Designing Space Applications Using Synthesisable Cores

Sandi Habinc

European Space Agency
European Space Research and Technology Centre (ESTEC)
Postbus 299, NL-2200 AG Noordwijk, The Netherlands
Tel. +31 71 565 4722 Fax. +31 71 565 4295
sandi@ws.estec.esa.nl

ABSTRACT

Very high integration levels of microelectronics will be required to fulfil the ever-increasing demands for high processing performance, low mass and power. With an increasing number of available gates on silicon, the functionality being implemented will move away from the use of traditional components to more advanced and complex systems within a single device. To develop such complex circuits the design methodology will have to change from being gate-level oriented to the integration of complex building blocks. The designers will have to rely on pre-existing building blocks with already verified functionality, with documentation and production test vectors being available, and which have ultimately been validated on silicon.

I. INTRODUCTION

This paper will concentrate on how building blocks for microelectronics are developed in VHDL and how they are purchased, distributed and used in the scope of European Space Agency (ESA) activities, ranging from in-house developments to contractor work and from simple Field Programmable Gate Arrays (FPGAs) to complex System-On-a-Chips (SOCs).

A background to ESA microelectronics developments and usage of VHDL will be given. This is followed by a discussion on how VHDL cores are developed in ESA contracts, giving a listing of cores available or under development. Different aspects related to the purchase of Intellectual Property (IP) products will be discussed, giving a listing of commercial cores available through ESA or currently under negotiation.

The paper will also give examples, based on real flight developments, of good and poor design methodologies when developing FPGAs. As a comparison to what is achievable in FPGA technology, a current SOC development will be presented.

II. BACKGROUND

VHDL was introduced in ESA developments in the early nineties, then being the only Hardware Description Language (HDL) supported by multiple tool vendors. During this period the ASIC (Application Specific Integrated Circuits) design process was formalised for ESA developments. In such developments a VHDL model is normally required when an ASIC is designed. This allows the functionality to be independently verified by simulation.

A logical follow-on activity is to model and simulate complete board designs using VHDL. Board-level simulation can therefore be defined as simulating the functionality of one or several printed circuit boards built with standard components, possibly incorporating ASICs and Application Specific Standard Products (ASSPs). Models for board-level simulation have been previously developed by ESA to promote the usage of standard components in spacecraft [1]. While distributing VHDL source code would require a minimum effort, since the user would be made responsible for the adaptation for different simulators if necessary, such an approach was found not acceptable for reasons of protecting design information. The company that designed the component was understandably unwilling to let the information become available to its competitors, especially in those cases where the VHDL code was synthesisable. The availability of VHDL source code would have encouraged the redevelopment of similar devices leading to increased costs for ESA as well as significantly decreasing the interest for foundries to support the devices as ASSPs.

But with the constantly shrinking number of foundries offering space qualified ASIC processes the ASSP approach is receiving lesser attention. Instead, VHDL cores of key components are now made available to European space companies allowing them to reduce their dependency on one silicon supplier and also to design their own custom SOC solutions which will ultimately generate business for the foundries. While all models could be freely distributed in source code, such a service would not be feasible since the cost for maintenance and distribution would not be covered.

ASSPs are still being developed under ESA contracts but the conditions have changed compared to earlier developments. ESA had previously little or no rights to the VHDL code developed in its activities. Nowadays the developed VHDL code is licenced to ESA, being given the right to use it in the field of space related activities. In some cases when the company wants to protect their investments, a licensing model is used which allows ESA to reuse the VHDL code, by itself or through a third party, should the developed device cease to be sold or supported as an ASSP.

In the following sections some ESA activities related to VHDL cores will be presented. The intention with the listing of the cores is to provide a catalogue of cores developed or distributed by ESA and at the same time illustrate the different development approaches that have been chosen for each specific case.

III. Developing VHDL Cores

For functions that cannot be found in commercial applications it is necessary to develop the corresponding IP products from scratch or to find a space company willing to license their internal IP cores if existing. In this section a description of different approaches to develop VHDL cores will be given, concentrating on background, functionality, ownership and distribution rights.

Although IP products cover a broad range of possible formats such soft cores, hard cores etc., ESA is currently concentrating its efforts on VHDL source code development. Source code provides the flexibility, portability and controllability required when designing mission critical space applications.

A. CCSDS Telemetry Channel Encoders

Reed-Solomon and Convolutional encoding have been used widely on European spacecraft and are endorsed by ESA standards and Consultative Committee for Space Data Systems (CCSDS) recommendations. The first European radiation-hard Reed-Solomon encoder was developed in the eighties and the design belongs to the design house. When it was announced that the manufacturing of the component would be discontinued a new development was initiated by ESA.

Instead of directly developing an encoder device for a specific technology, it was decided first to develop reusable encoder cores. As a second step, a device based on these cores would be developed and supported as an ASSP by a new foundry. Synthesisable VHDL models of the encoder cores were initially developed by ESA [2]. Further development was performed by industry who also performed extensive verification of the VHDL models versus the discontinued device. The property rights of the modified and verified code remained with ESA.

In agreement with NASA and other national space agencies it has been decided to include a set of Turbo codes in a new issue of the CCSDS telemetry channel coding recommendations. Turbo coding will be an add-on option to the recommendations without modifying the existing coding schemes and will retain compatibility with the CCSDS packet telemetry recommendations. Turbo coding will be an alternative to concatenated Reed-Solomon and Convolutional coding for deep space and near-Earth missions. In parallel with establishing the new recommendations, an initial Turbo encoder VHDL model has been developed by ESA for which it has the full property rights [3].

B. CCSDS Packet Telemetry Encoder

A CCSDS compliant Packet Telemetry Encoder chip set was developed in the beginning of the nineties under ESA funding. In a normal configuration one would need about five components and four memory chips to build a complete encoder. These components have been recently discontinued by the foundry which has prompted ESA to release some of the original design information to companies developing their own replacements.

However, the use of the original code is not simple since it requires good knowledge of the application and part of it is not written in VHDL. It is also difficult to port the designs since they contain multiple clock domains and are described on a low abstraction level. Even schematic entry had been originally used in one case. An additional complication is that most companies prefer to merge the chip set into one device and at the same time reduce the number of memories. To overcome these problem it was decided to develop a new encoder core from scratch, taking advantage of new process technologies and targeting the new type of applications from the start.

Since there is sufficient knowledge and experience in ESA for this type of telemetry encoders it was decided to develop it in-house. When finished, the VHDL core will be immediately fed into an in-house FPGA development and into a SOC development done in industry, both being presented later in this paper. An important element in the development of this core is that all the previously developed testbenches used for verifying the corresponding models for board-level simulation will be reused. As a final step, the telemetry encoder and the previously presented channel encoder cores will be merged into a resulting CCSDS Packet Telemetry and Channel Encoder VHDL core, for which ESA has the full property rights.

The original CCSDS telemetry channel encoders and CCSDS packet telemetry encoder are currently being distributed by ESA. The code and documentation can currently only be purchased by companies from ESA member states. The cores are bundled and are moderately priced. They come initially without support, but that can be provided under a cost reimbursement scheme.

C. CCSDS Packet Telecommand Decoder

A CCSDS compliant Packet Telecommand Decoder ASIC was developed under ESA funding as part of a technology and methodology study. The ASIC was partially funded by the design house, to whom the property rights consequently belong. The decoder function is considered as an important part of a satellite platform and it is therefore crucial that it can be made available to spacecraft developers without disruptions. When the foundry announced that the process used for the decoder would be discontinued, it was decided to transfer the design to a new foundry. The transfer was performed under ESA funding and the property rights remained with the design house. The same decoder is now planned to be reused by the original design house in the frame of a new ESA funded development. The difference is, compared to the previous two developments, that it is envisaged that ESA will gain access to the VHDL core, acquiring the rights to distribute the code to European companies developing ASICs under ESA contracts.

In addition, a partial decoder has been developed at ESA, but will require further development and refinement before becoming suitable for industrial use.

D. SPARC V7 (ERC32)

In 1992, ESA initiated a technology programme aiming to develop a radiation tolerant general purpose 32-bit processor, the ERC32, initially based on three devices. The development of the single chip version, the ERC32SC, is now being completed. The ERC32 chip set and the ERC32SC device are available from a European foundry [4].

When the ERC32 development was completed, VHDL models intended for board-level simulations were made available to the design community. The code was however not synthesisable. The models have not only been used in several board developments targeting the space market, but have also been used as reference designs during peripheral and software driver developments for commercial applications.

In parallel with the above VHDL models, which were developed by industry, an instruction simulator for the ERC32 was developed by ESA using the C language. Also this tool has been used by non-space companies/organisations, e.g. the simulator is often used as a reference and test vehicle in GNU software developments. Both the VHDL models and the instruction simulator are freely available on the net.

E. VMEbus Controller (EVI32)

Until now the interface between the aforementioned ERC32 chip set and the VMEbus has been implemented using discrete logic or FPGAs, resulting in larger board designs, lower reliability, higher power consumption and higher cost than would be the case when implemented in an ASIC. With a large number of VMEbus based ERC32 systems being foreseen, it has been identified as crucial to develop an ERC32 specific VMEbus interface device.

The EVI32 device is a 32-bit ERC32 interface circuit designed to interface the ERC32 processor chip set to the VMEbus. The EVI32 device fully adheres to the IEEE 1014-1987 VMEbus standard and is compatible with the commercial VMEbus specification. EVI32 can act as a system controller and provides both master and slave interfaces.

The EVI32 VHDL design is an in-house ESA development for which the property rights belong to ESA [5]. The code and the specification are available on the net and have already been used by several European and non-European companies in their internal FPGA developments. In one case a non-European company has re-targeted the VMEbus controller to an Intel processor bus and shipped the design in less than a month. Another FPGA implementation is scheduled to be flown early next year. This kind of fast development has resulted in useful feedback from industry, allowing quick improvements and corrections of the core.

The EVI32/ERC32 concept has thus been promoted by FPGA developments and is now being consolidated by an ASIC development under ESA contract. The same, but updated, VHDL code is used in this development. The design will be modified also to interface the ERC32SC single chip device and potentially the space qualified TSC21020E DSP. ESA will have the property rights to the modified VHDL code.

F. SPARC V8 (LEON)

The LEON project was started by ESA in late 1997 to study and develop a high-performance processor to be used in European space projects. The objectives for the project were simple: to provide an open, portable and non-proprietary processor design, capable to meet the identified requirements for performance, software compatibility and low system cost. One objective is to be able to tolerate the use of a Single Event Upset (SEU) sensitive semiconductor process. To maintain correct operation in the presence of SEUs, extensive error detection and error handling functions are needed. The goals is to detect and tolerate one error in any register without software intervention, and to suppress effects from SEU errors in combinational logic.

The LEON processor is based on a new SPARC V8 core in-house development by ESA [6]. The choice of the SPARC architecture will guarantee full software compatibility with the present ERC32 and ERC32SC based systems. The core will include a floating-point unit capable of executing both single and double floating-point operations. For performance reasons, a separate data and instruction cache is used.

To allow rapid prototyping and porting, the processor is implemented as a foundry independent, synthesisable VHDL model. It does not require any custom macro-cells apart from synchronous SRAM used for the caches and register files. The model is extensively parametrisable: register window size, cache size, fault-tolerance functions and clocking scheme can be defined through a single configuration file. The VHDL model will be made available in two versions: a basic, fully synthesisable model will be freely available, while an enhanced model incorporating the fault-tolerance features, PCI interface and floating-point unit could be licenced to European space companies. By freely distributing the basic design, thereby increasing the user base, it is believed that improvements and correction to the design will be contributed in the same way as with open-source software.

When the LEON development is completed, it is planned to manufacture a prototype device in a commercial technology for further evaluation. After the evaluation, a radiation-tolerant device suitable for space applications could then be manufactured on the best available process at that time.

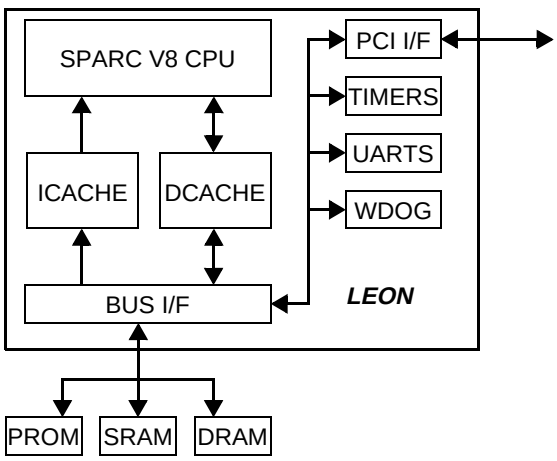

Figure 1: LEON block diagram

G. PCI Local Bus

The PCI Local Bus has initially been baselined as an inter-chip interface for microprocessors and peripherals in on-board applications. It is foreseen that the interface will be extended to encompass compact PCI for on-board equipment. The ESA approach for acquiring PCI VHDL technology will most likely be twofold. For projects where schedule and risk are to be minimised a commercial VHDL core will be purchased from an IP provider. The first approach will be touched upon later when discussing how to purchase VHDL cores. For projects that can initially live with a core that is not that well supported and validated, an in-house VHDL core development has been initiated. The core design is completed and partially verified. The next step will be to validate the core with respect to the PCI standard and checklist using an independent VHDL testbench from a commercial IP provider. The purpose of this next step will be to have eventually an in-house developed core that can be freely offered to industry under the same conditions as the aforementioned SPARC V8 LEON core.

H. PSS-04-255 OBDH Bus Terminal Modem

In the frame of a PSS-04-255 Bus Terminal modem development under ESA funding, a synthesisable VHDL model of the design is being negotiated as an additional output. The PSS-04-255 standard has been submitted as a candidate to the CCSDS standardisation of on-board bus interfaces.

I. Controller Area Network (CAN)

The CAN bus is currently baselined for an ESA spacecraft development. The bus controller, used by the platform and payload, will be implemented in an Actel RT54SX32 FPGA. The prime contractor will either design the controller or purchase a VHDL core from a commercial IP provider specifically for this spacecraft development. In parallel, an in-house reference VHDL design is being developed by ESA to be used for comparison with the industrial results and to facilitate future developments. The final objective of this in-house design is to have unrestricted long term access to the bus technology and to secure it for space industry. The design is freely available from ESA.

J. Wavelet Image Compression Engine

A Wavelet Image Compression demonstrator device targeting a commercial process is being developed under ESA contract. One of the outputs will be a synthesisable OCAPI based C++ description of the device which can be used for re-targeting the design to a space qualified technology. ESA will have the right to provide the core to any contracted European company involved in the development of similar devices as long as their use is strictly limited to the space segment. The C++ approach was chosen because of the high complexity of the device and since compression is an algorithm intense process that is well suited for description at a high abstraction level. The OCAPI based C++ approach allows generation of VHDL as well as VERILOG source code.

IV. PURCHASING VHDL CORES

Developing cores, both in-house and through European space companies, gives ESA high flexibility to use and distribute cores. However, it is not always the best approach. For some specific functions it is better to purchase existing cores from commercial IP providers. All such cases have one thing in common, they are all based on well defined and widely used commercial standards.

The two main guiding principles when deciding whether to develop or to purchase a core should be cost and schedule. Both factors are tightly coupled to ones design capabilities and experience with the specific function.

For the cores that are being considered for purchase a trade-off between suitability for space applications and development schedule is to be performed. Since on-board applications are sometimes implemented in technologies that are prone to single event upsets, the purchased cores must often be modified to reduce their susceptibility to such effects. These modifications are not always simple and often require complete re-verification/validation of the core before it can be used. In some cases the effort for modifying a commercial core would require less effort than to develop the core from scratch, even though it would require a larger number of gates in the end. This drawback has been seen as acceptable since the overall development schedule is reduced and core specific knowledge investments are avoided.

The quality level of commercial cores, defined by functional correctness, should not be forgotten during the decision process. Much effort has been spent on verifying the correct functionality of commercial cores to be sure that they do not introduce bugs in our designs. In one case the test object was first subjected to all verification suites that were provided with the core, followed by pseudo random testing [7]. The reference was represented as a commercial hardware emulator that was part of a software development environment. The stimuli were generated with a custom made C program and output in executable binary. Several bugs were found, likely to jeopardise the functionality of the design if not corrected. This demonstrates that independent verification of a purchased IP core could be an important investment.

Another drawback with commercial cores that has been experienced is that the documentation and the commenting of the code is often poor compared with our requirements [8].

One possible licensing approach that is being considered is that ESA pays an initial acquisition fee for a core and that companies involved in subsequent developments would benefit from a discount scheme in developments funded by ESA. All licensing and contractual aspects would be handled by the IP provider, thus avoiding further ESA involvement.

As a conclusion, there are many questions to answer when selecting an IP core and a generous amount of time should be set aside for the evaluation and selection process. The following important list of issues that need to be addressed has been published in Computer Design [9].

Concerning model issues:

Is the code readable, commented and documented? Is the model synthesisable using my design tools? Is the model easily configurable in terms of bus size etc.? Is the model fully synchronous? Is it possible to fully reset the model? What is the fault coverage of the test vectors?

Concerning credibility:

Has the IP core been implemented in an ASIC? Is there already any commercial device using the IP core? Are there any existing customers whom you can talk to?

Concerning extra services:

What is included in the support? Do you have direct access to a support engineer? Is it possible to have training? What is the bug fixing policy? Does this policy include the notification to all users of all bugs reported?

Concerning verification:

How has the model been verified? Is there a test plan and a test suite delivered with the IP core? Was the test suite defined by the IP provider or a device manufacturer or in a standard? Can I reproduce the test in my environment?

The following sections will concentrate on a commercial VHDL core which has been purchased by ESA and other cores that potentially will be purchased.

A. Intel 8032 Compatible VHDL Core

In 1997, ESA initiated an activity in which a commercial 8032 core was purchased after a commercial and technical evaluation. The final product from that activity is a design featuring the 8032 core together with peripherals and functions dedicated to space applications. The 8032 core and the several peripherals developed within the activity are available to European space industry for ESA funded ASIC developments. ESA has paid an initial acquisition fee and companies participating in subsequent ESA developments will only need to pay a smaller instantiation fee. The core is sub-licenced by ESA to its contractors. This scheme will reduce the cost as compared to having each company choosing their own IP provider and consequently paying redundant acquisition fees.

The 8032 core purchased is based on an architecture developed by Intel, but the actual synthesisable VHDL core has been developed by a European company. By having the core available, the European space industry can benefit from both the widely used Intel 8051 microcontroller platform and functions dedicated to space applications.

The drawback with the core is that it does not fit in any currently available FPGA suitable for critical space applications. This could potentially be overcome if the on-chip memory is placed externally, but such an experiment has not yet been performed. The baseline is instead to develop an ASSP using the core and add additional on-chip memory and features. It is foreseen to use commercial high density FPGA for prototyping during the ASSP development. It will enable early software development and provide user feedback.

B. Other Cores

Although a PCI core is being developed by ESA it has been identified as important to have a commercial high quality VHDL core available to European space industry. The commercial core would be targeted to developments where little risk is allowed and where proper product support is expected. The purchase of such a core is being discussed with some potential IP providers. A reuse scheme will be established with the IP provider enabling European space companies to purchase the core at a discounted price when involved in ESA developments.

The next purchase that is being negotiated is an IEEE Std-1355 VHDL core, also known as SpaceWire, that has been developed by a European space company. This point-to-point interface has been submitted as a candidate to the CCSDS on-board interface standardisation.

Also the IEEE Std-1394 interface, known as FireWire, is being currently considered for general on-board interfacing, also covering dedicated applications such as cameras. Whether a core will be purchased or developed is under investigation.

Finally, a Mil-Std-1553 VHDL core is available from a European company independent from the large space system houses. ESA is currently not planning to purchase and distribute the core since it is already available in Europe under conditions that are meeting our objectives.

V. Managing Cores In ASIC Developments

An attempt to defining a VHDL core policy has been made in one specific microelectronics development funded by ESA. The main points that were addressed are listed hereafter.

In the case VHDL code is provided by ESA for a particular function, ESA shall retain the ownership of any updates made to the code by the contractor, with an unlimited licence to use the code being granted to the contractor.

In case commercial cores are to be integrated on the device, ESA shall hold any commercial core licence. As a baseline, the licence shall allow future new designs by other companies than the contractor with no or limited additional cost.

If a foundry holds the property rights to a core incorporated in the design, ESA shall be guaranteed worldwide rights to us the device for space applications without any licensing fees.

For the complete design and for each building block developed in an activity, ESA shall be granted the ownership of the design and the associated VHDL code that has been produced. ESA shall be guaranteed the right to use the design or building block for space applications without any licensing fee in the case that the contractor or foundry does not make it available for ASIC design for space applications. Also for this aspect the baseline should be that the licence shall allow future new designs by other companies than the contractor with no or limited additional cost.

VI. Developing FPGAs

A. Design Methodology

One problem when designing FPGAs for on-board applications is that the design methodology used is often more casual than when developing ASICs. FPGAs are perceived as simple to correct and easy to modify late in the design process. These beliefs often lead to design methods that would never be acceptable for ASIC or software design.

When boards carrying FPGAs are designed as part of spacecraft or unit development, the FPGAs are often considered as part of the board and are treated as black boxes. The FPGAs are assumed to be covered by the board specification and there are seldom any specific documents associated with the FPGAs. From our point of view, being involved in reviewing the designs, it is difficult to assess the correctness of the FPGAs. Things become even more difficult when one is summoned to solve problems discovered late in the development. Problems are always discovered late since nobody has bothered to verify the FPGAs properly before integrating them on the board. Since there are no specifications for the FPGAs it is also difficult independently to verify their correctness. The design is often verified by the designer only, this being a great potential for error masking.

The recommendation is always to use a proper design methodology when designing FPGAs. The same level of documentation and verification should apply as in ASIC developments. Two documents have been used in many ESA developments. The ESA *ASIC Design and Manufacturing Requirements* [10] and *ASIC Design and Assurance Requirements* [11] are written for ASIC developments but most parts are relevant to FPGA design as well. The *VHDL Modelling Guidelines* [8] cover several issues related to HDL based design without being tied to any particular tool.

B. Developing an FPGA Using VHDL Cores

The first serious in-house attempt made by ESA to design an FPGA using cores is a CCSDS compliant packet telemetry and channel encoder to be used in an in-flight transponder experiment.

The previously described Reed-Solomon and convolutional encoder cores were used for the channel encoder part. The integration of these encoders was fairly simple, since the main specifications were met by the cores and the maximum interleave depth of the Reed-Solomon encoder was a configurable parameter. The only modification necessary was to place the 1280 memory elements required for storing the check symbols outside the FPGA in a static memory. The third channel encoder, for Turbo codes, had previously been developed in VHDL mainly targeting an FPGA and had already the associated memory elements located in external memory, but since designed for a specific companion chip set in mind, all interfacing and most of the control circuitry required redesign. The resulting VHDL code became more generic in the process and is being refined to the quality level becoming a core.

The packet telemetry encoder has previously been implemented by a chip set comprising an assembler and a multiplexer component. Each virtual telemetry channel required one assembler component and an external memory. The assembler component was originally written in the VERILOG language on a relatively low abstraction level and would require substantial rework before being suitable for the new application in mind. Instead, the assembler and the multiplexer part were both redesigned from scratch in VHDL. The new code was written in a generic way, taking into account future reuse and configuration. The task was difficult since the final telemetry encoder, comprising two virtual channels, had to fit with the aforementioned channel encoders in the same FPGA and share a single external memory. At one point this dual approach was abandoned and part of the code had to be optimised to fit the FPGA. The resulting design comprising one FPGA and one memory device would correspond to eight components if implemented using existing devices.

One conclusion from this development is that properly developed cores seem easier to use than code that was not developed with reuse in mind. In this case it was more efficient to start from scratch, using previous design knowledge, experience and verification infrastructure, than try to comprehend the old code written in a foreign HDL. A second conclusion drawn from this development is that modifications of a core are often required to overcome the low density properties of present flight worthy FPGAs.

C. In-flight Experience Using FPGAs

In this section a few examples will be presented of positive and negative results experienced using FPGAs for on-board applications in ESA related developments. The examples have been chosen to illustrate specific aspects of FPGA design.

One of the first times FPGAs were used in an ESA funded activity was during a bus interface ASIC development. The design was first prototyped using four FPGAs. The design was then transferred to a standard cell ASIC technology. The transfer was made on the gate-level which resulted in many difficulties during layout. Although the FPGAs had been used in operational surroundings during the validation of the prototype, there were several bugs that were not discovered until after ASIC manufacture. The lessons learned from this development are that an application should initially be designed targeting the ASIC technology and only when the code is completely verified should it be partitioned to meet the requirements posed by the FPGAs. It might seem as a good idea to transfer the design on the gate-level, since the FPGAs were already tested, but the disadvantages outnumbered the potential advantages. For example, circuitry required for the partitioning between FPGAs were unnecessarily incorporated in the ASIC. The verification effort for the FPGAs was lower than it would have been for an ASIC since FPGAs are perceived as simple to correct after testing in the operational environment. The verification effort for the ASIC was also reduced since the design had been validated using the FPGAs. Bugs that could have been easily detected during routine ASIC verification were therefore not discovered.

In two other developments it was decided to transfer multiple FPGA designs into a single ASIC which could be configured to operate as one of the FPGAs at a time. This would reduce the cost of the development since the targeted FPGA parts were rather expensive compared to an ASIC solution. It was considered less expensive to qualify one ASIC than multiple FPGA designs. There were also power and reliability performance to be gained by going to ASIC technology. In the first case the transfer was done in nine months, successfully transferring six FPGAs. The second case was not that successful since many of the FPGAs needed to be upgraded or even completed and tested before the transfer could begin. In the end the transfer was not possible due to schedule limitation resulting in unforeseen FPGA usage for the flight model. The lesson learned is again that FPGA prototyping is not a guarantee that a transfer to ASIC technology will be successful.

The first in-house development of mission critical flight worthy FPGAs by ESA was performed in the frame of the TeamSat mission [12]. TeamSat was the first satellite delivered to orbit by the Ariane 5 launcher on its second qualification flight. The 350-kg satellite, which was embedded in Ariane 5's upper test platform, carried five experiments provided by various universities. The entire project, from the initial idea to the end-of-mission, lasted exactly one year. The development, integration and testing were executed by ESA in the record time of seven months. Two FPGAs were developed from scratch in less than six weeks, both located in critical parts of the on-board data handling system. Specifically, all up- and down-link data passed through the FPGAs. The designs were first time right. Lessons learned was that good knowledge of the surrounding system is important when designing interface intensive components and that FPGA specifications need to be understood and agreed upon by everyone involved.

Also on-board the TeamSat satellite was the Visual Telemetry System (VTS), developed by European industry. The VTS comprises a controller that can communicate with up to eight cameras and compress the acquired images. FPGAs were used in both the controller and cameras. The separation images between the Ariane 5 upper stage and the TeamSat satellite were taken by the first Active Pixel Sensor in space.

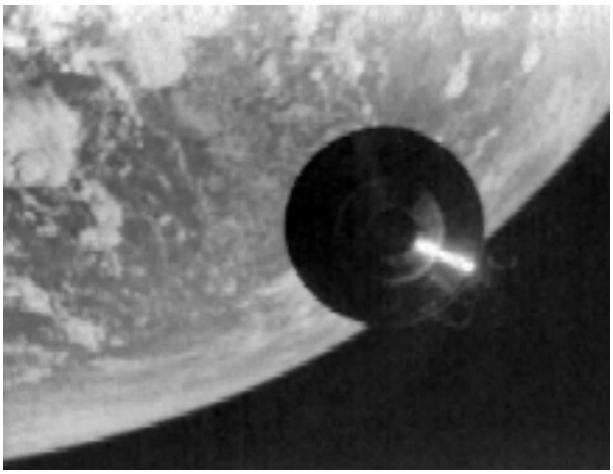

Figure 2: Separation captured by Visual Telemetry System

The success of the VTS experiment lead to the development of the next generation camera also based on FPGA technology. The Visual Monitoring Camera was developed and integrated on a spacecraft in less than one year. Thanks to good visibility into the FPGA design process, by means of proper design documentation and VHDL code delivery, it was possible to discover bugs in the design before the camera was assembled.

VII. Developing a SOC Using Cores

In a current ESA activity a contracting company is developing a SOC application based on various VHDL cores. The objective is to demonstrate that integration is feasible when using VHDL cores from different sources, be that ESA furnished items, purchased IP products or reuse of code from previous developments. The objective is also to demonstrate that an on-board system application can be designed bringing substantial benefits to the development of the spacecraft in terms of power and mass savings and performance enhancements. The challenges faced in the development range from technology issues such as sub-micron layout to legal aspects such as sub-licensing of IP rights.

The foreseen application is based on the ESA developed SPARC V8 LEON core to which a PCI bus interface is added. The peripheral functions include a telecommand decoder, telemetry and channel encoders and a Mil-Std-1553 interface. The complexity of the device is estimated to be in the range of 300 kgates and 300 kbits of on-chip memory. Of the five main cores, two are provided by ESA, two are the property of the contracting company and one will be purchased from a commercial IP provider. The main development effort will be to efficiently integrate all VHDL cores on the die without degrading the inherent performance of each core. Efforts will be concentrated on interconnecting the data paths of the different cores using direct memory access, not to waste the processing power of the microprocessor. It seems not that difficult to put many cores on a single chip, but to make them work together as a system is the real challenge. As the above complexity figures show, it is not possible to design a comparable on-board SOC application using FPGA technology today. For example, the telemetry part in the SOC is three times larger than what could be implemented in an Actel RT54SX32 component. Some parts of the application are being prototyped using commercial high density FPGAs. The drawbacks are that it is still difficult to meet performance targets in terms of clock frequency and amount of on-chip memory when using FPGAs.

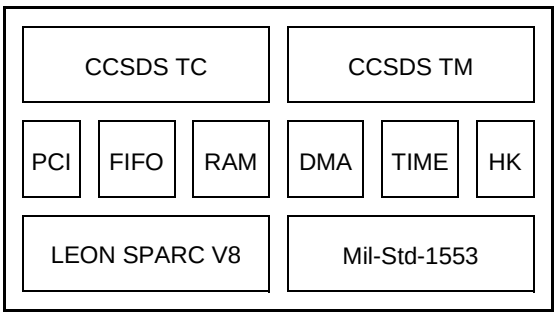

Figure 3: On-board SOC application using VHDL cores

VIII. Conclusions

Designing and using IP cores in FPGA and ASIC developments is not an entirely easy task. Many aspects need to be covered to ensure that a development will be successful. Our experience is that when IP cores are used for flight applications the above task becomes even more difficult.

The VHDL hardware description language has been used in all of our core developments and in most of our ASIC and FPGA developments. Although reuse was an issue when the language was adopted as the standard HDL to be used in ESA developments, it was not until the introduction of a core based methodology that all the language's benefits have been appreciated. In one rare case, VERILOG was used by a contractor for designing an ASIC. When later on trying to reuse the source code, ESA has faced difficulties since no previous investments in training and tooling had been made for the VERILOG language. It was instead chosen to begin from scratch and to develop new code using VHDL.

Developing HDL code with reuse in mind is an expensive task and a pay-off can only be expected when the core is used in more than one device development. A large portion of the cost is related to the inclusion of optional features used for configuration of the core to a certain application. Each feature, or combination of features, requires substantial verification effort which may have been unnecessary if the feature is actually never used later on. The level of detail and clarity in the documentation must also be higher for a core than for a one-off ASIC design, since designers unfamiliar with the core might not only need to understand the code but also modify it.

To purchase a ready made IP product might seem to be the solution to overcome some of the costs associated with developing a core. However, much effort might have to be spent on verifying the correct functionality of commercial cores to be sure that they do not introduce bugs in our designs.

Since on-board applications are sometimes implemented in a technology prone to single event upsets, the purchased cores must often be modified to reduce their susceptibility to such effects. These modifications are not always simple and often require complete re-verification/validation of the core before it can be used. All these factors will increase the actual cost of the purchased core as compared to the listed price.

Licensing aspects will always be difficult if cores are to be used by more than one company or institute. Negotiations can take a long time before agreeable solutions can be found. Ample time should therefore be set aside for these issues when a new IP product is to be introduced in a development.

The use of cores is limited when developing critical on-board applications using FPGA technology since the complexity of flight worthy devices is still low. Only small cores can be used when designing with current anti-fuse FPGAs and is often limiting the application to the functionality of the core itself, not leaving enough space for application specific circuitry. However, for non-critical applications where means exist for reconfiguring the FPGA as part of nominal operation, the use of SRAM based FPGAs becomes feasible and enables use of complex processing cores such as JPEG.

Synthesisable cores have the greatest application potential in large System-On-a-Chip developments where the use of many cores on one die facilitates design on a system level. This application field is however out of bounds for present flight worthy FPGA technologies.

Acknowledgements

The author would like to thank Jiri Gaisler and other colleagues at ESA and in industry for their valuable input.

Point of Contact

Direct questions regarding purchase of VHDL cores to Martin Hollreiser at ESA (mhollrei@estec.esa.nl).

References

[1] *Using VHDL for Board-Level Simulation*, S. Habinc et al., IEEE Design & Test of Computers, USA, Fall 1996, ftp.estec.esa.nl/pub/vhdl/doc/BoardSim.pdf

[2] *A CCSDS Reed-Solomon Encoder Core and Radiation Hard Device*, S. Habinc, Workshop on Microelectronics for Space Applications, The Netherlands, December 1996, ftp.estec.esa.nl/pub/vhdl/doc/Rescue.pdf

[3] *Development Plan for Turbo Encoder Core and Devices Implementing the Updated CCSDS Telemetry Channel Coding Standard*, S. Habinc et al., 1st ESA Workshop on Tracking, Telemetry and Command Systems, The Netherlands, June 1998, ftp.estec.esa.nl/pub/vhdl/doc/Turbo.pdf

[4] *ERC32 Home Page*, www.estec.esa.nl/wsmwww/erc32

[5] *ERC32 VMEbus Interface (EVI32)*, www.estec.esa.nl/wsmwww/erc32/evi32.html

[6] *A Portable Fault-tolerant Microprocessor Based on the SPARC V8 Architecture*, J. Gaisler, Data Systems in Aerospace - DASIA 99, Portugal, May 1999

[7] *Accelerated Verification of Digital Devices Using VHDL*, S. Habinc et al., VHDL Users' Forum in Europe, Switzerland, September 1998, ftp.estec.esa.nl/pub/vhdl/doc/AccVHDL.pdf

[8] *VHDL Modelling Guidelines*, P. Sinander, ASIC/001, European Space Agency, The Netherlands, 1994, ftp.estec.esa.nl/pub/vhdl/doc/ModelGuide.pdf

[9] *Buying soft intellectual property demands a close look*, Y. Trivedi, Computer Design, USA, July 1998

[10] *ASIC Design and Manufacturing Requirements*, P. Sinander et al., WDN/PS/700 Issue 2, European Space Agency, The Netherlands, October 1994, ftp.estec.esa.nl/pub/vhdl/doc/DesignReq.pdf

[11] *ASIC Design and Assurance Requirements*, R. de Marino, QC/172/RdM, European Space Agency, The Netherlands, June 1992

[12] *The TeamSat Data Handling System*, C. Smith et al., First ESA Workshop on Tracking, Telemetry and Command Systems, The Netherlands, June 1998, ftp.estec.esa.nl/pub/vhdl/doc/TeamDHS.pdf