Development of on Board, Highly Flexible, Galileo Signal Generator ASIC

Louis Baguena, Emmanuel Liégeon, Alexandra Bépoix,
Jean-Marc Dusserre, Christophe Oustric, Philippe Bellocq, Vincent Heiries
Alcatel Alenia Space Toulouse (France)

Abstract

Alcatel Alenia Space is deeply involved in the Galileo program at many stages. In particular, Alcatel Alenia Space has successfully designed and delivered the very first navigation signal generator, based on a 0.35µm Atmel ASIC technology, which has been launched in the satellite demonstrator GIOVE-A in December 2005.
The Galileo project is now in a second phase including the development of four of the thirty satellites of the final constellation. The new navigation signal generator requires both high performance and high flexibility (various waveforms to cope with the different Galileo services: open, commercial, governmental ...) for a very long life time system. Besides, the challenge is increased due to the specific space constraints such as mass, volume and power consumption. These requirements will be achieved through the implementation of a 3 million gates ASIC in a 0.18µm European Radiation Tolerant Atmel technology.
This paper will, after a brief description of Galileo system, present the constraints of space environment and technologies challenges. It will then present the ASIC and the development flow of this project, emphasizing the up to date tools that have been used (architectural synthesis, physical synthesis). A conclusion will then be drawn on the requirements on technology and tools for space domain.

1. Galileo system

Galileo, the new European navigation system which will provide an alternative to the American GPS , is one of the great challenges of Europe in this beginning century.

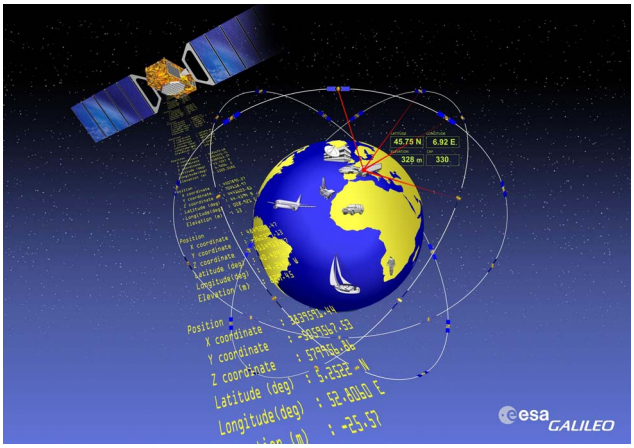
It will guarantee the availability of the service under all but the most extreme circumstances and will inform users within seconds of a failure of any satellite. This will make it suitable for applications where safety is crucial, such as running trains, guiding cars and landing aircraft.

The fully deployed Galileo system consists of 30 satellites (27 operational + 3 active spares), positioned in three circular Medium Earth Orbit (MEO) planes at 23 222 km altitude above the Earth.

Each satellite broadcasts signals to be used by the receiver to determine the user position regarding each of the visible satellites.

10 signals are broadcasted :

- 6 for free services
- 2 for commercial services
- 2 for public regulated services (governmental)



The NSGU equipment (Navigation Signal Generation Unit) is part of the payload core and is responsible for the generation of the Navigation Signals.

The unit includes Navigation Message data handling and spreading by PRN codes, a high level flexible signal modulation engine.

Thanks to digital compensation, the NSGU achieves also very demanding performance, in terms of linearity, phase noise, stability, coherency, delivering also important payload compensation capability.

All digital processing of the NSGU is made by an ASIC named NSGE.

2. Space environment constraints

The electronics embarked on board a satellite are submitted to very specific constraints. First of all, during the satellite launch the equipments are submitted to severe vibrations, and all electronic boards must withstand these constraints. Then during their in orbit life, all components will be subject to irradiations by energetic particles. These particles might either destroy the component (Single Event Latch-up), modify its functionality (Single Effect Upset, Single Event Transient) or modify its electrical behaviour (Idd, Vth modifications by Total Ionizing Dose effects). Therefore, specific technologies and design techniques will have to be used to guarantee the full functionality of the equipment, within the initial specification range, during the satellite life time. In addition to these constraints, a major one lies in the required reduced power consumption. Indeed, the energy on board is provided only by solar cells and batteries, and is therefore to be used parsimoniously! More globally, mass and volume must be reduced as much as possible due to the launcher criteria and associated cost. Finally, last but not least, reliability over the satellite life time (over 15 years for telecom applications) is one of the key points that we have to guarantee.

Most of these constraints (mass, volume, power consumption, reliability) are solved by integration at component level, i.e. the development of ASICs, and when sufficient or complementarily the development of FPGAs. Accordingly, the technologies used for these components must be radiation tolerant or radiation hardened, and also quite mature. This is why the qualification process of technologies suitable for flight models is rather long, and also explains why the space technology generations are well behind the 'commercial ones'.

Nevertheless, since the requirements for embarking more and more computation power are always increasing (e.g., the actual complexity of the signal generator is 3 times the GioveA one), we are also always pushing for using the latest available technologies, and trying as much as possible to prepare the future.

Alcatel Alenia Space has anticipated the needs of such complex digital products for many years. Indeed, the successive development of several test vehicles held in Alcatel Alenia Space Toulouse site has allowed to gradually accumulate knowledge in Deep Sub-Micron (DSM) technologies (0.18µm, 0.13µm) and the associated design flows.

Complementary to this physical expertise, Alcatel Alenia Space has studied for many years the productivity improvements that a C-based design flow can bring, especially because the opportunity to use IPs is very limited in our environment. Optimization of complex systems is indeed the core of our work.

3. NSGE ASIC development flow

The design flow used for the development of the NSGE ASIC is presented in the following chart.

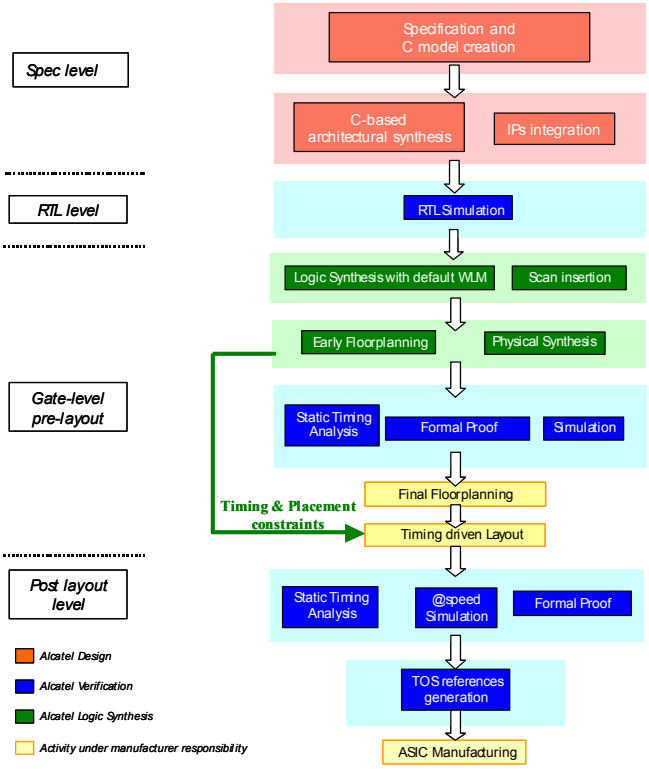


Figure 1 : NSGE ASIC development flow

Two specific areas of this design flow are going to be developed in detail: the C-based architectural synthesis and the Physical synthesis.

3.1. C-based architectural synthesis

The main innovation used for the development of the NSGE ASIC is the C-based architectural synthesis by inserting in the design flow the Mentor tool named Catapult System Level Synthesis.

Indeed, after having evaluated the tool for a while and running pilot projects to validate the technology, Alcatel Alenia Space Toulouse has decided early last year (2006) to deploy the tool on its new ASIC and FPGA projects.

For us, four criteria were essential:

a) Quality of results

Obviously, we couldn't compromise with design quality. In our applications, every gate and every MHz

counts! So the tool really had to produce at least comparable results to our hand coded designs.

Mentor has a large team continuously improving the tool and many customers feedbacks have given to the tool a very interesting and useful orientation. In general, complexity of designs generated by this C-based flow is better than manual implementation and this tendency has been really improved with the maturity of the product.

b) Pure C input

This is really were Catapult fundamentally differentiates from other tools. Catapult System Level Synthesis is the only tool we have found, that reads in pure C/C++ code. It doesn't require any extension (such as SystemC classes and constructs) or worse, proprietary libraries.

This makes a huge difference for us. All our algorithms are written in plain vanilla C code. We didn't want to change our way of writing them, and certainly not wrapping them in useless SystemC code.

Mentor, gives very good information on how to write "good" C in its "Catapult System Level Synthesis Style Guide". In our case, we had a good surprise because most of these rules was already applied by our digital Signal Processing C coder team. So the effort was not really important to deal with Mentor recommendations.

The key thing with pure C, beyond being our language of choice, is that it allows keeping the source code completely implementation independent. Since you don't specify clocks, modules, ports, processes or whatsoever in the source, the tool is truly able to generate very different architectures from the same C code.

A side benefit of this is that the C code is really reusable. It is not the case for our RTL code which is manually optimized for a specific FPGA or ASIC technology target.

c) ASIC and FPGA support

Our primary target is ASIC, but the FPGA prototype is a mandatory step for us. Being able to synthesize RTL for both targets is very important.

Moreover, targeting FPGA allows us to quickly take the C code onto FPGAs for at-speed testing in an electrically and functionally representative environment.

d) C/VHDL verification

In order to verify RTL design functionality, we used to generate reference files from the simulation of the golden C model and next compare them to the RTL simulation results. The test coverage is measured thanks to Modelsim code coverage feature.

The main problem encountered during this verification step is to obtain an exact functional equivalence between C model and the RTL model. Indeed, the way the RTL designer has implemented the design often differs a little from the golden C model, for instance concerning the initial values of counters, memories…

Catapult System Level Synthesis guarantees an equivalence between the input C code and the generated RTL. Besides, a verification flow is included in the tool in order to simulate the input C and the RTL code with the same C test-bench. The tool manages also Modelsim to automatically run simulations from the C test-bench and to compare the results obtained from both sources. Therefore, any difference due to a wrong implementation of the design will be highlighted very early in the flow.

The C-based approach is very different from the VHDL/Verilog one because one writes ‘untimed’ C without any sequencing information. Catapult System Level Synthesis generates the scheduling (a Finite State Machine) according to the architecture constraints given by the designer. The designer has full control upon what gets generated and how. In fact, this tool shall be driven by RTL designers as it will always require good hardware skills to produce good hardware!

The output of Catapult System Level Synthesis is an RTL code (VHDL or Verilog) equivalent with the C code in term of functionality. Catapult generates also the Simulation infrastructure (for Modelsim in our case), and the synthesis scripts (in our case Precision RTL for FPGAs and Design Compiler for ASICs).

If the C code is well written, RTL will be generated in just a few minutes. The tool gives an estimation of area and speed, and we have observed a good accuracy of these estimations for FPGAs and ASICs. Thanks to this fast run time between algorithm and RTL implementation, we have made deeper architecture exploration and really optimized the design implementation, and this, in a very short time.

We also found out that a good communication between C team and RTL team is mandatory, and mentalities must change in order to take the best advantage of this C based methodology. The traditional barrier between algorithm and architecture has been overcome thanks to this tool. The discontinuity between the high level model and RTL implementation is definitely suppressed.

We could expect to make all the type of blocks of such an ASIC with Catapult System Level Synthesis. If anything which is compute intensive or has complex decision-making is a great fit, for complex control, standard RTL flows are still preferable. Therefore, we have been using HDL designer tool for all TMTC/sequencing parts of the NSGE ASIC.

3.2. Physical synthesis

An other innovation in our flow is the insertion of Physical Synthesis step before sending a netlist to the foundry for layout.

During our past experiences on several test vehicles in 0.18µm or 0.13µm technologies, we have observed numerous and often large damages on timings during layout phase, not only on setups but also on holds. Besides in our designs, we have often many hard macros (such as memories) to place on the die and it is really difficult to specify this placement to the foundry, in charge of the layout, without having an idea of the impacts on the design performance.

Thanks to Physical Compiler (PC) from Synopsys and its very useful mode RPP (RTL Performance Prototyping), it is possible to make a physical synthesis without any strict floorplan. As we are working for space, we have to face specific constraints induced by qualification aspects. That is why, despite we are using standard cells, we are nevertheless working with standard predefined matrixes by our foundry (Atmel). This means that some of the constraints are already fixed such as die size, core size, power grids and pad pitch. We just have to place the hard macros and pads thanks to either PC graphic interface or script mode.

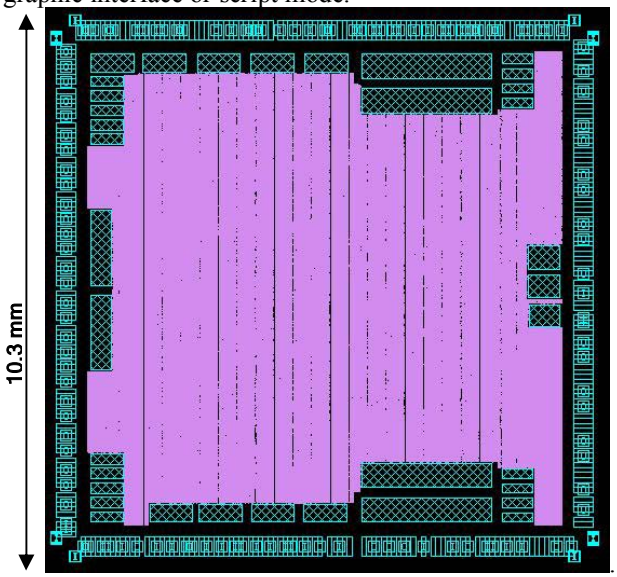


Figure 2 : NSGE ASIC placement made with PC

We always start the physical implementation with a standard logic synthesis (Design Compiler from Synopsys) and scan insertion. Next, we load the gate netlist, the SDC file (timing constraints) generated by DC and the placement constraints into PC. It is important to notice that we are also using PC to insert the spare cells. After having launched a physical optimization, including cell placement and global routing, PC reports the DRC and timing pending violations. This gives us the expected performance of the design after layout, considering that a preliminary calibration of the physical flow has been correctly set. In order to calibrate the flow, a first layout of the whole design, or only a portion of it, is routed from a PC netlist by the foundry, and the obtained performance is compared with the one after physical synthesis of the same netlist. A correction factor is calculated and this ratio will be given to PC for the future physical synthesis.

Physical Compiler gives also very useful information concerning the design routability. Indeed, PC generates congestion tables which gives us the possibility to estimate and optimize the placement to secure final layout at the foundry.

When we are satisfied with performance and congestion, we generate from PC a physical constraint file (.DEF) and a timing constraints file (.SDC), which will be used for the layout by the foundry.

3.3. Conclusion on the development flow

As a very quick summary, we would simply say that the benefits given by Catapult System Level Synthesis, being able to explore several algorithms and reducing the time needed to update a design after a late specification change, have convinced all our engineers, from C coders to RTL designer.

Thanks to such a flow including physical synthesis, we have kept our design schedules, by limiting iterations with the foundry, whereas the number of gates of our designs has greatly increased and the targeted technology is always going deeper in the sub-micron domain.

4. NSGE ASIC

The following digital signal processing functions are implemented in the NSGE ASIC :

- Generation of 1 second period signal (1PPS) for synchronization
- Generation of PRN Spreading Codes
- External secured spreading codes support
- Commercial signal ciphering
- Spreading of Navigation Data
- Digital Modulation and eventually, up-conversion to IF (depending on signal)
- Digital filtering, pre-compensation of DAC distortion and amplitude/phase compensation

The processing is split in two ASIC:

- ASIC1 generates E5AB_I and E5AB_Q signals, which are I and Q channels for E5A+E5B signal
- ASIC2 generates E6_IF and L1_IF signals, which are respectively E6 and L1 signals up-converted to $30.F_0$ carrier.

It has to be noted that only one ASIC NSGE has been developed and can be configured as ASIC1 or ASIC2 by static on board configuration.

The following figure shows the ASIC architecture:

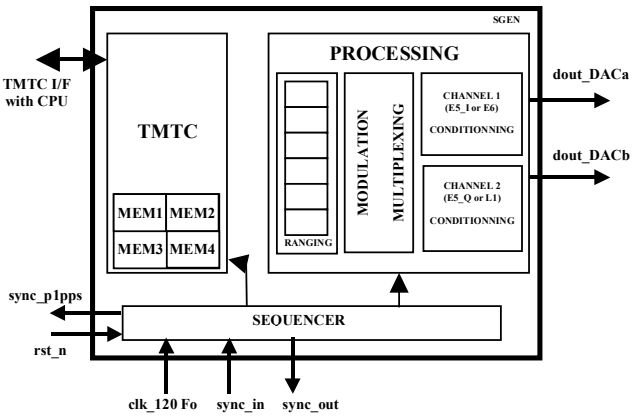


Figure 3 : NSGE ASIC architecture

The sequencer section is in charge of clocks and synchronizations signals distribution inside the ASIC but also synchronization signals exchange between both ASIC and other equipments (security unit and platform computer) to guarantee perfect synchronization.

The TMTC section is in charge of communication with CPU to program all signals parameters and provide navigation data to be spread.

The processing section is the core of the ASIC and comprises 3 parts : ranging, modulation/multiplexing and signal conditioning.

The budget of the NSGE ASIC presented above is summarised in the following table.

Complexity	3 Mgates (including 700 kbit of RAM)
Working Frequency	> 100 MHz
Technology	ATC18RHA (ATMEL 0.18 µm)
Matrix	ATC18RHA95_404 (3.5M usable gates)
Package	MQFPF 256
Useful pins	125
Core Power Supply (V)	1.8 V
Periphery Power Supply (V)	3.3 V
Power Consumption	6 W

Table 1 : NSGE ASIC budget

5. Challenges

Galileo system, and in particular the Signal Generator, requires optimal performance and flexibility, all the more that it is a quite new application in a strong international competition. Furthermore, as the "in space" signal generator will be in operation during at least 20 years, it shall anticipate the future needs and this leverages its complexity. This has caused a lot of difficulties in performance requirement definition for a feasible system in a timely acceptable schedule. Indeed, the "time to market" concept has now become a common notion for space industry, particularly for commercial satellites where return on investment shall be as early as possible. This means that a permanent trade-off/ compromise had to be carried out during all the ASIC development process between:

- Stringent technical requirements and constraints
- Available technology capabilities
- Tight development schedule for the IOV (In Orbit Validation) constellation

This was managed thanks to the process flow that has been described here-above, allowing us a full concurrent engineering approach, with fast, optimal and efficient implementation of any new change that the design team was required to implement. It is noteworthy also that we were able to maintain a full specification performance at well above 100 Mhz system speed, in worst case conditions (the whole temperature range being –55°C, +125°C). This has been achieved on the 0.18µm technology named ATC18RHA, newly proposed by Atmel, France. This technology is QML-V qualified (Qualified Manufacturer List from DSCC) and is undergoing E-QML (European-QML) certification, making it suitable for on-board satellite utilization.

6. Conclusion

A brief overview of Galileo European positioning system has been presented, and a particular development of one of its on-board critical equipment, the signal generator, has been highlighted. A strong emphasis has been made on the NSGE ASIC (the core of the equipment and the focal point of the overall system performance) development flow, particularly showing the interest of two major design tools: one at the high level design, one at the physical design stage. The interest of these tools does not require anymore demonstration for us, and we have been able to manage the critical challenge of high performance/technology/schedule, thanks to their utilization.

We do hope that NSGE will be the first 0,18µm European technology ASIC in flight, and that it will ensure the Galileo signal generation for a very long time .