553). However, the scope of GENCOD project might be extended to other safety-critical markets (automotive for instance), as the initial goals are reached.

8. The GENCOD project design flow

In order to improve productivity and give an acceptable answer to airworthiness issues, GENCOD project partners decided to choose a SCADE Suite from Esterel Technologies as the main tool in the workflow. SCADE suite is indeed a well known tool by system and software designers in aerospace domains. Moreover, a lot of SCADE Suite features already suit the GENCOD project requirements.

The first step of the flow consists in modelling textual requirements with the SCADE language. SCADE is a graphical and textual high level modelling language. This formal and unambiguous language is able to describe the architecture and the behaviour of synchronous digital systems. As the SCADE models are executable specifications, we can then run test cases with the built-in SCADE simulator, and performs some formal verification of properties, to improve safety assurance level of the design.

Once the SCADE model is checked and fits the requirements, the next step in the flow is the automatic RTL code generation. In the initial tool, SCADE Suite only provides a qualified C code generator (DO178B Level A compliant). For VHDL RTL code generation, several concurrent approaches are being evaluated and will be exposed later in the article.

To comply with the GENCOD certification objectives, a transversal traceability process has been added to the flow. SCADE Suite (RM Gateway) operates with the collaboration of the Geensoft Reqtify tool to offer requirements to model, and model to RTL code traceability.

Besides, Reqtify supplies automatic reports generation, like traceability matrix, which improves workflow productivity.

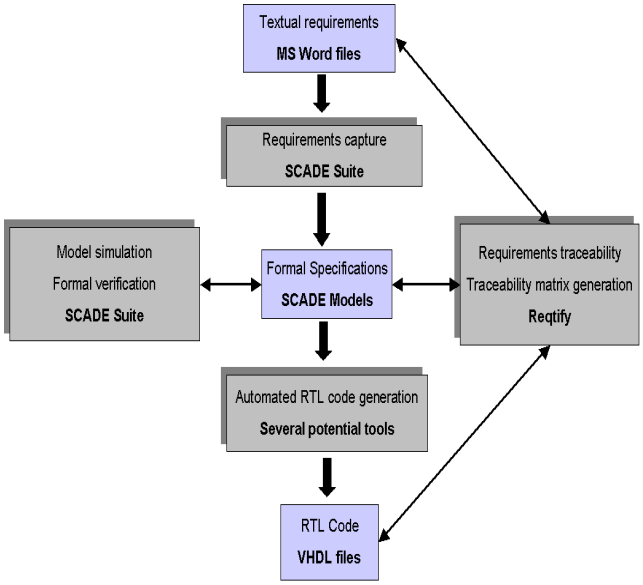


Figure 2 : GENCOD project design flow

The novelty of this flow is to start by a modelling activity in SCADE.

SCADE is being developed specifically to address mission and safety-critical embedded applications. it is certified/qualified according to some international safety standards : DO-178B qualified up to DAL A for Military and Aerospace Industries, IEC 61508 certified at SIL 3 by TÜV for Heavy Equipment, and Energy, EN 50128 certified at SIL 3/4 by TÜV for Rail Transportation, IEC 60880 compliant for Nuclear Energy.

For Aerospace usage, SCADE is commonly used by systems and software designers; the extension for hardware designers may be an advantage; these designers may share the same models of a design, allowing better concurrent engineering, better share of constraints, better validation of requirements.

9. Airbus case study

Each industrial partners is experimenting the GENCOD design flow with its own case study.
The COMETH FPGA is a representative example of control intensive design embedded in Airbus aircraft electronic system. It performs an interface between a physical Ethernet module and multiple hosts in a communication computer (Airbus A340 ATSU) through a proprietary internal bus, ECSB.

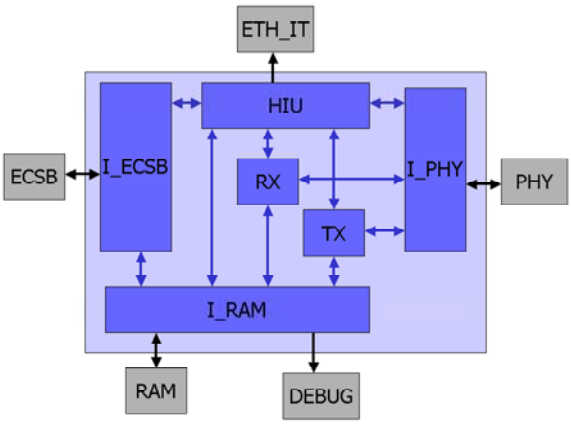


Figure 3 : Cometh FPGA block diagram

It has been already designed using the traditional design methodology and further comparisons with GENCOD results could be proceeded.

10. Formal hardware specification with SCADE

A SCADE model is built from hierarchical block diagrams and the basic functional element is called “operator”. It supports mixed data-flow and control-flow description.
Initially designed for software development, SCADE suite did not provide appropriate types for hardware modelling like bit vectors. Therefore, Esterel Technologies has to develop new types and their dedicate hardware libraries.

For Cometh FPGA case study, most of the textual requirements have been translated using SCADE safe state machines (SSM) [11]. SSM provide very interesting features regarding traditional state machines, especially in control-intensive design.

SSM support hierarchy which help to raise abstraction level and allows a more direct translation of requirements. The communication links between nested state machines are provided by pre-emption and synchronized transitions. With these features, the communication is simpler to express than in traditional synchronised concurrent state machines.

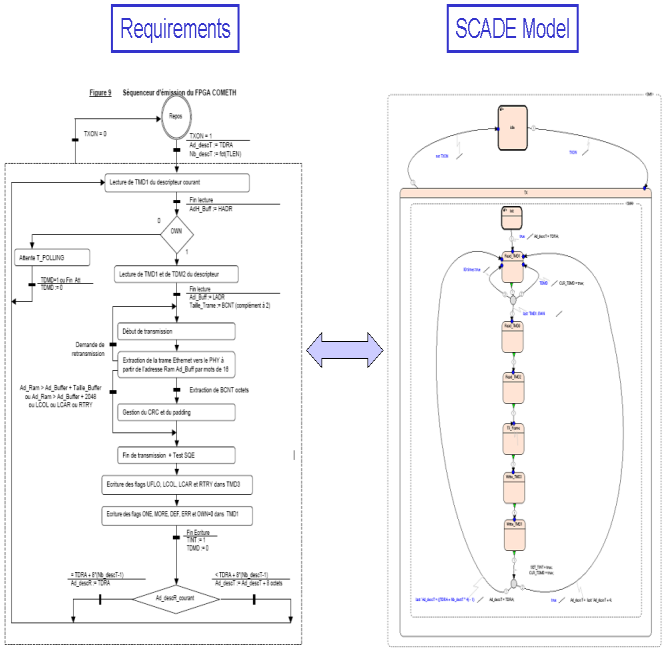


Figure 4 : Formal specification from requirements (FPGA Cometh TX block)

In addition, the SSM transitions priority avoids ambiguous specifications and makes the model easy to read. In the same way, nested dataflow capabilities simplify specification description as well as they help abstraction.

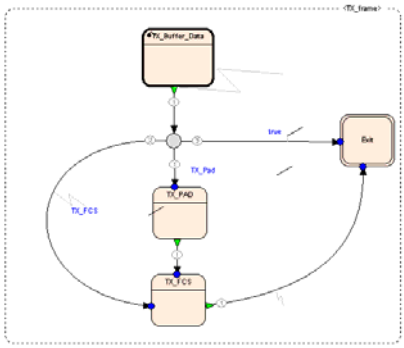


Figure 5 :SSM transitions priority

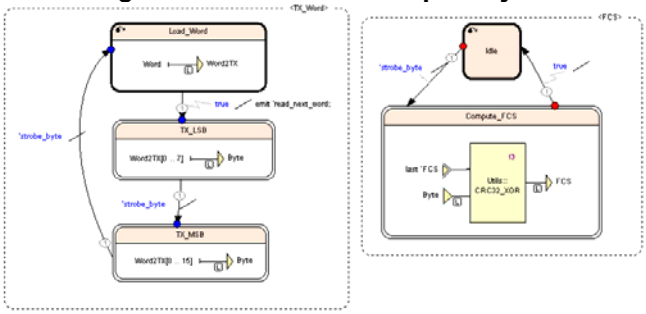


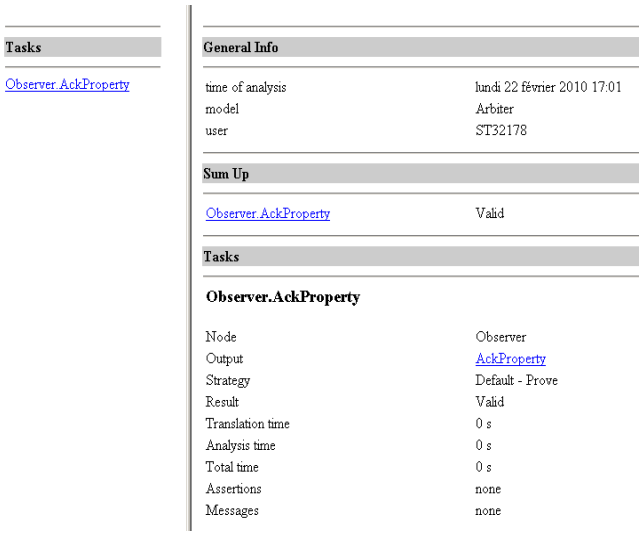
Figure 6 : SSM nested dataflow

As long as the Cometh FPGA is one clock domain design, nearly all the requirements could be

modelled. However, special hardware features, like high impedance 3 state buffer, are not supported yet.

11. Simulation and Verification

Beyond hardware specification modelling, GENCOD project partners are testing the additional capabilities brought by model simulation and formal verification.
SCADE simulation environment is very basic compared to other VHDL code simulator, like ModelSim. The model can be run interactively clock step by clock step when changing inputs values. For more complex and automatic testbenchs, TCL scripts need to be written.
In order to keep only one simulation environment and not write the testbenchs twice for SCADE models and RTL code, Esterel technologies propose to develop a co-simulation tool.
Useful in safety-critical design verification, DesignVerifer, the SCADE formal proof engine, has been experimented to prove temporal properties. In the Cometh RAM arbiter for instance, RAM access need to be acknowledged within a predictable time slot for each client. Once the excepted property is modelled in SCADE langage and link to the target DUT, DesignVerifier can parse the model execution space to find a counter example. Formal verification allows saving time as the user does not have to write testbenches to check the property



Tasks

Observer.AckProperty

General Info

time of analysis	lundi 22 février 2010 17:01
model	Arbiter
user	ST32178

Sum Up

Observer.AckProperty	Valid

Tasks

Observer.AckProperty

Node	Observer
Output	AckProperty
Strategy	Default - Prove
Result	Valid
Translation time	0 s
Analysis time	0 s
Total time	0 s
Assertions	none
Messages	none

Figure 7 : Formal verification report

DesignVerifier can also be useful to help design debugging. Actually, if the property is falsifiable, it provides a counter scenario that be launched and analyzed in the SCADE model simulator.
The major issue when using DesignVerifier is to find the good way to model the properties to be checked, but also to write accurate assertions to limit execution space.

12. Exploration of RTL code generation

RTL code generation from SCADE models is one key element of the GENCOD project.

When the project started, the only generated output code available from SCADE Models was C code. The first experiments to generate RTL code consisted in using COTS C to RTL generators. The results produced by Altium C-to-H or Impulse C-to-FPGA didn't match with the partner expectations. There were too many steps to format the original C code for the target generator, and the FPGA resource utilisation, after synthesis of RTL generated code, growths about 3000% compared to handwritten code. Actually, most of C to RTL tools available on the market can properly manage algorithmic C code, but GENCOD case studies are more control-intensive oriented.

A more convenient approach was found by taking a closer look at C code generated by KCG (the SCADE qualified C generator). Indeed, the data-flow structure of the KCG C code allows an easy syntax translation of the C code into behavioural (sequential description) VHDL RTL code. This pattern has been implemented into the KCG2VHDL generator tool by Geensoft (derived from the FerroCOTS project HDL generator tool). This integrated tool is simpler to use than COTS C to HDL tools and the first experiment results, on early beta versions, shows suitable performances. The FPGA resource utilisation of the synthesized RTL code is moderate and the generated code is rather easy to read (better for traceability).

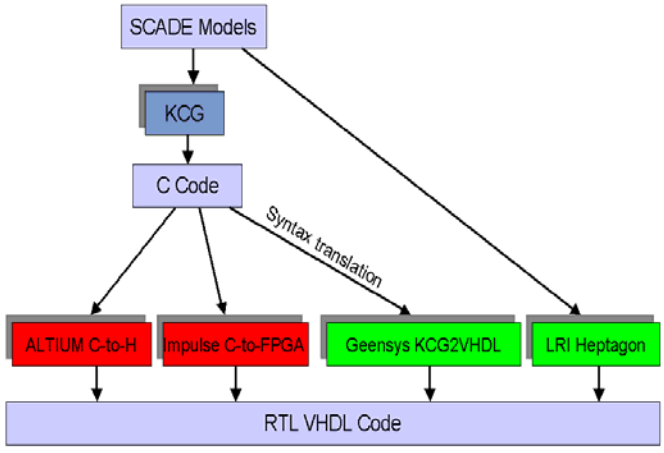


Figure 8 : RTL code generation experimentations

The following table gives some comparison results between generated VHDL and manual, from different solutions.

Case study	Manual VHDL	VHDL (a)	VHDL (b)	VHDL (c)
IRQ CTRL	3 Slices	98 Slices	N/A	3 Slices
CRC (*)	10 LC	396 LC	346 LC	30 LC

Table 2 : FPGA resources utilisation (RTL code synthesis)

(a) : VHDL generated by Altium tool
(b) : VHDL generated by Impulse tool
(c) : VHDL generated by KCG2VHDL tool
(*) note : case study from Thales Communications

The following tables give some elements for comparison between SCADE modelling, and generated C and VHDL.

Case study	SCADE elements	Generated C lines	Generated VHDL lines
FUNCTION TX	507	5140	5500
FUNCTION I_RAM	38	700	730
FUNCTION I_PHY	216	1370	1465
FUNCTION HIU	572	1800	2450

Table 3 : Comparison of modelling in SCADE and generated C and VHDL)

We can note that the generated lines, either from C or for VHDL are equivalent. The tool transformation C to VHDL doesn't introduce additional and numerous lines.

CASE STUDY	RATIO GENERATED C LINES / SCADE	RATIO generated VHDL lines / SCADE
Function TX	10,14	10,85
function I_RAM	18,42	19,21
function I_PHY	6,34	6,78
function HIU	3,15	4,28

Table 4 : Comparison of complexity in SCADE and in generated C and VHDL

We can note than the complexity ratio (if SCADE is baseline 1) is between 3 and 19. Writing a model in SCADE allows reducing complexity, compared to write directly VHDL.

Additionally to the previous proposition, the LRI laboratory is also working on direct RTL code generation from SCADE model. This transformation is based on Heptagon, the SCADE compilation kernel. The first prototype is expected soon for experimentation.

13. Traceability needs

Requirements traceabilty is an important part of certification process. The designers have to add dedicated tags in the design documents and VHDL code to allow requirements traceabilty. Geensoft's Reqtify is a leading tool in this domain. This tool is able to analyse heterogeneous documents to find tags that identify requirements and coverage links.
Once integrated in SCADE Suite, as the RM Gateway tool, it helps to define coverage links between the textual requirements with SCADE model elements.

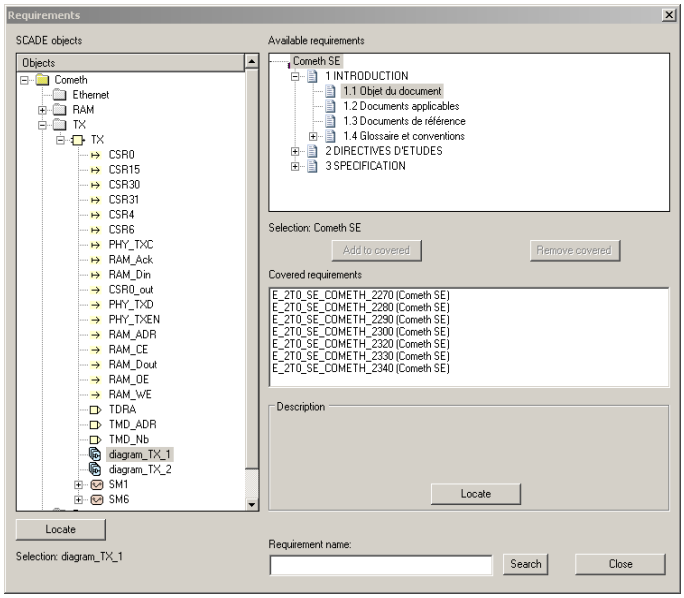


Figure 9 : RM Gateway tool

With Reqtify tool, productivity is improved as requirements coverage report, traceability matrix generation, impact analysis can be automated.

To complete the traceability process, Esterel technologies is working on propagating traceability information to the HDL code. This step is still needed as the RTL code generation is not qualified.

At that point, GENCOD partners also wonder how to express coverage links with derived requirements (formal specifications needed in the SCADE model but not directly related to textual requirement) that are not considered in the initial tool.

14. Tool qualification concerns

Tool qualification is also a key issue for safety critical applications. It is the process necessary to obtain certification credit for a tool within the context of a specific airborne system. The Qualification of a tool is needed when processes of a flow are eliminated, reduced or automated by the use of a software tool without its output being verified.

The objective of the tool qualification process is to ensure that the tool provides confidence at least equivalent to the one associated to the process(es) eliminated, reduced or automated.

Only deterministic tools may be qualified, that is, tools which produce the same output for the same input data when operating in the same environment.

In our GENCOD flow, a great part of the flow is already qualified (or data are available to perform it), thanks to the fact that KCG chain SCADE to C is qualified.

So the limited but necessary effort is to assess the C to VHDL chain; which can be done by classical activities such as code rules checking, code coverage, or simulation techniques. In this case, it may be not necessary to qualify the C to VHDL tool.

These considerations were a key driver for the choice of the tools in the flow. The outputs of GENCOD project will give certification data, or means to produce those data, regarding the tools used in the flow.

15. Conclusion

The GENCOD project is still under experimentation, but first results are encouraging regarding productivity and safety objectives.
First, Formal modelling with SCADE language leads to a clearer and less ambiguous specification. Advanced features of SSM allow raising abstraction level compared to traditional state machines.

As far as the proposed verification environment is concerned, it doesn't provide all the features that hardware designers are used to find in that kind of tool. Hopefully, co-simulation with HDL simulator might solve this issue.

The use of formal verification on SCADE models shows the partners that this approach is efficient to save time to debug and improve design robustness.

After some disappointing experiments with COTS C to HDL generator, Geensoft and LRI brought more accurate answers to RTL code generation. With these tools, both resources utilization and timing performances are expected to be reached.

Finally, from the certification point of view, the design flow is not mature enough. Indeed, the traceability link with the RTL generated code is missing at the moment, and the issue of derived requirements in SCADE models need to be explored.

Ending September 2010, GENCOD project should get close to its main objectives and be a first step in hardware model driven engineering.

16. References

[1] EASA : http://easa.europa.eu
[2] FAA : www.faa.gov/aircraft/air_cert/design_approvals
[3] DO254-ED80 : RTCA-EUROCAE, "*Design assurance guidance for airborne electronic hardware*", 2000
[4] RTCA : www.rtca.org,
[5] EUROCAE : http://www.eurocae.net/
[6] Advisory Circular AC 20-152 : FAA, "*Design assurance guidance for airborne electronic hardware*", 2005
[7] DO178B : RTCA-EUROCAE, "*Software Considerations in Airborne Systems and Equipment Certification*", 1992
[8] Esterel : http://www.esterel-technologies.com/
[9] Geensoft : http://www.geensoft.com/
[10] System@tic : http://www.systematic-paris-region.org/en/index.html
[11] SSM : C. André, "*Semantics of SSM"* (Safe State Machines). Guyancourt, France, April 2003

17. Glossary

A/C : Aircraft.
ARINC : Aeronautical Radio Inc.
ASIC: Application Specific Integrated Circuit
ATSU: Air Traffic Service Unit
COTS: Component Off The Shelf
DAL: Design Assurance Level
EASA: European Aviation Safety Agency
ECSB: Embedded Computer System Bus
ESL: Electronic System Level
EUROCAE: European Organisation for Civil Aviation Equipment
FAA: Federal Aviation Administration
FPGA: Field Programmable Gate Array
HDL : Hardware Description Language
IP: Intellectual Property
LRU: Line Replaceable Unit
PDF: Portable Document Format

PLD: Programmable Logic Device

RTCA: Radio Technical Commission for Aeronautics

RTL: Register transfer Level

SME: Small and Medium Enterprise

SSM: Safe State Machine

USB : Universal Serial Bus

VHDL: VHSIC Hardware Description Language

VHSIC: Very High Speed Integrated Circuit