


*Report on evaluation of the integrated design flow (final)**Public*

	FP7-ICT-2013- 10 (611146)		CONTREX
	Design of embedded mixed-criticality CONTRol systems under consideration of EXtra-functional properties		
Project Duration	2013-10-01 – 2016-09-30		Type IP

	WP no.	Deliverable no.	Lead participant
	WP5	D5.3.2	Vodafone
Report on evaluation of the integrated design flow (final)			
Prepared by Luca Ceva (Vodafone) Issued by Vodafone Document Number/Rev. CONTREX/STM/R/D5.3.2/1.1 Classification CONTREX Public Submission Date 2016-09-30 Due Date 2016-09-30 Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)			

© Copyright 2016 OFFIS e.V., STMicroelectronics srl., GMV Aerospace and Defence SA, Vodafone Automotive SpA, Eurotech SPA, Intecs SPA, iXtronics GmbH, EDALab srl, Docea Power, Politecnico di Milano, Politecnico di Torino, Universidad de Cantabria, Kungliga Tekniska Högskolan, European Electronic Chips & Systems design Initiative, ST-Polito Societa' consortile a r.l., Intel Corporation SAS.

This document may be copied freely for use in the public domain. Sections of it may be copied provided that acknowledgement is given of this original work. No responsibility is assumed by CONTREX or its members for any application or design, nor for any infringements of patents or rights of others which may result from the use of this document.

History of Changes

ED.	REV.	DATE	PAGES	REASON FOR CHANGES
OFFIS	0.1	2016-09-19	33	Initial version based on D5.3.1
VOD	0.4	2016-09-29	47	Integrated contents from INTECS and OFFIS
GMV	0.5	2016-09-29	56	Integrated GMV contribution
VOD	0.6	2016-09-29	57	Integrated content from EDALAB
EUTH	0.7	2016-09-29	58	EUTH update
IX	0.8	2016-09-29	58	IX update
Intel	0.9	2016-09-30	59	Intel update
UC	1.0	2016-09-30	59	UC update
KG	1.1	2016-09-30	59	Final version for submission

Contents

1	Introduction	4
2	Evaluation of project objectives	5
	UC1a. Avionics Domain – Multi-Rotor Demonstrator	5
	UC1b. Avionics Domain – Industrial Demonstrator	8
	UC2 Automotive Domain.....	8
	UC3 Telecommunications Domain	13
3	Avionics domain	20
	Tools coverage overview	20
	Evaluation	22
4	Automotive domain.....	38
	Tools coverage overview	38
	Preliminary evaluation results	38
	Preliminary Evaluation Summary	48
5	Telecommunication domain	51
	Tools coverage overview	51
	Evaluation results	52
	Evaluation Summary	57
6	References	59

1 Introduction

This report describes the status of the industrial evaluation strategy for each of the use cases concerning the integration of the design tools.

The use-cases can be considered as “CONTREX design flow experiments”. Depending on their structure and characteristics, each use-case targets a specific path in the design flow and focuses on a specific tool subset. As an example, the avionics use case focuses on the high-level specifications (based on UML models), while the two other use-cases more specifically focus on implementation and optimization steps. The use cases should provide quantitative metrics to verify the compliance level of the CONTREX methods and tools with regard to the industrial needs and use case goals and objectives.

2 Evaluation of project objectives

UC1a. Avionics Domain – Multi-Rotor Demonstrator

The use case 1a multi-rotor demonstrator has been developed for the CONTREX project to have a fully and freely accessible system for all partners evaluating their methodologies. Description of all demonstrator parts is available in the following public CONTREX deliverables:

- **D1.2.1 [5]: Section 2.1** with an introduction and basics about the multi-rotor system
- **D2.3.2 [6]: Section 6.1.8** with an overview of the virtual demonstrator prototype
- **D4.1.2 [7]: Chapter 3** with the application description
- **D4.2.2 [8]: Section 2.2** with the demonstrator's platform description

As mentioned in the referenced deliverables, the multi-rotor demonstrator uses the Xilinx Zynq 7020 MPSoC on its avionics. One of CONTREX project's main goals is:

“Reduce the power consumption by at least 20%”

The following evaluation compares the power consumption of the CONTREX solution, using the Xilinx Zynq MPSoC as a mixed-criticality single package system, with state-of-the-art techniques, using segregated processing elements for tasks of different criticality levels.

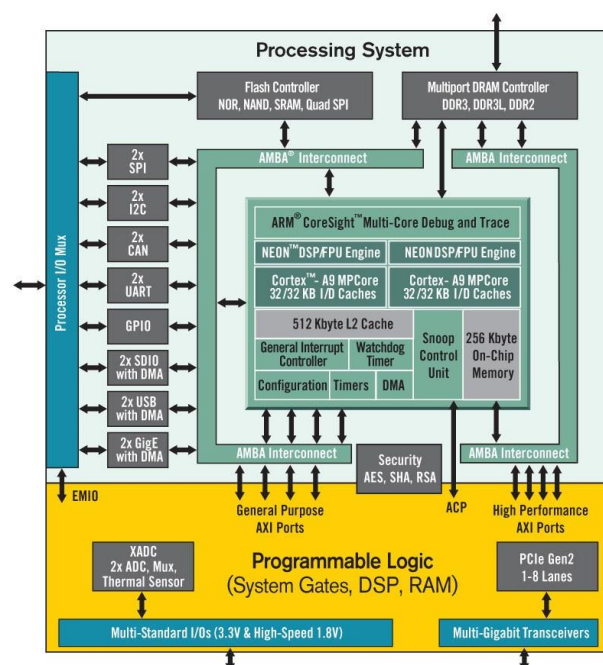


Figure 2-1 Overview of the structure of the Xilinx ZYNQ family [9]

Figure 2-1 gives an overview of Xilinx Zynq architecture. Its package contains in the version 7020 two major parts:

Report on evaluation of the integrated design flow (final)

1. **Processing System:** ARM Cortex-A9 Dual Core up to 866MHz
2. **Programmable Logic:** Artix-7 FPGA with 85k logic cells

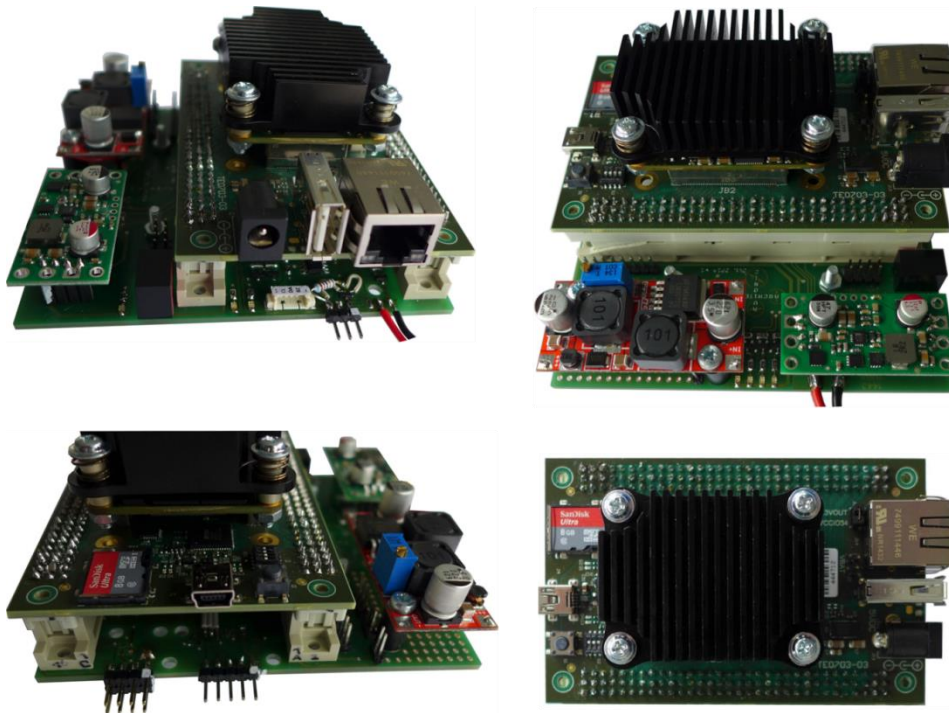


Figure 2-2 Developed avionics based on a Xilinx ZYNQ MPSoC

Figure 2-2 shows the developed avionics of the mixed-criticality multi-rotor system. The stack consists three parts. The distribution board (bottom in stack) with power supplies (small PCBs), the carrier board (center in stack) from Trenz Electronic [10] with the small Xilinx Zynq 7020 industry board on top (also from Trenz Electronic, completely covered by the heat sink). This solution is very flexible, since many different industry boards are available to fit different use cases in future and for this evaluation.

D4.1.2 [7] describes the mixed-criticality applications executed by the Zynq avionics. The ARM cores computes a mission-critical video processing task, while two MicroBlaze soft processors in the programmable logic compute the safety-critical flight algorithms. This describes the CONTREX setup of the avionics. The CONTREX solution uses in our case a Zynq 7020 with the industrial speed grade variant “-2I”. With speed grade variant “-2LI” Xilinx offers a low power variant of the Zynq 7020 that has 40% less static power consumption and with a lower voltage for the programmable logic 10% less dynamic power consumption [11]. Since this variant of the Zynq is not available on the standard industry boards at Trenz Eletronic, a custom-made product is too expensive. In that way we will subtract only 15% of our Zynq “-2I” power consumption, since the dynamic power consumption counts much more into the overall power consumption than the static ones.

Like mentioned the Zynq industry board is easy exchangeable. For the state-of-the-art solution we exchanged the Zynq industrial board with a FPGA-only board, which has a comparable Artix-7 FPGA with 100k logic cells (part number “XC7A100T-2CSG324C”, see Figure 2-3). All parts of the Zynq programmable logic were transferred to this FPGA. Therefore, it contains mainly the two MicroBlazes to compute the flight algorithms.

Report on evaluation of the integrated design flow (final)

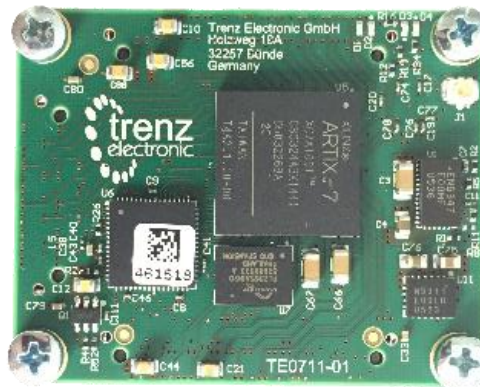


Figure 2-3 Artix-7 board of Trenz Electronic with 100k logic cells

Next to this board, we added another board that contains a dedicated ARM Cortex-A9 Dual Core with nearly the same features (792MHz frequency, 1GB RAM, USB and Network support) to the avionics. It is a Mars Board of Embest [12], which is shown in Figure 2-4. This board is also powered over the avionics' 5V power supply. As well, the mission-critical software of the Zynq system was ported with the same configuration to this board.

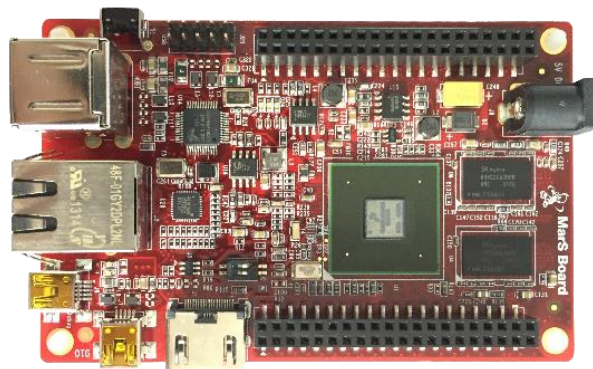


Figure 2-4 ARM Cortex-A9 Dual Core Mars Board of Embest

All following measurements are based on the input power consumption of the whole avionics. The mission-critical part, in both cases the ARM Cortex-A9 cores use further external peripheral devices via an USB connector. The power consumption of the four devices, WiFi-stick, camera, gimbal controller and USB hub, is not included in these power consumption values, since the payload system is a replaceable part for different use case tasks. Further, the CONTREX methodologies consider only the processing parts of the multi-rotor system. The power consumption of the payload system is shown in Table 1.

Table 1 Power consumption of payload parts

Payload Part	Power Consumption in mW @ 16V
USB-Hub	320mW (20mA)
WiFi-Stick	1,600mW (100mA)
Camera	640mW (40mA)
Gimbal-Controller	640mW (40mA)
Overall	32,000mW (200mA)

The overall power consumption value of the payload system is already subtracted in the following tables. Since the video processing task always calculates as much frames as possible per second both ARM Cortex-A9 cores' utilization is 100%.

*Report on evaluation of the integrated design flow (final)***Table 2 Power consumption of CONTREX solution**

CONTREX sol.	Power Consumption in mW @ 16V
Zynq 7020 avionics	3,360mW (210mA)
-15% for “-2LI”	-504mW (-31.5mA)
Overall	2,856mW (178.5mA)

Table 3 Power consumption of state-of-the-art solution

State-of-the-art	Power Consumption in mW @ 16V
Artix-7	1,280mW (80mA)
Mars Board	2,400mW (150mA)
Overall	3,680mW (230mA)

By comparing the overall values of Table 2 and Table 3 it is obvious that the CONTREX solution of the avionics reduced the power consumption more than 20% regarding the state-of-the-art solution. The exact value **22.4% reduced power consumption**, under the assumption that the “-2LI” variant of the Zynq has less 15% overall power consumption as our used “-2I” variant.

This satisfies fully with the mentioned main goal of CONTREX and the multi-rotor demonstrator showed that the used and developed methodologies in the project are promising in its case.

UC1b. Avionics Domain – Industrial Demonstrator

The UC1b execution platform was also based Xilinx Zynq MPSoC (see Figure 2-1).

The general project objective of reducing the power consumption by at least 20% was also successfully achieved by Use Case 1b. Concrete figures and more details about this, as well as evaluation results about other domain- or use case-specific objectives regarding Use Case 1b are addressed in section 3.2.3.

UC2 Automotive Domain

2.3.1 Project goals

“Increase the number of functionality by a factor of ten.”

The following new functionality have been implemented from scratch and are either already deployed on test set of vehicles, or will be deployed in October/November 2016.

- Self-calibration to determine orientation of the device w.r.t. the vehicle
- Detection of the direction (forward/backward) during parkin manoeuvre

Report on evaluation of the integrated design flow (final)

- High-precision data collection for severe crash reconstruction
- Low-energy events and acts of vandalism detection
- System wake-up (degraded low-energy event detection) in ultra-low power mode
- Cloud-based crash monitoring service

The introduction of the power monitoring and management infrastructure and the availability of new sensors on the node platform, will also enable new functionality, which are currently under investigation, namely:

- Improved and faster self-calibration algorithms exploiting gyroscopic data
- More sophisticated crash classification algorithms exploiting high frequency data
- Low-energy events classification performed on the node, to reduce server-side loading and communication costs.

In general, the shorter design turnaround time achieved through simulation and fast reconfiguration of the power management policies will allow for a number of new algorithms to be integrated on the node.

“Reduce the power consumption by at least 20%”

According to the results outlined in D3.3.3, the methodology, the simulation tools and execution platforms, both hardware and firmware, allowed a reduction of the power consumption of more than 70%, much more than the required 20%.

2.3.2 Use-case specific goals

The specific goals for the automotive use-case, as defined in D1.3.2, have been met and a short summary of quantitative results are summarized in the following tables, one for each of the four scenarios identified for the use-case-

Device Installation		
Expected results	Means of validation	Results
Average installation time reduction	Comparison of the installation time of the new device with average figures extracted from past installation reports.	The typical installation time of devices without the self-calibration function was on average 80 minutes. Self-calibration allows installing the device in a wider range of (more easily reachable) positions in the vehicle, without any need to find the correct orientation.

Report on evaluation of the integrated design flow (final)

		Furthermore, the “static calibration” procedure can now be skipped, further reducing the installation time. First estimates on a limited number (1000) of vehicles show an average installation time below 30 minutes.
Call rate reduction	Comparison against historical data. Note that collecting significant call rate data might require longer times than those allowed by the project duration, and thus the quantitative evaluation will only be considered preliminary.	Errors in the installation procedure led to a call-rate of approximately (1,5%, 6000 vehicles/year). Extrapolating from limited installation base that is now in field, the call rate shall drop to 0.1%, i.e. 400 vehicles/year. After a further tuning we expect to reach the 0.02% or call rate i.e. 80 vehicles/year.
False positive crash reports reduction	Comparison against historical data.	Thanks to the improved quality of crash data produced by the device node and the more sophisticated classification flows that can be implemented on the sever, the false positive fraction has dropped from 85% to less than 15%. The combination of the algorithm improvements both server and device side led to manage autonomously (i.e. without human intervention) 85% of the crash events received (false positive and true positive). It is expected that this figure will grow above 90% in the next few months.

Crash Management		
Expected results	Means of validation	Results
Crash-to-call latency reduction	Comparison with historical quality-of-service data.	The latency between crash detection, crash notification and the start of assistance management by operators in the control room (as in normal conditions, i.e. with the mobile network available) was on average 120s.

Report on evaluation of the integrated design flow (final)

		<p>Thanks to the improved crash management and filtering of false positives, the queue of events to be managed has been reduced and the latency to start the managing by the operators is now around 60 seconds.</p> <p>Thanks to the improved communication stack implemented on top of Kura, the estimate of the new average latency is approximately 45s.</p>
Cost reduction	Evaluation of the expected cost reduction as a consequence of false positive crashes and the improved sensing quality of the sensor nodes.	The new enhancements, studied and developed in the scope of CONTREX project allowed VA to manage the steady increase of customers (and the according increase of alerts) without increasing the staff of the SOC (Security Operation Center). In particular, the doubling of the customers has been managed with the same SOC staff.

Key-off services		
Expected results	Expected results	Expected results
Key-off services	New business opportunities	Thanks to the availability of the new set of key-off services, new insurance companies have been contacted. One of these new companies (one of the biggest companies in Italy) has been engaged in an official negotiation, which is expected to be finalized in Q4 2016 or Q1 2017. The potential number of new customers is approximately half a million.

B2B Scenario		
Expected results	Expected results	Expected results

Report on evaluation of the integrated design flow (final)

Scalability	Provide evidence of the scalability opportunities offered by the cloud-based solution.	The scalability of the device-to-cloud approach depends primarily on the capability of the cloud platform, that is in charge of collecting data and managing all the vehicles equipped with the ECU. The cloud platform is based on Amazon EC2, that provides an “Auto Scaling” functionality, enabling to follow the demand curve of the applications closely and dynamically.
Cost reduction	Comparison of the expected cost growth of the current ad-hoc solution with the cloud-based approach.	<p>Device-to-cloud solutions introduces five main sources of costs reduction, when compared to ad-hoc solutions:</p> <ul style="list-style-type: none"> • Optimization of hardware usage and of power costs for enterprise server. It also provides resilience without redundancy of own hardware. • Reduction of personnel costs, with a specific simplification of IT administration activities. • Zero capital costs, because it doesn't require investments, just a “pay per use” contract. • It provides open APIs, that guarantees flexibility, reduction of development costs and simple integration

Report on evaluation of the integrated design flow (final)

		with existing enterprise application.
--	--	--

UC3 Telecommunications Domain

2.4.1 Relationship of use case goals to project goals

As stated in the Description of Work (B1.1.2), “... the main goal of the project is to combine platform independent models represented in domain specific modelling languages and formalisms; management and abstraction of multi-core hardware platforms; management and abstraction of communication resources; with management and control of extra-functional properties power and temperature.”

The components of this stated goal of the project represent exactly the specific goal of the telecom use case, to create a **suitable development environment for its next generation of applications and products**. This development environment consists of both processes (methodologies) and tools. The goal was pursued with the principal cooperation of four main partners in the use case:

- **“Platform independent models represented in domain specific modelling languages and formalisms”.** A year-long collaboration with partner KTH on the use of the ForSyDe domain specific language produced successful modelling and abstraction of the core elements of the telecom application, when was then in a subsequent phase subjected to experimentation in Design Space Exploration.
- **“Management and abstraction of multi-core hardware platforms.”** Partner OFFIS produced an accurate simulation environment of the new modern multi-core Zynq platform intended for the next generation of applications in the Intecs telecom segment. Working together, Intecs and OFFIS were able to successfully port the telecom application to run on the simulated platform, with execution behaviour that was faithful to the original.
- **“Management and abstraction of communication resources.”** Partner EDALab provided the necessary tools and methodology to abstract and integrate a legacy network communication device into the overall simulated environment in order to ensure an accurate simulation of the entire component in its proper execution context. In addition, it provided an essential learning experience that was necessary as a “bridge” between the real execution environment and the simulation environment – that is, the integration of proprietary and legacy IP that is often described and implemented in non-compatible ways to the eventual simulation environment. Without this essential know-how, most of the benefits of the overall simulation environment would not be realizable.
- **“Management and control of extra-functional properties power and temperature.”** Partner Docea / Intel provided the needed power and temperature measurement tools to generate the necessary feedback and guidance on design and execution loops for the telecom applications under consideration. The key step forward beyond the current state of the art in the development environment was the provision of thermal and power measurement capabilities *both* in the simulated environment and in the real

Report on evaluation of the integrated design flow (final)

environment, which involved issues both of functionalities (e.g. extracting the measurements from the simulated platform) and in accuracy (that is, sufficient fidelity of the simulated measurements to the real ones).

Another specific stated goal of the project was “support for scaling up of mixed critical applications per SoC by a factor > 10 , while reducing power consumption by at least 20%.” This goal constituted an indirect one expected to be attained through the main goal outlined above – that is, the establishment of a new development environment for the next generation of telecom applications. The application in the use case was intended to provide the base case for the introduction of the new processes and platform/tools, rather than to be extended itself with new functionalities (since it was not destined to extend its product life beyond the project). However, the previous platforms and processes (single core platform, no pre-construction modelling, no simulation, only end of line power and thermal measurement) were clearly entirely inadequate for the new generation of telecom products which are intended to support higher levels of throughput of both business grade and critical (e.g. emergency) communication and consumer-grade (e.g. voice) communication while keeping power and thermal characteristics in check. In any case, a direct side-by-side comparison would not have been possible, given the legacy structure of the demonstrator application, and given the naturally higher power requirements of the far more powerful execution platform introduced in CONTREX (Zynq versus the legacy PowerPC). But the preliminary results of the simulation modelling and power/thermal measurement have indicated that, with the modelling formalisms provided by CONTREX (in our case, ForSyDe) for creating the new structures of the applications, the new platform (Zynq), together with its co-simulated environment (provided by OFFIS and EDALab), and measurement tools provided by Docea/Intel, it will be possible to implement much more powerful functionality on the modern platforms while controlling power/thermal “costs”. Those costs will *not* be less than the previous costs in absolute terms, since the more powerful platform naturally consumes more power; rather, we are seeing a relative advantage in the costs – in layman’s terms, “more bang for the buck.” While this is a qualitative result in some respects, that is to be expected when the primary project objective was an overhaul of the entire development environment. But it remains a very concrete result, with repercussions for efficiency not only in the products themselves but for efficiency and productivity in the development environment.

2.4.2 Individual evaluation of use-case specific goals

In the following table, more specific information is provided on the validation of the original goals of the use case that was carried out. Some of the original goals were considered to be optional “nice to have” goals and were not further considered as later activities coalesced around the principal goals of providing the new development environment in all of its components.

Conversely, however, an important goal – TLC-UC08 – considered only *optional* at the beginning, was later judged to be essential for the true achievement of the new development environment and cross-verification of the real and simulated platforms. Thus, much of the effort originally under consideration for other optional goals was utilized to ensure that TLC-UC08 was carried through and then augmented with the new cross-verification activities.

The originally envisioned means of validation are provided in clear text, and below the eventual means of validation are provided in italics.

Report on evaluation of the integrated design flow (final)

Requirement	Priority	Means of validation
TLC-UC01 Technology providers of CONTREX shall provide to Intecs lab the Zynq environment (tool chain, kernel,...)	Mandatory	<p>Intecs lab will verify the possibility to generate the Kernel and the Root File System for the Zynq platform, using the environment provided by technology partners.</p> <p>This was carried out in full in particular by partners OFFIS and EDALab, working together with Intecs to ensure that the Zynq platform was in full working order with the baseline software operational.</p>
TLC-UC02 The Telecom demonstrator currently running on a MPC880, shall be made runnable on Open Virtual Platform equipped with a reliable model of a Zynq board (the Zynq model must be provided by technological partners)	Mandatory	<p>Several tests will be performed on the Open Virtual Platform in order to verify the correct modelling of the Zynq platform, e.g., verification of the correct implementation of:</p> <ul style="list-style-type: none"> - Get/set from a MIB browser running on an external PC - web-server connection from external craft terminal - CPU load monitoring <p>This will be a full implementation of the application on the Zynq model. The aim is to increase the Intecs Telecom know-how in the usage and exploitation on the Open Virtual Platform, in order to use it in upcoming product development cycles. The objective is to reduce time-to-market by having the possibility to test and dimension the software before final deployment on hardware.</p> <p>In collaboration with OFFIS in particular, a full port of the telecom application was made to the OVP simulated Zynq platform and its</p>

Report on evaluation of the integrated design flow (final)

		correct functioning was validated. Different loads and CPU speeds were experimented with in order to acquire experience and understanding of the use of the environment.
TLC-UC03 Technology provider of CONTREX shall provide to Intecs lab all the peripherals models (e.g. Ethernet interface) to be integrated with the Zynq model on the Open Virtual Platform environment	Mandatory	<p>The Open Virtual Platform setup of the board is executed.</p> <p>The primary partner involved in the validation of this requirement was EDALab, which provided a full integration of the legacy Ethernet interface of the telecom demonstrator application into the OVP simulation environment using its HIFSuite tool and associated methodology.</p>
TLC-UC07 Power and Thermal analysis tool-set shall be provided and integrated in the Open Virtual Platform by technological partners	Mandatory	<p>Power and thermal computation will be performed exploiting the tool-set provided by the technological partners. The aim is to increase the Intecs Telecom know-how in the usage and exploitation on such tool-set, in order to be able to use it in next product developments, having the possibility to estimate the power consumption before to integrate the software development with the hardware one.</p> <p>The primary partner involved in the validation of this requirement was Docea / Intel, who provided support for the use of the Aceplorer tool not only in the real environment but also in the simulated environment. The bridge to the simulated environment was provided by Docea / Intel jointly with OFFIS, through the generation of power traces from the simulated model. Then both Intecs and OFFIS made experiments both on the real implementation (see below) and on</p>

Report on evaluation of the integrated design flow (final)

		<p>the simulated environment implementation to verify whether the extra-functional properties were being properly modelled. One technique used in order to have a more systematic, controlled observation of the thermal properties in particular, was to create both simulated and real different processor speeds, in order to observe the expected differences in temperatures over the range from startup to full regime. The results were extremely coherent across the real and simulated platforms, providing a remarkable validation of the attempt to provide a consistent environment with simulation and real development facilities.</p>
<p>TLC-UC08 The Telecom demonstrator, currently running on a MPC880, will be made runnable on a Zynq real board</p>	Optional	<p>The same tests performed on the Virtual Platform, both in terms of functional requirements (registers mapping) that of extra-functional requirements (power consumption) will be performed on a real Zynq platform to verify the:</p> <ul style="list-style-type: none"> - precision of the modeling of the Zynq on the Open Virtual Platform - the accurate functioning of the power tool-set and its integration in the simulation environment <p>Note: This requirement is Optional because there is already a current baseline implementation of the demonstrator running on MPC880 hardware, and thus the porting is principally of interest for cross-verification of the results obtained using the virtual platform and the power/thermal tool-set.</p>

Report on evaluation of the integrated design flow (final)

		<p>Rather than re-engineering the baseline demonstrator <i>per se</i>, the primary strategic aim is to use it to provide a basis to extend the company know-how in the CONTREX innovation domains (virtual platform, power estimation) in order to integrate them in future product development processes.</p> <p>As noted both in the main text and in the individual requirement descriptions above, this requirement was originally thought to be only “nice to have”. But it turned out to be essential in order to validate the entire concept of the parallel simulation and “real” development environments. Without it, there would have been no credible validation of the overall approach that the partners were implementing in the telecom use case. Given that this was one of the principal use cases in which the methodological aspects of the CONTREX goals were being experimented, it was decided by Intecs and its partners that the real implementation on the new Zynq platform had to be carried out, and the necessary resources were marshalled to do so. The implementation went through without undue complications, and provided at the same time the necessary familiarity with the new kind of SoC envisioned for next generation telecom application development.</p> <p>As noted in the other individual requirements descriptions above, this real implementation then provided the baseline for nearly all other validation activities, as the real behavior of the application on the board was compared in various ways (functional results, timing,</p>
--	--	---

Report on evaluation of the integrated design flow (final)

		power, thermal) with the simulated behavior in the OVP environment.
--	--	---

3 Avionics domain

This section provides a description about the status and preliminary results of the validation of the integrated flow applied to the development CONTREX Use Case 1.

Tools coverage overview

An integrated environment for system modelling, model transformation and code generation has been developed for the development of the Use Case 1 (corresponding to the avionics domain) in CONTREX. Based on the Eclipse framework, it consists of the Papyrus modelling tool, the MARTE profile, the Acceleo model-to-text transformation engine and the CONTREX Eclipse plug-in (CONTREP). CONTREP tool includes extra profiles for the modelling activities (so to make UML/MARTE models that fulfil the CONTREX modelling methodology), as well as the model validation, code generation and model transformation features.

Additionally, CONTREP includes the necessary functionality to serve *“as a cockpit from where the user can drive most of the design flow tasks (different types of code generation, compilation, simulation, launching DSE, ...), which enables a reduction (unless eliminating) of the interaction with the underlying tools.”*. This way, system analysis, simulation and DSE can be launched directly from the Eclipse front-end (by means of a specific menu added by CONTREP to the Eclipse GUI), simplifying the design flow from the user perspective.

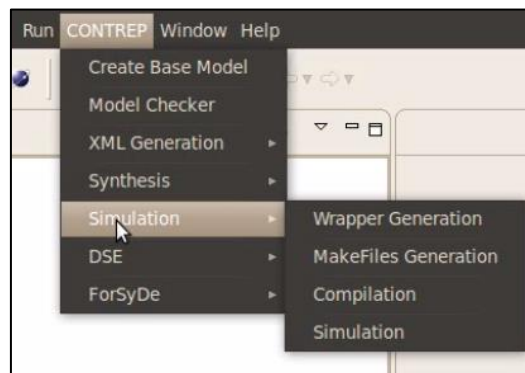


Figure 5. CONTREX menu in the Eclipse front-end.

The Use Case 1 tool-chain includes also the following tools:

- A-DSE: enables analytical DSE for worst case time analysis.
- eSSYN: produces the SW infrastructure of the application.
- VIPPE: generates the performance executable model that enables native simulation of the system, based on the SW infrastructure, the platform information and the corresponding DSE parameters. The resulting executable model is configurable; i.e., it can be launched for different HW/SW mappings and system configurations without re-generation.

Report on evaluation of the integrated design flow (final)

- MOST: enables the automatic DSE of the system by launching different executions of the performance model (under different system configurations) and selecting the best solutions based on discrete optimization technology.
- Intel® Docea™ Thermal Profiler: enables thermal simulation from a physical description of a system. Stimuli for the simulation are power traces, that tools like VIPPE or VP featured with power estimation capabilities can generate.

The execution of these tools, with the exception of OVP and Thermal Profiler, is managed by the CONTREP tool. The tools all take as input the intermediate files/models generated by CONTREP from the CONTREX UML/MARTE model. Both, OVP and Thermal Profiler have been launched and handled separately.

In order to support the evaluation of previous tools, Imperas OVP has been also used. This tool enables an OVP-based simulation of the system thus enabling assesment and validation, in terms of simulation performance and accuracy, of the higher level native simulation technology implemented tool (VIPPE). The automatically generated code from eSSYN speed-up and enhances the coherence of the evaluation.

In the CONTREX avionis use case, the dynamic thermal profiler analysis has been used for post-DSE validation of the solutions. For that, CONTREP enables to generate the VIPPE command producing the power traces. Then, the user can explore interactively the solutions of interest filtered by MOST-VIPPE, by reading the power traces of each assessed solution.

After the sucessfl evaluation of this integration work, and given the experience with Thermal Profiler further possibilities are devised, specifically, the integration within the automated DSE flow. The Thermal Profiler Application Programming Interface (API), by allowing the launching of the application, and the retrieve of the thermal analysis results, should allow a fully automated execution and integration in CONTREP.

The following figure shows how the CONTREX tools have been integrated into the pre-existing design flow.

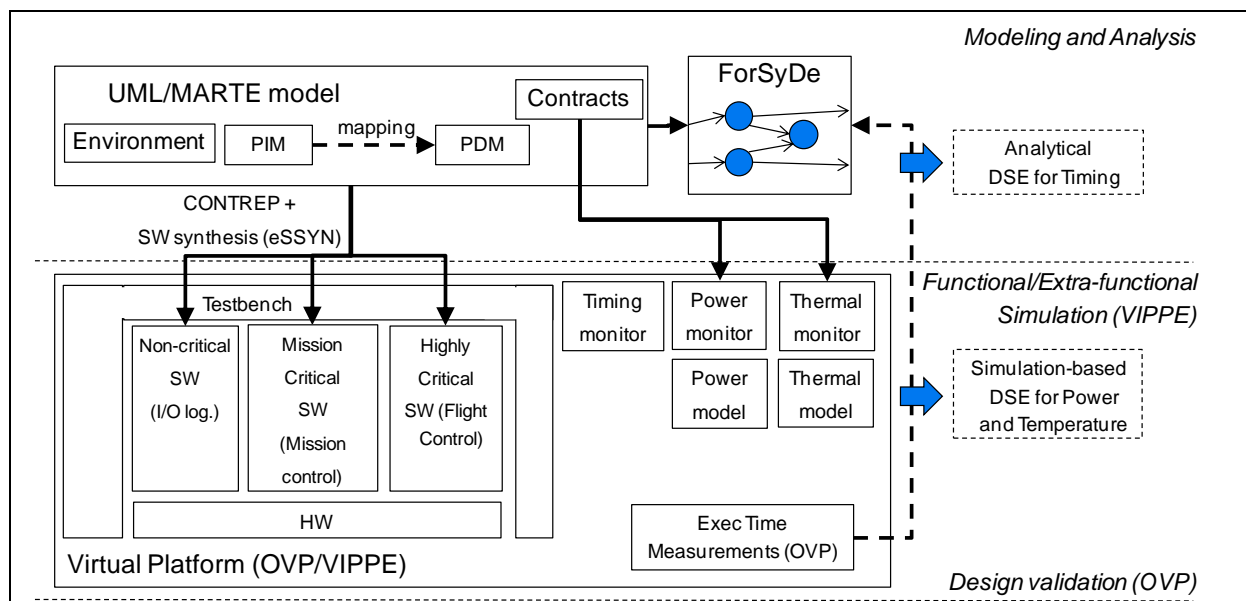


Figure 6. Tool roles into the UC1 integrated design flow.

Evaluation

This section provides a brief summary of the general objectives pursued in CONTREX regarding the avionics domain, the evaluation process followed for the CONTREX integrated flow and a summary of the evaluation results.

3.2.1 Objectives

GMV aims to tailor its Flight Control Computer (FCC) system developed for the ATLANTE RPA to future RPA platforms, particularly, to light RPAs (<150 kg) for new markets and countries, enabling to maintain a competitive supply base in Europe.

Taking into account that size, weight and power (SWaP) constraints are a key factor for light RPA equipment, the adaption of the FCC SW to diverse commercial all-purpose platforms and low-cost avionics sensors will be required.

However, as current avionics flow only focuses on custom platforms for systems under construction, it is required a more flexible approach that enables an early assessment of the system performance as well as the efficient (automatic) exploration of a wider design space. This will permit finding optimal platforms and configurations that minimize cost, size, weight and power consumption without compromising safety and overall performance.

With the aim of supporting these new required capabilities, it is expected that CONTREX improves the current avionics development flow by introducing extra stages for system modelling, model-based analysis, simulation and DSE during the design phase. The results gathered during this enhanced design phase would enable the designer to make informed architectural decisions (such as platform selection and HW-SW mapping) based on reliable performance figures, so the number of design errors (and thus, the need for design re-work at late stages) can be reduced. Moreover, in the case of mixed-criticality systems, assessing system performance in multi-core general-purpose platforms at early stages might lead to significant time savings and cost reductions.

The main result expected from CONTREX is the integration of its design methodology into the overall avionics development flow, as shown by figure below.

Report on evaluation of the integrated design flow (final)

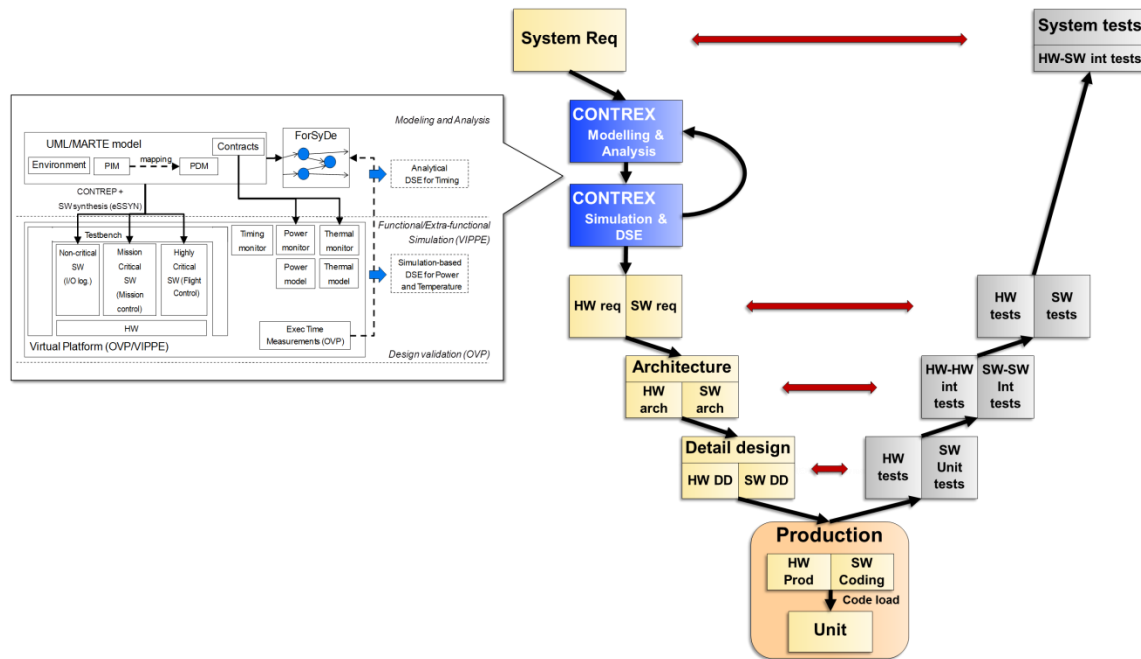


Figure 7. CONTREX integrated design flow applied to the avionics development flow.

3.2.2 Evaluation process

In order to demonstrate the benefits of CONTREX, a subset of the FCC (Flight Control Computer) software developed by GMV for a medium sized Remotely Piloted Aircraft has been reused. Two key principles were followed for the selection of the specific SW subset to be included in the CONTREX demonstrator:

- The resulting application should be complex enough to permit a useful exploitation of the CONTREX tools and methodologies. Mixed-criticality and extra-functional properties must therefore be relevant for the demonstrator, as they are main concerns in CONTREX.
- The resulting application should be simple enough to be manageable in the context of the CONTREX project. The main objectives of the Use Case 1b in CONTREX are to exercise and evaluate the new methodology and tools and assess the experience and results; using a too-complex demonstrator might hinder rather than contribute to achieve these objectives.

Taking into account these two main principles, the sensor I/O module was considered the most suitable subset of the FCC's software for being used as the CONTREX avionics demonstrator. It was simple enough to put the focus on the evaluation of the tools and not devoting too much effort in the adaption of the legacy code to the new platforms and design methodology, but at the same time it included several mixed-criticality and extra-functional property features that made it fully compatible with CONTREX's main concerns.

The FCC's sensor I/O module sends the corresponding commands to the I/O sensor devices and collects (in real-time, according to the data rates supported by the sensors) data from them:

Report on evaluation of the integrated design flow (final)

- On the one hand, the SW components of the sensors I/O module are assigned different criticalities according to the sensor data they process: the failure of a given component might make the RPA fall on populated areas or crash with another aircraft; the failure of others may not have catastrophic consequences but however impede the accomplishment of the mission. Thus, **timing and reliability requirements are critical in this system.**
- On the other hand, the TE0720-01-2IF industry board based on the Xilinx Z-7020 MPSoC has been utilized in CONTREX as the execution platform for the sensor I/O module. In the real system, this board would be expected to be embedded in the FCC chassis. Therefore, **heat dissipation, power consumption and temperature are also relevant variables to be observed.**

Once the avionics demonstrator was selected, the process illustrated by the next figure has been followed during the project in order to exercise and evaluate the integrated flow developed in CONTREX for the avionics domain:

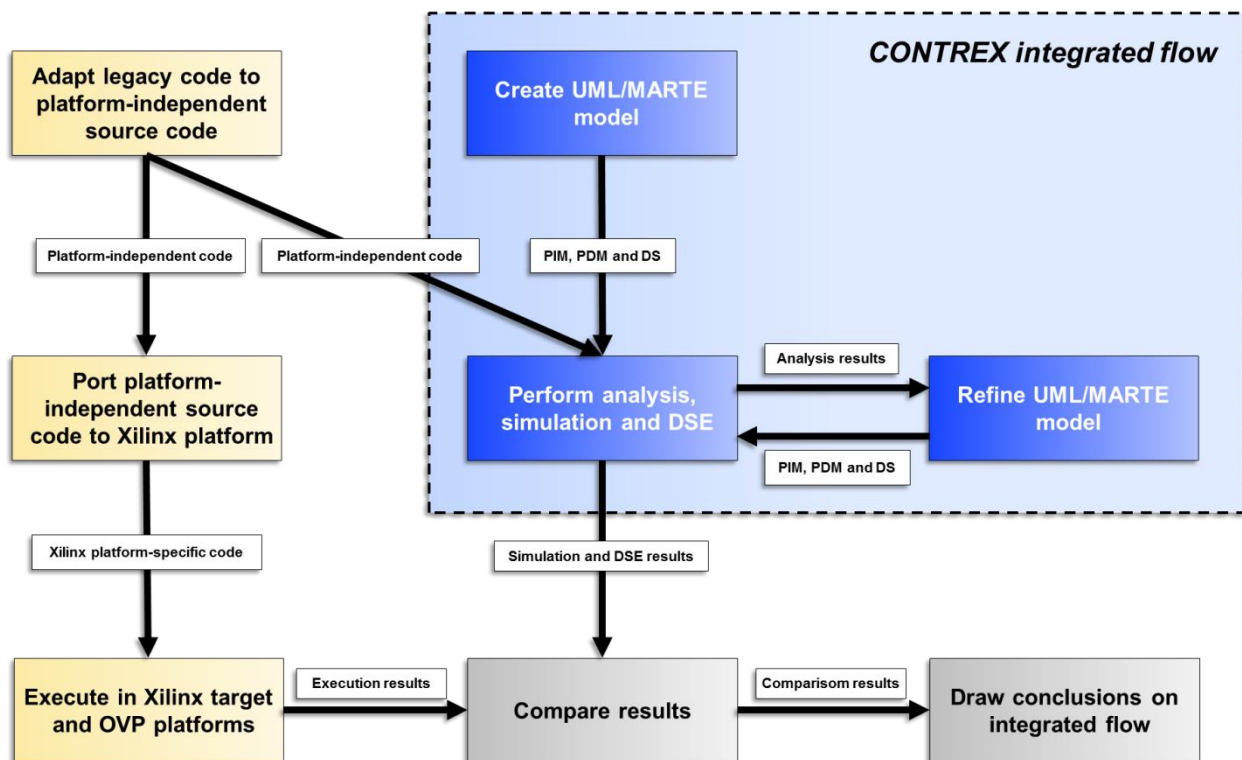


Figure 8. Evaluation process of CONTREX integrated design flow

A brief description of the tasks shown in previous figure is provided here below:

- **Adapt legacy code to platform-independent source code:** In order to enable the execution of high-level system simulations and the exploration of different design alternatives (or different system configurations; mainly, different SW/HW mappings), the original sensor I/O module source code needed to be adapted to platform independent source code (only the application's pure functional code was to be reused from the original source code).

Report on evaluation of the integrated design flow (final)

- **Create UML/MARTE model:** Produce a UML/MARTE model compliant with the modelling methodology developed for CONTREX. This model includes the PIM (Platform Independent Model; i.e., the high-level description of the sensor I/O module application), the PDM (Platform Description Model; i.e., the description of the platform components to be considered for the final system, in this case based on the Xilinx platform) and the DS (Design Space; i.e., a set of platform specific models captured in the UML/MARTE model as the different design alternatives or system configurations).
- **Perform analysis, simulation and DSE:** Exercise the analysis, simulation and design space exploration of the system based on the UML/MARTE model and the corresponding platform-independent source code. This step will output the corresponding performance metrics (on time, power and temperature) and DSE reports about the optimal system configurations.
- **Refine UML/MARTE model:** Based on the analysis results, the model (or even the source code) may require some modifications before continuing the design loop (simulation and DSE).
- **Port platform-independent source code to Xilinx platform:** The application's pure functional code is added the needed source code to be successfully executed onto the Xilinx platform, which was selected in CONTREX as the target execution platform for the sensor I/O module.
- **Execute in Xilinx target and OVP platforms:** The platform-specific code resulting from previous step is executed on the real target and the OVP platform under the same system conditions than in the high-level simulations. Performance result from this execution (on time, power and temperature) will be gathered.
- **Compare results:** Performance metrics from real target, OVP and high-level simulations will be compared in order to assess the accuracy of high-level simulation results.
- **Draw conclusions on the CONTREX integrated flow:** Based on the accuracy of results observed from high-level simulation, as well as other performance metrics (such as native simulation time against lower-level OVP simulation time or the time devoted to produce the corresponding models), conclusions on the potential benefits expected from the CONTREX integrated flow (in terms of utility, reliability and effectiveness) will be drawn. This way the efficiency gain of using the CONTREX-enhanced development flow instead of previous avionics flow can be assessed.

The utility, reliability and effectiveness of CONTREX approach as well as the benefits it was expected to bring to the current avionics development flow have been assessed in WP5 according to the evaluation strategy outlined in D1.3.2. This strategy consisted mainly of the:

1. Assessment of the accuracy of the estimations and results obtained using the CONTREX methodology.
2. Assessment of the efficiency gain when using the CONTREX-enhanced development flow instead of previous flow, considering the objectives enunciated above.

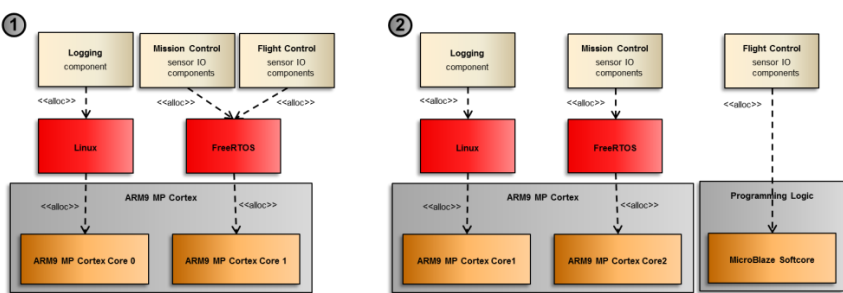
Report on evaluation of the integrated design flow (final)

The first step was mainly aimed at verifying the reliability of the estimations produced by the high-level native simulation technology integrated into the Use Case 1's CONTREX flow. Once this reliability has been verified, the efficiency gain obtained with the CONTREX approach could be assessed. This was done from two different perspectives:

- *Resources savings.* Comparison between usual estimations on resource needs and those obtained from the application of the CONTREX approach.
- *Development time savings.* Comparison between times spent in exploring the design space automatically and manually, and comparison of total time spent in the development of the Use Case 1 using CONTREX approach against the total development time that would be spent in case a problem was detected at the HW-SW integration phase.

It is also worth mentioning that, in order to guarantee the suitability of the CONTREX methodology for the avionics domain, a set of requirements, whose fulfilment has been also assessed in WP5, were defined in D1.1.2.

The following tables briefly report about the final status and results of the tasks outlined in Figure 8.

Task	Adapt legacy code to platform-independent source code
Status	Completed.
Summary of results	<p>Successful generation of a platform-independent source code based on the original legacy code. This source code was developed in a way that it could be used without any further modification under any SW/HW mapping (originally, two alternatives were considered, as illustrated by figure below).</p>  <p>This way, exactly the same functional source code was used for:</p> <ul style="list-style-type: none"> - Native pure functional testing (Windows). - Native emulation using CONTREX tool-chain (Linux).

Report on evaluation of the integrated design flow (final)

	<ul style="list-style-type: none"> - High-level native simulation of design alternative #1 using CONTREX tool-chain. - High-level native simulation of design alternative #2 using CONTREX tool-chain. - Real target execution of design alternative #1. - Real target execution of design alternative #2. <p>For each case, a different platform-specific source code, adapted to the particularities of the corresponding execution platform was used. In the case of the <i>pure functional testing</i> and <i>real target execution</i> versions, this platform execution code was developed by GMV. In the case of <i>native emulation</i> and <i>high-level native simulation</i> versions, the platform specific source code was automatically generated by the CONTREX tool-chain (in particular by <i>CONTREP</i> and <i>mSSYN</i> tools developed by UC).</p>
--	---

Task	Create UML/MARTE model
Status	Completed.
Summary of results	A UML/MARTE model, including the PIM, PDM and the corresponding was successfully developed for being used as the base for the analysis, simulation and DSE loop. This model was developed according to the modelling methodology documented on D2.2.2 that, as reported in D5.1.2, adequately achieved the requirements derived from the avionics domain (specified in D1.1.2). In D5.1.2 and D5.2.2 several excerpts of this model can be found.

Task	Perform analysis, simulation and DSE
Status	Completed.
Summary of results	System simulations and DSE were carried out for the three SW/HW mappings identified above. These simulations outputted the

Report on evaluation of the integrated design flow (final)

	corresponding estimations that were taken into account by the DSE loop in order to enable the user selecting the best options.
--	--

Task	Refine UML/MARTE model
Status	Completed.
Summary of results	As said above, some improvements were required in the model. For instance, the periods of some tasks had to be adjusted after the results thrown by preliminary system analyses and simulations.

Task	Port platform-independent source code to Xilinx platform
Status	Completed.
Summary of results	This consisted on developing the platform-specific source code to be added to the platform-independent code in order to execute the two design alternatives shown above in the target platform.

Task	Execute in Xilinx target and OVP platforms
Status	Completed.
Summary of results	<p>Results from executions in target and OVP platform were collected and analysed. The common platform-independent source code guaranteed that exactly the same functional code was executed in the high-level native simulator and the real target platform.</p> <p>In the case of the OVP, the amount of effort required to develop a platform model that emulate the ones being used for the two design alternatives in the high-level simulator and the Xilinx target platform was out of the scope of the project. In any case, the Imperas OVP was not actually part of the tool-chain associated to the CONTREX Use Case 1b flow. Its usage was devoted instead to compare the results provided by VIPPE tool with the results provided by the OVP</p>

Report on evaluation of the integrated design flow (final)

	for the same application and platform and under the same execution conditions. Thus, a simpler synthetic example was developed to be executed in both VIPPE and the Imperas OVP tool, so this comparison could be performed in more controlled conditions. The results obtained allowed to draw conclusions about the differences in terms of performance of the two simulation technologies (<i>native</i> against <i>OVP-based</i>).
--	--

Task	Compare results
Status	Completed.
Summary of results	The detailed results about the comparison between <i>native</i> and <i>OVP-based</i> simulation technologies can be found in section 3.2.3.1.

Task	Draw conclusions on the CONTREX integrated flow
Status	Completed.
Summary of results	Detailed results from this task are provided in section 3.2.3.2.

3.2.3 Evaluation results

As said in previous section, the utility, reliability and effectiveness of CONTREX approach as well as the benefits it was expected to bring to the current avionics development flow have been assessed in WP5 according to the evaluation strategy outlined in D1.3.2. This strategy consisted mainly of the:

1. Assessment of the accuracy of the estimations and results obtained using the CONTREX methodology.
2. Assessment of the efficiency gain when using the CONTREX-enhanced development flow instead of previous flow, considering the objectives enunciated above.

This section presents the results of these assessments.

3.2.3.1 Assessment on the speed and accuracy of the estimations.

In the avionics flow, a main objective is the achievement of a fast exploration of the design space, releasing the system designer from the need to implement each explored alternative, or to have available each assessed platform. This way, a fast, high-level assessment of the convenient future implementations is achieved.

The DSE methodology integrated in the CONTREX Eclipse Plug-in (CONTREP) relies on two tools: VIPPE for the simulation and assessment of performance and MOST for the exploration. In the evaluation, the smooth integration of these tools has been checked. The DSE infrastructure generated by CONTREP serves to automatically launch the design exploration loop. In that loop, the time taken by MOST to read results and trigger the next solution is negligible compared to the time to simulate each solution. At the end of the simulation, there is also a time for post-processing and producing, e.g. an HTML report, in case it is required to the tool, and which is practically constant. Since dynamic thermal simulation is not integrated in the automated DSE loop, the main concern is on the simulation performance (given an accuracy that allows for design decisions).

Following figures on synthetic examples show the benefits and trade-offs in the use of VIPPE technology. Synthetic examples are required to ensure the determinism and thus fairness on the inputs to the tools being compared. In CONTREX, OVP-based simulation technology is taken as the reference technology for the evaluation of VIPPE.

The synthetic examples shown in Table 4 refer to an example of a bubble sort of 10k elements, and to a factorial of relatively large number (20k), to make the simulation time more relevant. The table shows the simulation and estimation times for OVP and VIPPE. The “speed-up” column shows the speed-up of VIPPE vs OVP (values bigger than 1 indicate that VIPPE outperforms OVP and vice-versa). These examples shown stress VIPPE in several aspects in order to enable a fair comparison (or at least to ensure that VIPPE is not favored and that VIPPE will show better performance on different conditions). The bubble sort has small basic blocks, which degrades VIPPE simulation speed, because it adds more annotations per simulated time unit. The factorial example includes a large amount of recursive calls in comparison with most of the applications, which is a factor degrading accuracy on instructions estimations of the user code. Additionally, the table shows the results for a void application (starts and returns) right to let quantify the time required to load and configure the platform plus the later results report at the end of the simulation.

Table 4 Evaluation of execution times for a native version of the application and for simulation times for OVP and VIPPE in bare metal.

Example	Optimizations	Cache	OVP	VIPPE	speed-up
bubble sort(10k)	No	no	0m1.557s	0m2.776s	0,56
bubble sort(10k)	-O2	no	0m0.864s	0m0.989s	0,87
bubble sort(10k)	No	1way32B64Kwb	1m4.898s	0m5.594s	11,60
bubble sort(10k)	-O2	1way32B64Kwb	0m23.675s	0m1.374s	17,23

Report on evaluation of the integrated design flow (final)

factorial (20k)	No	no	0m3.963s	0m9.404s	0,42
factorial (20k)	-O2	no	0m0.799s	0m1.374s	0,58
factorial (20k)	No	1way32B64Kwb	2m34.301s	0m32.403s	4,76
factorial (20k)	-O2	1way32B64Kwb	0m27.374s	0m1.604s	17,07
Void	No	no	0m0.233s	0m0.013s	17,92
Void	-O2	no	0m0.262s	0m0.012s	21,83
Void	No	1way32B64Kwb	0m0.291s	0m0.020s	14,55
Void	-O2	1way32B64Kwb	0m0.242s	0m0.030s	8,07

Several observations which can be directly checked on the data of the table can be done.

A first one is that VIPPE is comparable to OVP, i.e. they are in the same range of values, when estimating performance (instructions) in a scenario of processor without cache¹. Specifically, in Table 4's examples, OVP runs in around half of the time.

Later on, a fairer comparison for the scenario without caches (but with RTOS in the system model) is provided.

In any case, the OVP performance for this scenario is surprisingly good if we take into account the difference among estimation methods (binary translation vs source code annotation). The hypothesis handled so far is that OVP actually relies on a translation buffer which acts as a cache, so that for recurrent code as the case of the examples shown, the translation effort is highly reduced on the overall, and the simulation times have a an overhead with respect to a host simulation of 1 degree of magnitude or even less. This is concluded after comparison with Table 5 show this fact, which shows measurements on the applications compiled and running for the host computer (data for -O2 optimizations are not considered as in that case, the compiler removes all the operational code). In any case, this is a performance similar to native simulation in the no cache scenario.

Table 5 Execution times in host of bubble sort and factorial examples

Example	Optimizations	Native
bubble sort(10k)	No	0m0.258s
Factorial (20k)	No	0m1.158s

A second observation based on Table 4 refers to a scenario where caches impact needs to be estimated. This is fundamental in mixed-criticality design context, where the goal is the exploitation of low-cost performance optimized architectures. In such a context, VIPPE clearly outperforms OVP (speed-up x5 in most of the cases). In must be also considered that the cache configuration tested (1way, line size 32bytes, size 64K, write-back) is expected to be among the most painful for VIPPE in terms of estimation times.

¹ For a fair comparison in a scenario of processor without caches, VIPPE annotations of caches where disabled. The table shows the simulation and estimation times for OVP and VIPPE.

Report on evaluation of the integrated design flow (final)

VIPPE simulation speed comes at an assumable cost in precision.

VIPPE simulation speed comes at an assumable cost in precision, i.e. which enables effective DSE. Table 6 provides figures on the amount of instructions estimated by VIPPE in contrast to OVP. OVP here is the reference, as it computes and thus accounts accurately every single instruction present in the assessed object code. Table 6 figures refer to the execution on a CortexA9, i.e. armv7 instruction set. As seen, in the painful example with a large amount of recursive calls (factorial) and still considering compiler with optimizations, VIPPE keeps a 25% bound error, while in other cases get assumable accuracies (around 13% or less).

Table 6 Accuracy in the estimation of instructions of VIPPE taking OVP as reference.

Example	Opts.	Cache	OVP	VIPPE	diff
bubble sort(10k)	No	no	1300090016	1200089895	-8%
bubble sort(10k)	-O2	no	550035007	550074948	0%
bubble sort(10k)	No	1way32B64Kwb	1300090016	1200089895	-8%
bubble sort(10k)	-O2	1way32B64Kwb	550035007	550074948	0%
factorial (20k)	No	no	3000012600	3399989560	13%
factorial (20k)	-O2	no	800022455	999989901	25%
factorial (20k)	No	1way32B64Kwb	3000012600	3399989560	13%
factorial (20k)	-O2	1way32B64Kwb	800022455	999989901	25%

Analyses conducted by the end of this report let check that the accuracy for the cache analysis associated to the reported data is also sufficient for DSE. For instance, for the bubble sort example, OVP estimations report 1281 and 1280 total misses (for the optimized and non-optimized case), while VIPPE reports 1251 and 1252 respectively.

Figure 9 summarizes the results of the analysis of the scalability of VIPPE and of two professional variants of the OVP simulator. The experiment consists in launching N threads (executing the bubble sort of 10K elements) without any coupling among them. Therefore, the experiment also tested VIPPE for cases of platforms with OS, specifically SMP-OS.

Specifically, Figure 9 shows the simulation times for the cases of 1, 2, 4 and 8 threads, being executed on an ARM Versatile platform of 1, 2, 3 and 4 cores respectively. The data show how the OVP simulation relying on the CPUMulti version of the OVP simulator (version available with the Europractice license, capable to simulate heterogeneous platforms and with other nice features) augment the simulation times linearly with the amount of threads (and thus with the computational load to be simulated). Therefore, this simulator is not capable to exploit the underlying parallelism of the host platform.

However, if the parallelized version of the OVP simulator (the QuantumLeap technology²) is used, the results show how the underlying parallelism of the host platform can be exploited (in this experiments a 32cores machine was used, and it was checked that the load let guarantee

² UC want to recognize and thank Imperas for providing the QuantuLeap license for research Imperas for research purposes.

Report on evaluation of the integrated design flow (final)

the tools to use as many cores as threads in the experiments). It can be appreciated that OVP exploits up to 4 cores (the amount of cores in the target platform).

Figure 9 shows how VIPPE can exploit equally good as the cutting-edge QuantumLeap technology the underlying parallelism. In the case of Figure 9, VIPPE simulation times are slightly over the QuantumLeap ones (remind that Figure 9 took the less favouring case for sequential estimation for VIPPE).

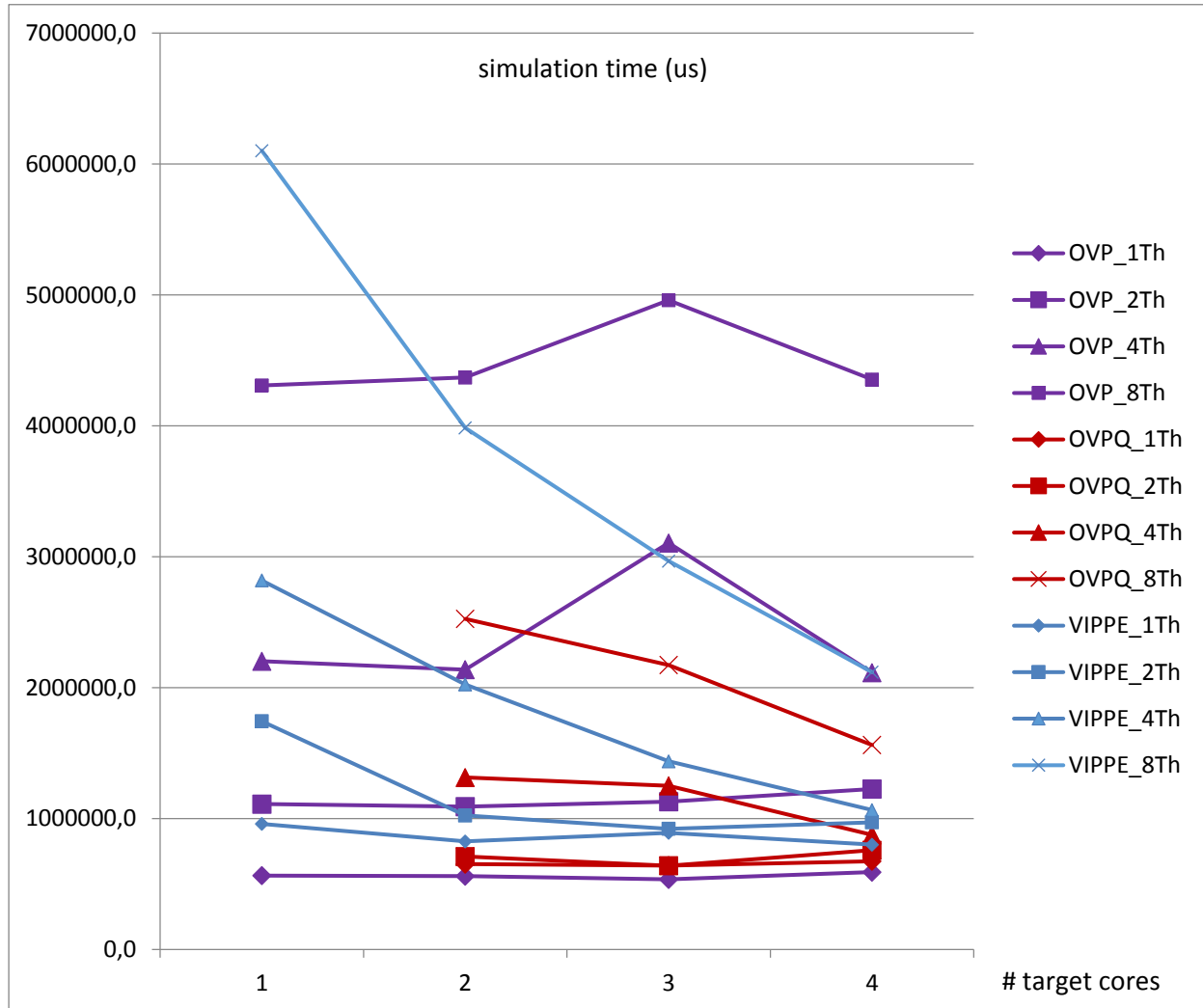


Figure 9. Simulation speed scalability of OVP (EuroPractice), OVP+QuantumLeap and VIPPE with the number of threads and target processing cores (host processors 32) without caches.

Reminding that the analysis leading to Figure 9 results has been done for the worst conditions for VIPPE (i.e. without cache estimation), it is expected that in a similar analysis VIPPE improves its position in Figure 9 in a similar range as was reported in Table 4.

The conclusion is that VIPPE is then the technology of choice for simulation for DSE. Regarding simulation speed, the tool clearly outperforms OVP on cache estimation scenarios, which are crucial in MCS and other design contexts.

Table 7 provides the results of a further analysis on the simulation time overheads of the different types of performance accounting. The results show how the cache estimation is the most noticeable aspect to consider in the simulation time.

Report on evaluation of the integrated design flow (final)

Table 7 Impact on simulation time of annotations for performance assessment.

Time (s)	Relative increment Vs (%)	Conditions
0.75	-	(default) time
0.932	24.3	+ instructions accounting
0.945	26.0	+ cycles accounting
1.076	43.5	+ instruction & cycles accounting
1.167	55.6	+ instruction & cycles & energy accounting
5.763	668.4	+ instruction & cycles & caches accounting
6.496	766.13	+ instruction & cycles & caches & bus & memory accounting
6.872	816.27	+ instruction & cycles & caches & bus & memory & energy accounting

Table 7 also shows that the estimation of other additional performance metrics (e.g. energy) come at a relatively lower cost. In other words, a holistic exploration (considering time, energy and power figures) can be enabled at a reasonable exploration time cost.

Moreover, VIPPE has the potential for still faster simulation times. Current version of VIPPE does not implement enhancements on the annotation techniques that have been recently published and proposed. These techniques enable lighter annotations after some static analysis of the basic-block tree.

Another very important advantage is an easier and more abstract development of the model of each assessed solution. Moreover, VIPPE allows building configurable model “templates”, which reflect a design space. These configurable models are already integrated with exploration tools, i.e. MOST.

3.2.3.2 Assessment of the efficiency gain with CONTREX-enhanced flow.

Results presented in previous section provide enough evidence about the reliability of the performance estimations and metrics provided by the VIPPE (native simulation) tool, which was the simulator integrated into the avionics demonstrator tool-chain.

Once this is proved, the efficiency gain obtained with the CONTREX approach can be assessed. As mentioned in previous sections, this has been done considering two different perspectives:

- *Resources savings.* Comparison between usual estimations on resource needs and those obtained from the application of the CONTREX approach.
- *Development time savings.* Comparison between times spent in exploring the design space automatically and manually, and comparison of total time spent in the development of the Use Case 1 using CONTREX approach against the total development time that would be spent in case a problem was detected at the HW-SW integration phase.

Resources savings.

Report on evaluation of the integrated design flow (final)

In the traditional avionics development flow, both the HW/SW partitioning decision and the platform configuration are made at an early stage of the cycle, being usually based on the designers' expertise. In order to avoid late integration issues, the design space is strongly limited and only a small number of possibilities are considered. For the same reason, the quantity and capacity of resources (processors, memories, etc.) is usually oversized (especially in the case of mixed-criticality systems, due to the *spatial and temporal isolation* principle). In critical systems, the need for using certified systems constrains even more the design space alternatives and hinders the usage of modern low-cost and efficient all-purpose MPSoC.

This was the case of the FCC (Flight Control Computer) developed by GMV for a medium sized Remotely Piloted Aircraft. The FCC, which was in charge of the guidance, navigation and control of the RPA, comprised the power supply converter and hold-up card, the navigation sensors, a CPU where most of FCC SW was deployed and a dedicated I/O card implemented onto the same CPU board for connections with sensors and processing sensor data (see Figure 10).

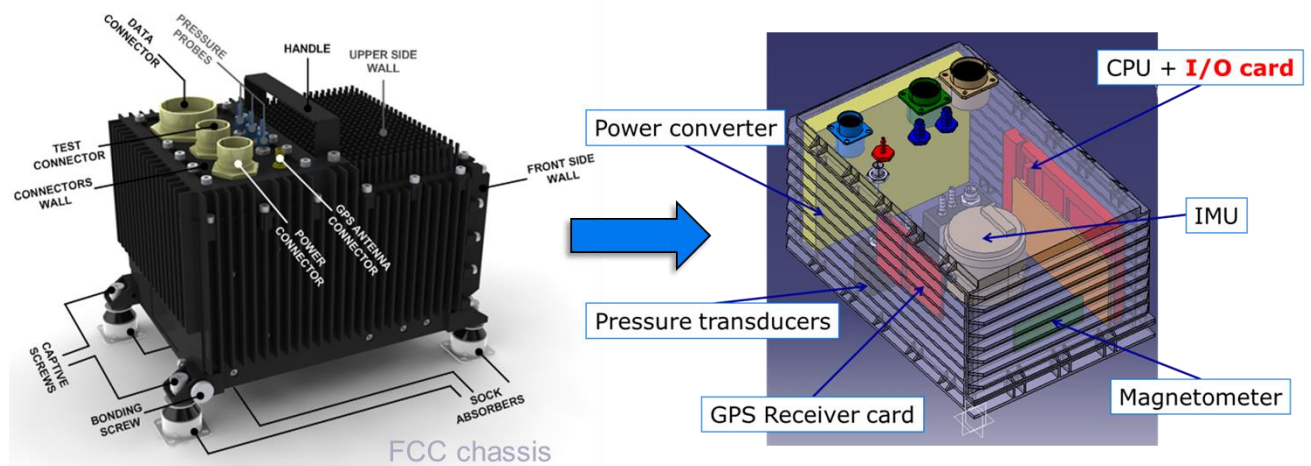


Figure 10. Flight Control Computer.

The kind of reasons exposed in previous paragraphs led to the usage of a certified version of the VxWorks OS running on a PowerPC processor and a specific I/O card for management of the corresponding sensors and actuators. The board composed by the CPU plus this I/O card (as shown in previous figure) was measured to consume about 20 W of power.

As said in section 3.2.1, GMV aims to tailor the FCC SW to future RPA platforms, particularly, to light RPAs (<150 kg). As SWaP constraints are a key factor for light RPA equipment, the usage of components as those mentioned above becomes unaffordable and makes it necessary to adapt the FCC SW to low-cost commercial all-purpose platforms.

The avionics demonstrator for CONTREX project consisted of the sensor I/O module (i.e., the functionality deployed onto the I/O card) of the FCC discussed above, running on the commercial Xilinx Zynq 7020 MPSoC. This demonstrator poses a representative case of adaption of the FCC legacy SW to an efficient all-purpose commercial platform.

Apart from some issues regarding the spatial and temporal isolation principles (timing, power and temperature interferences), inherent to the real-time, mixed-criticality nature of the application and the general-purpose character of the platform (see D4.2.2 for a more detailed discussion about these issues), which would need to be further investigated, and other concerns

regarding the certifiability of critical systems, the feasibility of such adaption has been sufficiently demonstrated during CONTREX. Moreover, CONTREX enhancements have demonstrated to be a promising approach for estimating such feasibility.

Apart from the evident reduction in size and weight with regard to the original configuration, one of the first consequences of the adaption of the FCC's sensor I/O module to the Zynq MPSoC is a significant **reduction in power consumption**.

Coming back to the original configuration of the FCC system, it was said that the power consumption of CPU and I/O card board was about 20 W. It was estimated that about 14 W of this consumption was due to the CPU and its associated auxiliary circuitry. The remaining **6 W** were thus attributed to the I/O card (highlighted in Figure 10).

As said, the I/O card's functionality (referred as the sensor I/O module) is the one adapted for CONTREX and executed onto the Zynq MPSoC. This platform, with a power consumption of about **3,5 W**, as reported by the manufacturer, together with the auxiliary circuitry developed in CONTREX for extra UART ports (see Figure 11) estimated to consume about **1 W**, sum up an overall of **4,5 W**.

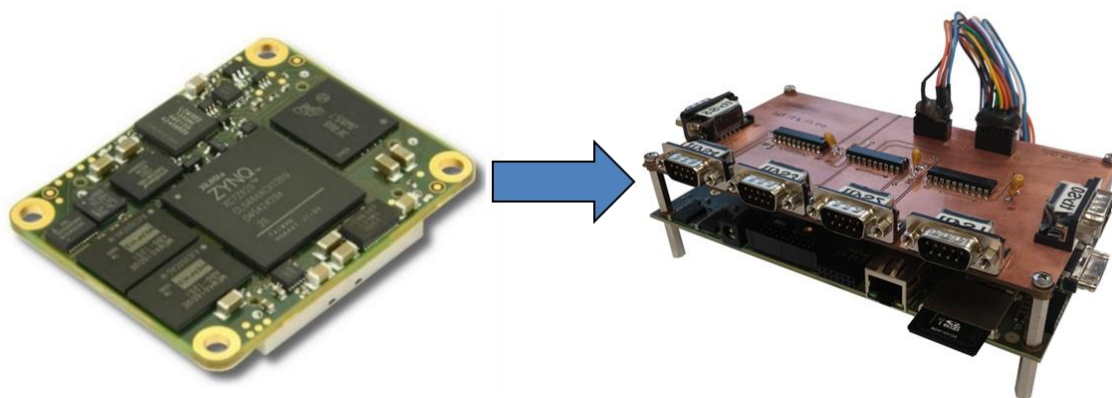


Figure 11. Zynq board customization for Use Case 1b.

This supposes a **reduction of about 25% in power consumption** with regard to the original platform, achieving thus one of the CONTREX main objectives. In addition to this, during system simulations and design space exploration it was observed that the sensor I/O functionality was not too demanding for the computational power of the Zynq board, so additional functionality could be potentially executed on it (as for instance the *Fault Detection, Isolation and Recovery, Navigation* or *Flight Control Laws* functionalities), leading to a significant scale up of the functionality running on a single MPSoC.

Development time savings.

There is also a second perspective from which the potential benefits brought by CONTREX enhancements were to be evaluated.

As said in section 3.2.1, the traditional avionics flow was not devised for the new context of tailoring legacy avionics SW to the commercial all-purpose platforms that the new light RPA market demands. Thus, the CONTREX enhancements were expected to bring a more flexible approach that enable an early and reasonable (in terms of cost and time) assessment of the

Report on evaluation of the integrated design flow (final)

performance of the legacy system on the new platforms, with the purpose of finding those optimal configurations that minimize cost, size, weight and power consumption without compromising safety and overall performance.

Finding those optimal solutions would require exploring a considerable amount of alternatives (different platforms, different SW/HW mappings for each platform)

While, due to the demonstrator nature of the project, only a few design alternatives have been explored in the context of CONTREX, they have been enough to provide strong evidence about the feasibility of automatically exploring huge design spaces in a more than acceptable amount of time. Hundreds of design alternatives could be simulated as part of the DSE loop (without user interaction in the meanwhile) in a period of time on the order of a few hours. Needless to say that the manual exploration of such a design space is completely unachievable.

The introduction of the simulation and DSE loop in the avionics development design phase (see Figure 7) increases its duration as much time as the DSE loop takes to provide results. To this, the time required to define the design space should be added (the rest of design time is comparable to the time devoted to system design in a classical development process). Once the engineers are trained and get familiar with the new methodology, this latter time can be considered negligible. Regarding the first one, it is clearly compensated by avoiding HW/SW integration issues and thus late re-design work.

4 Automotive domain

This section provides a description about the current status and preliminary results of the validation of the automotive domain's requirements that are applicable to the integration of the design tools used in CONTREX Use Case 2.

Tools coverage overview

In the scope Automotive domain, the following tools will be integrated into the pre-existing design flow and evaluated:

- Intel® Docea™ Power Simulator. Power modelling tool, used for the SeCSoC platform.
- EFPM. Extra-Functional Property Monitoring infrastructure.
- BBQLite. Low-cost sensor node run-time management firmware, used for the iNemo.
- N2Sim: N2Sim augmentation for automotive telematics network scenario.

The following Figure 12 shows how the CONTEX tools have been integrated into the pre-existing design flow.

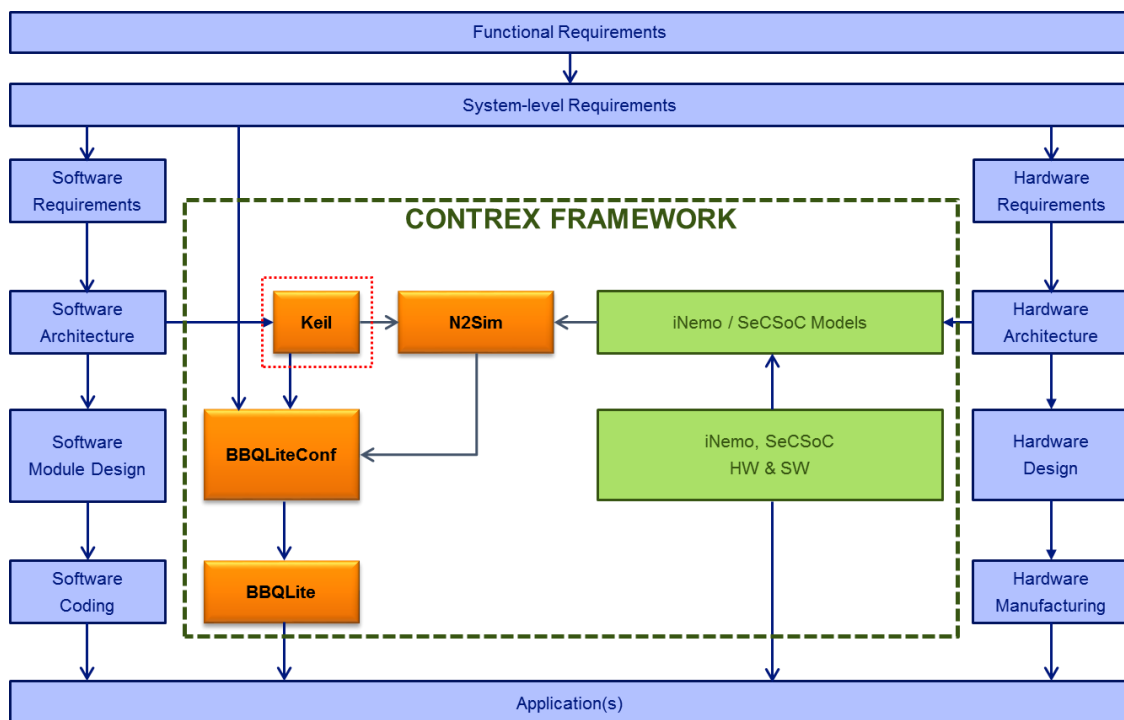


Figure 12. Overall CONTEX tool-flow for Use Case 2.

Preliminary evaluation results

The following subsections describe how the different tools have been integrated with each other and with the pre-existing design flow.

4.1.1 Integration of Intel® Docea™ Power Simulator in the toolflow

The Power Simulator tool is an independent tool that can use as input the output of the SeCSoc virtual platform. It is used to replace the current virtual platform in order to add the contribution of voltage regulator to the platform (not modelled in the virtual platform).

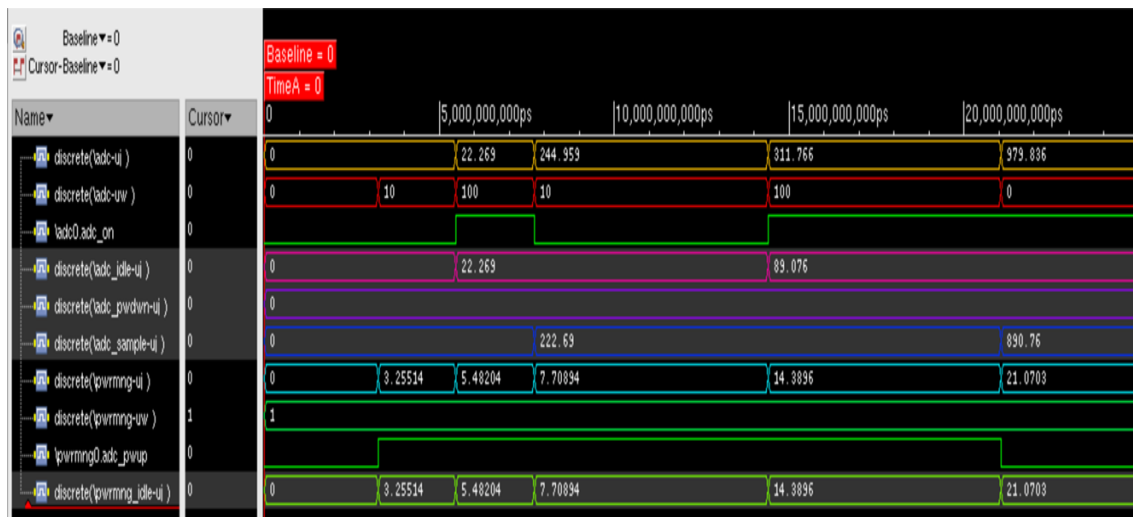


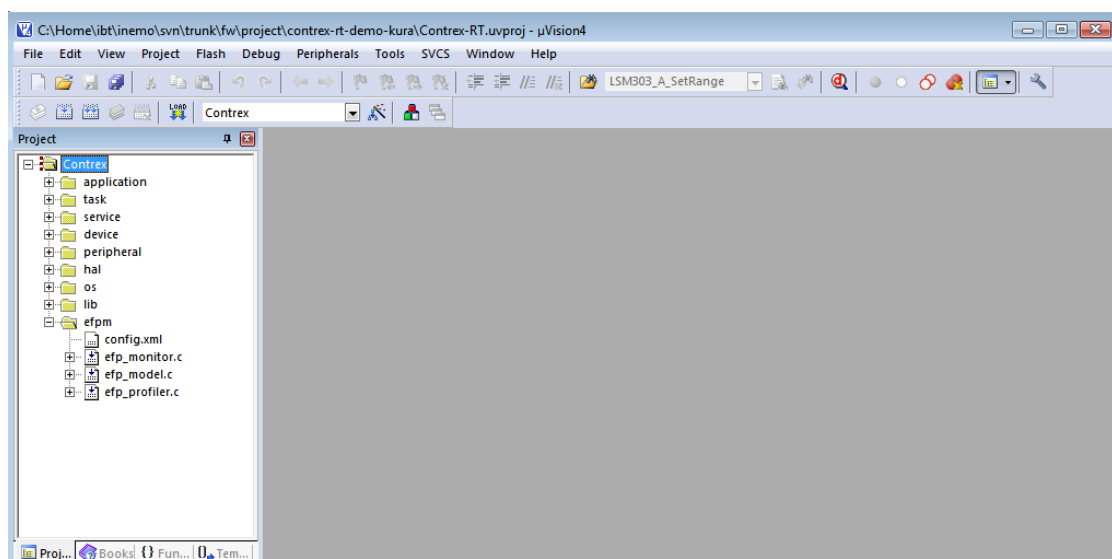
Figure 13: Power Trace file from SecSocVirtual Platform

It is a standalone tool and the integration into the overall workflow was ensured by using a standard format file (the vcd) to capture the activities

4.1.2 Integration of EFPM in the firmware development design flow

The Extra-Functional Property Monitoring infrastructure has fully been integrated into the standard firmware design flow previously used by Vodafone. The integration required the following seamless steps:

1. Creation of a new folder “efpm” in the project structure devoted to host the configuration files and the generated code.



Report on evaluation of the integrated design flow (final)

Figure 14. Adding EFPM to the project structure

2. Creation of new folder “tools/epfmg” In the project structure to host the code generator scripts.
3. Configuration of Keil uVision to run a pre-build script that is in charge of the generation of the EFPM infrastructure code.

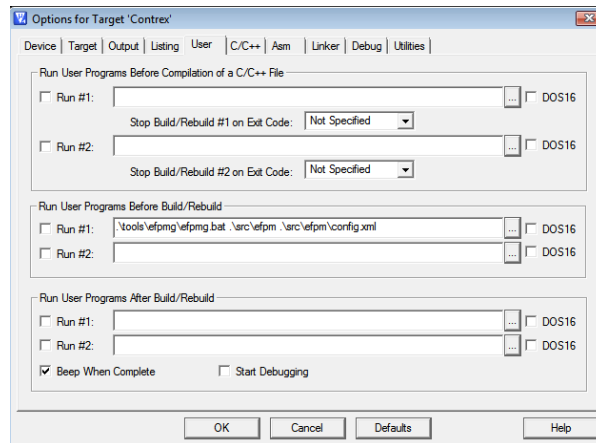


Figure 15. Automating code generation as a pre-build script

4. Integration of the measurement infrastructure in the tasks and drivers is straightforward. The following code excerpts show in bold the modification to the original code that have been performed to enable monitoring.

```

__task void TASK_RTRM( void )
{
    uint32_t E_ACC;    // Energy consumed by the accelerometer
    uint32_t E_UART;  // Energy consumed by the UART interface
    uint32_t E_TOT;    // Energy consumed by all the monitored sub-systems

    // Initialization of the EFPM infrastructure
    EFP_FRAMEWORK_INIT();

    // Starts the profiler
    EFP_PROFILER_BEGIN(EnergyProfiler);

    // Periodic task with 1s period
    os_itv_set(100);

    while( 1 )
    {
        // Waits for the timer to trigger
        os_itv_wait();

        // Stops the profiler and retrieves the metrics
        EFP_PROFILER_END(EnergyProfiler);
        EFP_PROFILER_GET_AVERAGE(EnergyProfiler, E_ACC, E_UART, E_TOT );

        // Original code
        ...

        // Restart the profiler
    }
}

```


Report on evaluation of the integrated design flow (final)

```

    EFP_PROFILER_BEGIN(EnergyProfiler);
}
}

```

Figure 16. Integration of EFPM into an application task

```

void LSM303_A_Read( OUT Tern12_t* tern, uint8_t fifo_length )
{
    uint8_t reg    = LSM303_A_CMD_AUTOINCREMENT | LSM303_A_OUT_X_L;

    // Starts updating the energy of the device
    EFP_MODEL_UPDATE(LIS,READ_START);

    PMU_TransactionBegin();
    I2C_Read(LSM303_A_I2C_ADDRESS, &reg,(uint8_t*)tern, 6*fifo_length);
    for( i = 0; i < fifo_length; i++ ) {
        tern[i].x = fp_rsh( tern[i].x, 4 );
        tern[i].y = fp_rsh( tern[i].y, 4 );
        tern[i].z = fp_rsh( tern[i].z, 4 );
    }
    PMU_TransactionEnd();

    // Finishes updating the energy of the device
    EFP_MODEL_UPDATE(LIS,READ_END);
}

```

Figure 17. Integration of EFPM into an application driver

These four steps provide the basic integration mechanism for the automatic generation of the monitoring infrastructure. Once the adapted toolchain is in place, the integration of the monitoring infrastructure into the actual firmware requires adding a few macros to define the sampling points and the measures that have to be collected. The specific macros, their syntax and their meaning have been describe in Deliverable D3.4.2.

4.1.3 Integration of BBQLite firmware development design flow

The BBQLite run-time manager has been fully integrated into the standard firmware design flow previously used by Vodafone and is currently under final testing. The figure below shows how the BBQLite infrastructure is related to the tool-flow (design-time) and to the application execution (run-time).

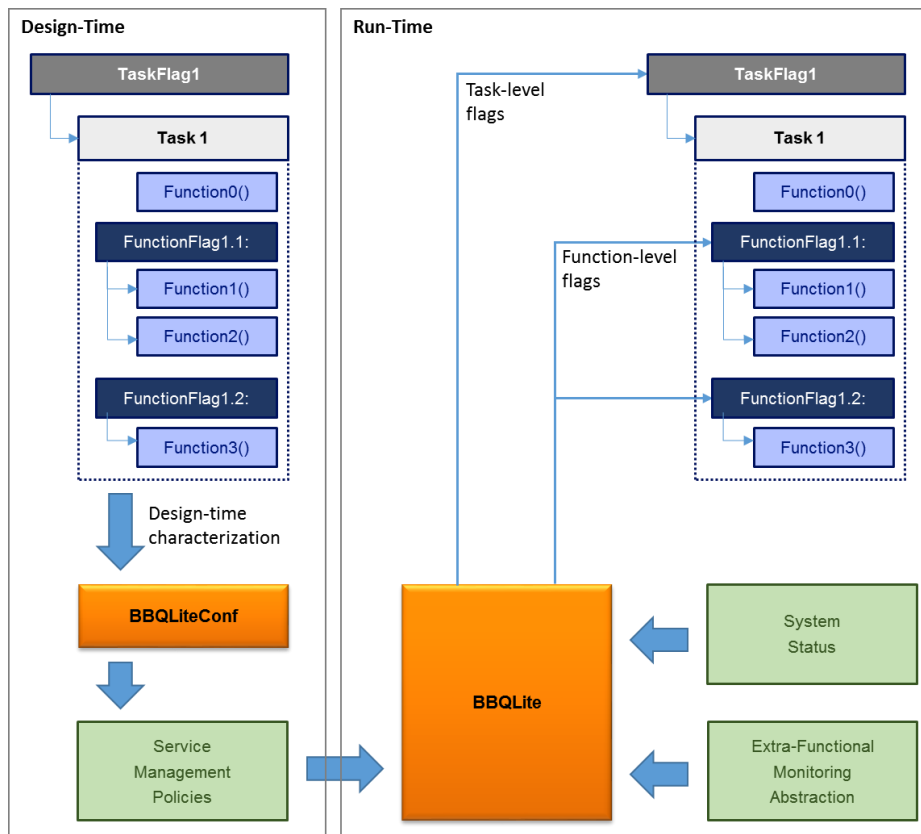


Figure 18. Integration of BBQLite into the existing flow/application

The integration required the following steps:

Report on evaluation of the integrated design flow (final)

1. Creation of a new folder “bbqlite” in the project structure devoted to host the configuration files, the BBQLite API and the generated code.

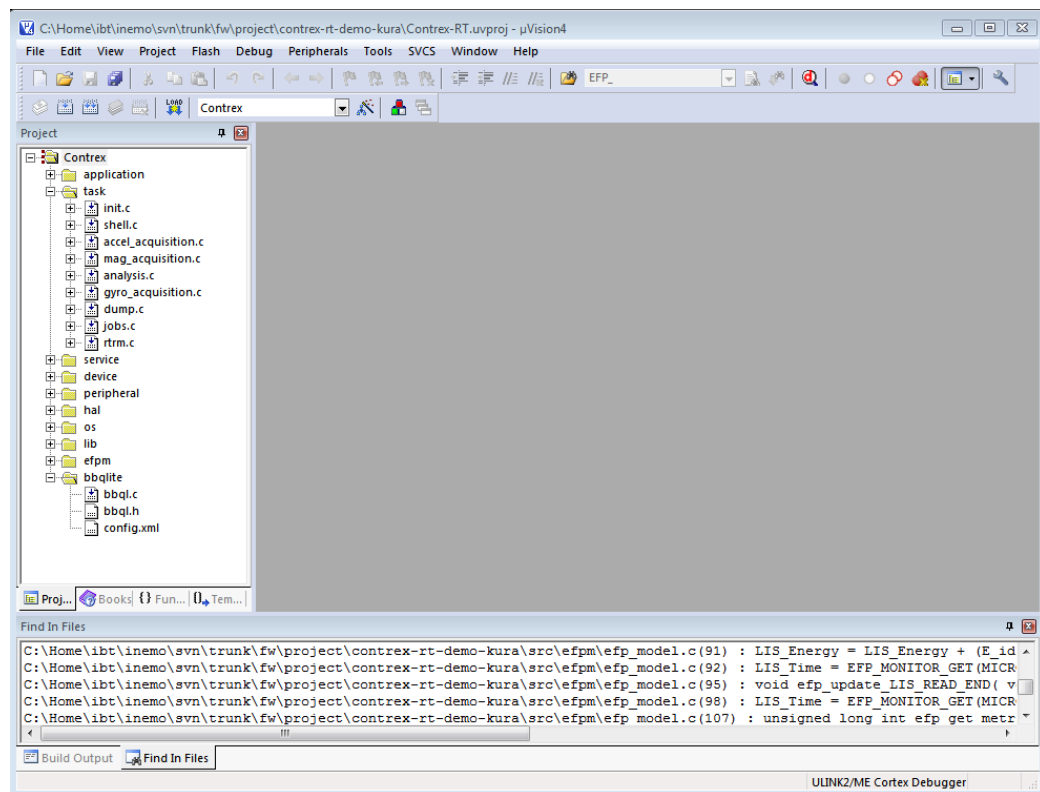


Figure 19. Adding BBQLite to the project structure

2. Creation of new folder “tools/bbqliteg” In the project structure to host the code generator scripts.
3. Configuration of Keil uVision to run a pre-build script that is in charge of the generation of the BBQLite configuration code.

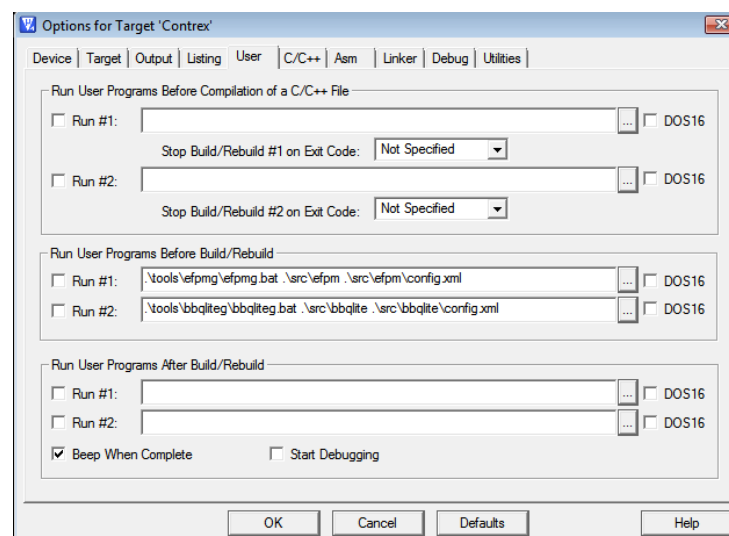


Figure 20. Automating code generation as a pre-build script

Report on evaluation of the integrated design flow (final)

4. Integration of the management infrastructure in the application requires minimal changes to the code, namely, the creation of a new task that collects the non-functional metrics, invokes the main BBQLite management function and selectively executes individual “jobs” under the control of the run-time manager.

```

__task void TASK_RTRM( void )
{
    // Original code
    ...

    while( 1 )
    {
        // Waits for the timer to trigger
        os_itv_wait();

        // Updates the flags according to:
        // - Functional status (operating mode)
        // - Non-functional status (from the EFPM monitor)

        BBQL_Manage();

        // Runs selected tasks based on the newly determined
        // set of flags

        if( BBQL_GetFlag(BBQL_FLAG_SelfCalibration) == BBQL_ENABLED ) {
            JOB_SelfCalibration();
        }

        if( BBQL_GetFlag(BBQL_FLAG_LowEnergy) == BBQL_ENABLED ) {
            JOB_LowEnergy();
        }

        ...
    }
}

```

Figure 21. Integration of BBQLite into the run-time management task

5. Integration of the BBQLite power management function into the idle task of the operating system and into the drivers’ interrupt service routines. This is shown in bold the code fragments that follow.

```

__task void os_idle_demon( void )
{
    for(;;)
    {
        // Puts the core in sleep.
        // The frequency needs not to be changed
        BBQL_SetCoreMode(BBQL_CORE_MODE_SLEEP,
                        BBQL_CORE_FREQUENCY_DEFAULT);
    }
}

```

Figure 22. Integration of BBQLite into the idle task for core power management

```
void EXTI9_5_IRQHandler( void )
{
    // Puts the core in sleep.
    // The frequency needs not to be changed
    BBQL_SetCoreMode(BBQL_CORE_MODE_NORMAL,
                     BBQL_CORE_FREQUENCY_DEFAULT);

    if( EXTI_GetInterruptStatus( 8 ) == 1 )
    {
        EXTI_ClearInterrupt ( 8 );
        isr_evt_set( EVENT_LSM303_A_READY, TID_ACCEL_Acquisition );
    }

    if( EXTI_GetInterruptStatus( 6 ) == 1 )
    {
        EXTI_ClearInterrupt ( 6 );
        isr_evt_set( EVENT_L3G_READY, TID_GYRO_Acquisition );
    }
}
```

Figure 23. Integration of BBQLite into an ISR for core power management

These five steps provide the basic integration mechanism for adding run-time management to an existing application. Once the adapted toolchain is in place, the integration of the run-time management infrastructure into the actual firmware requires adding a few function calls in selected points in the code. The opportunity to selectively change the power modes of individual peripheral has not been exploited since the microcontroller provides limited power gating capabilities and the clock gating approach does not significantly improve the power reduction that can be obtained. For more details on the specific function calls, their syntax and their meaning, refer to Deliverable D3.3.1. and D3.3.2.

4.1.4 Integration of N2Sim with the firmware development design flow

The integration of the non-functional node simulator into the existing Vodafone Automotive design flow is actually quite loose, since the simulator is an analysis tool whose results need to be understood and evaluated by a human designer, who will use them as a support to define the power-management policies. Such policies are then described and fed as input to the BBQLite configuration tool, that generates the suitable configuration files ready for integration in the production code.

Report on evaluation of the integrated design flow (final)

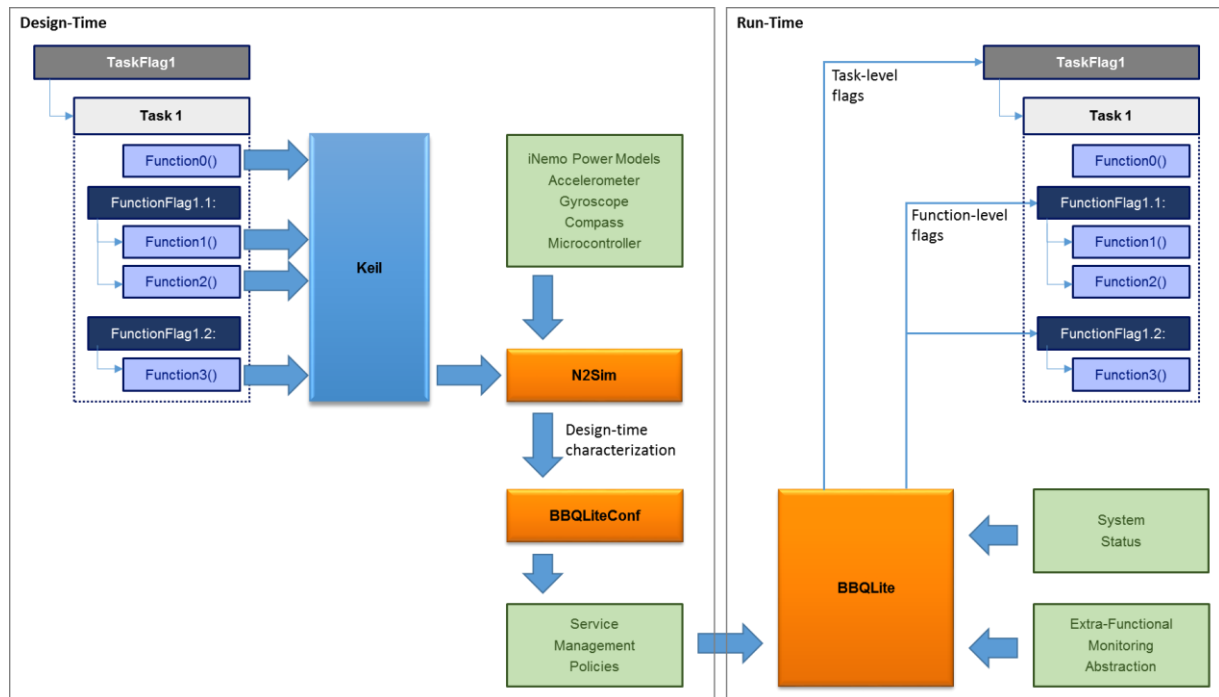


Figure 24. Integration of N2Sim into the design flow

To use the simulator, a suitably simplify model of the application architecture must be provided. Such a model shall represent the main components of the application, namely:

- All the relevant hardware signals. In the specific case, such signals are the two interrupts of the inertial measurement unit of the iNemo platform and the UART receive interrupt.
- The related interrupt service routine and the relevant portion of the drivers, namely read/write functions. All configuration functions can be neglected since they are execute only once at boot time and do not significantly influence the overall power consumption.
- The main functional portions of the code, split between services, jobs and tasks.
- The software signals (events, semaphores and flags) that influence the execution and synchronization of tasks.

It is worth noting that for software portions, the information that must be provided to the simulator only consists in average execution time of functions and the set of signals that are sensed and/or generated by such functions.

The specification is text-based only, but the model structure and the functional/non-functional details can be captured easily. As an example, consider the following model excerpts showing an IRQ/ISR pair (Figure 25) and synchronization of two tasks (Figure 26).

Report on evaluation of the integrated design flow (final)

```

// Hardware LIS device model
ENTRY(lis_sample)
START(lis_sample) {
    hwfifo.push(); // Samples go to hw fifo
    POST( lis_sample_duration, ACTION(lis_sample,End) ); // Schedules end action
    POST( lis_sample_period, ACTION(lis_sample,Start)); // Implements periodicity
}
END(lis_sample) {
    if( hwfifo.level() >= 5 ) { // When hardware fifo full...
        POST( 0, ACTION(lis_irq,Start) ); // ... generates an IRQ
    }
}

// IRQ model
ENTRY(lis_irq)
START(lis_irq) {
    POST( lis_irq_duration, ACTION(lis_irq,End)); // Schedules end action
}
END(lis_irq) {
    POST( 0, ACTION(lis_isr,Start), 0 ); // Schedules the ISR
}

// ISR model
ENTRY(lis_isr)
START(lis_isr) {
    POST( lis_isr_duration, ACTION(lis_isr,End), 0 ); // Schedules end action
}
END(lis_isr) {
    POST( 0, ACTION(task_acquisition,Start), 0 ); // Unblocks a task
}

```

Figure 25. Modelling hardware events, IRQs and ISRs

```

// Task Acquisition
ENTRY(task_acquisition)
START(task_acquisition) {
    lis_read();
    filter();
    swfifo.push(1);
    POST( T, ACTION(task_acquisition,End) );
}
END(task_acquisition) {
    if( swfifo.level() >= 5 ) {
        swfifo.clear();
        POST( 0, ACTION(task_analysis,Start) );
    }
}

ENTRY(task_analysis)
START(task_analysis) { ... }
END(task_analysis) { ... }

```

Figure 26. Modelling task synchronization

Report on evaluation of the integrated design flow (final)

The non-functional aspects are also captured rather straightforwardly, as the following Figure 27 shows.

```
// Device power consumption in different operating modes
//      Device  State      Voltage  Current
DECLARE_VI( CPU,    idle,    1.8,      0.55 )
DECLARE_VI( CPU,    running, 1.8,      9.00 )
DECLARE_VI( LIS,    idle,    1.8,      0.12 )
DECLARE_VI( LIS,    sampling, 1.8,     6.50 )
...

// Events and Jobs
//      Device  Event          Duration  Period  Power
DECLARE_EVENT( LIS,    lis_sample,      2,      800,    VI(LIS,sampling) )
DECLARE_EVENT( LIS,    lis_irq,         1,       0,    VI(LIS,irq      ) )
DECLARE_EVENT( CPU,    lis_isr,        150,     0,    VI(CPU,running ) )
...
DECLARE_JOB  ( CPU,    filter,          766,     0,    VI(CPU,running ) )
DECLARE_JOB  ( CPU,    task_aquisition,  0,       0,    VI(CPU,running ) )
DECLARE_JOB  ( CPU,    task_analysis,   0,       0,    VI(CPU,running ) )
...
```

Figure 27. Modelling non-functional aspects

An appreciable improvement would consist in the development of a GUI for graphical design entry. Its development, though, is outside the scope of the project.

Preliminary Evaluation Summary

This section summarizes the result of the preliminary evaluation.

Integration goal	Intel® Docea™ Power Simulator
Status	Completed under testing.
Evaluation Results	The integration of Intel® Docea™ Power Simulator with the SecSoC virtual platform was simple and straightforward and required no modifications to the platform, thanks to the possibility to integrate by the vcd files produced by the SecSoc virtual platform. The modelling of voltage regulator is an added feature that is an ongoing activity.

Integration goal	EFPM / EFPMG
------------------	--------------

Report on evaluation of the integrated design flow (final)

Status	Completed, Tested
Evaluation Results	<p>The integration of the EFPMG code generation tools into the Keil uVision IDE has been simple and straightforward and required no modifications to the project structure, other than that documented above.</p> <p>Integration of the monitoring infrastructure into the application firmware consist in specifying the configuration as a simple XML file and adding a few macros to the existing code. The entire process required minimum effort and the instrumented code has been compiled and executed successfully.</p>

Integration goal	BBQLite
Status	Completed, Tested
Evaluation Results	<p>The integration of the BBQLite code generation tools into the Keil uVision IDE has been simple and straightforward and required no modifications to the project structure, other than that documented above.</p> <p>Integration of the run-time management infrastructure into the existing application required the introduction of a new periodic task devoted to run-time management, minor modification to the application code (refactoring of the “jobs” into isolated functions) and the introduction of a few function calls in select points in the application (the new task, the idle task, the ISRs of selected devices). The entire process required minimum effort and the augmented code has been compiled and executed successfully.</p>

Integration goal	N2Sim
Status	Completed, Tested
Evaluation Results	The definition and construction of the application model has been rather straightforward, since it basically represents relations between signals and activities. The main limitation of the current

Report on evaluation of the integrated design flow (final)

	<p>implementation consist in the absence of a graphical entry tool to simplify the model description. The specification language, though, is based on a simple structure, a limited number of keyword and minimal C code portions.</p> <p>The results of the simulator are rather exhaustive, clear and readable. Thanks to a set of post-processing filters, raw simulation results can be transformed in more abstract figures such as peak power consumption, average power consumption, average bandwidth usage, processor duty cycle and so on. In this higher level form, results are directly usable to drive decisions on the power management policies.</p>
--	--

5 Telecommunication domain

This section provides a description about the preliminary results of the validation of the telecommunication domain's requirements that are applicable to the integration of the design tools used in CONTREX Use Case 3.

Tools coverage overview

Within the context of the Telecommunication domain, the following tools have been integrated into the pre-existing design flow and evaluated:

- ForSyDe. Tool for application modelling and Design Space Exploration
- VP. Cadence Zynq Virtual Platform, tool for simulation of the Zynq platform
- HIFSuite, tool enabling co-simulation of OVP and SystemC/C++ components.
- Intel® Docea™ Thermal Profiler. Thermal modelling and simulation tool, used for the Zynq platform.

The following figure shows how the CONTREX tools have been integrated into the pre-existing design flow.

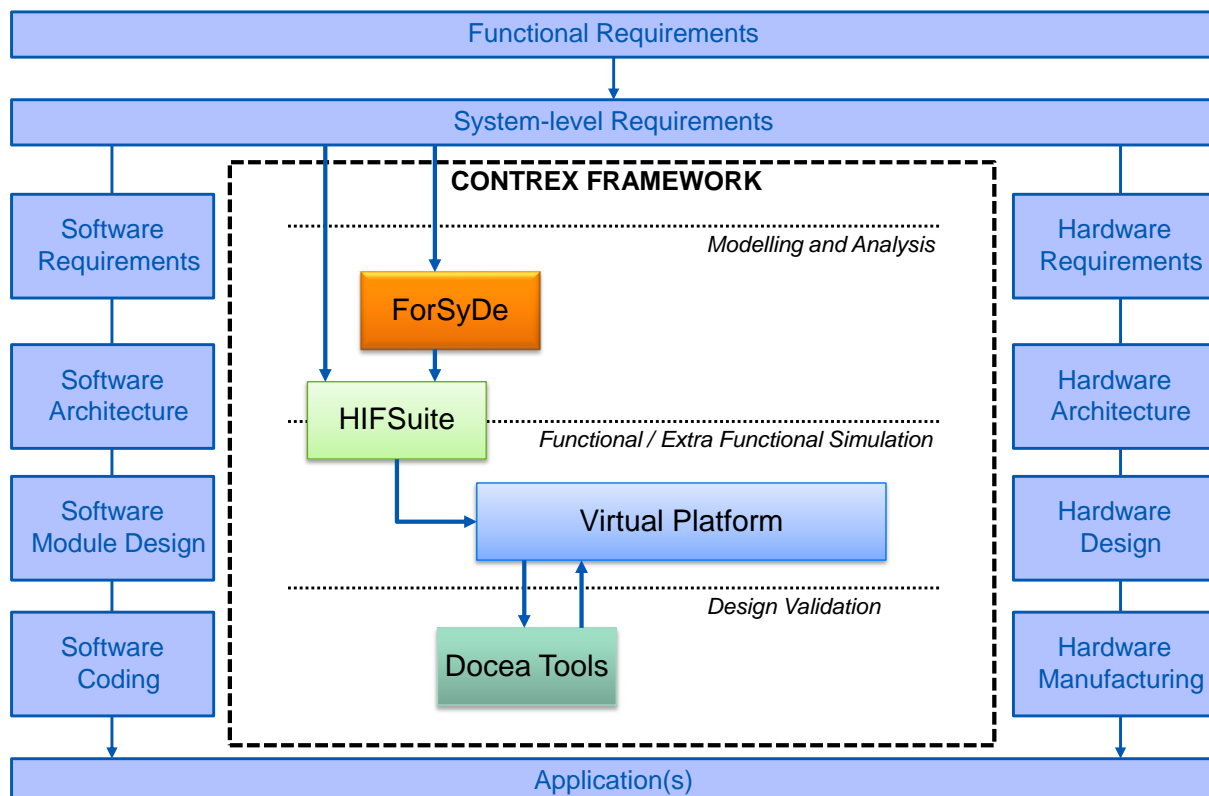


Figure 28: Overall CONTREX tool-flow for Use Case 3

Evaluation results

The following subsections describe how the different tools have been integrated with each other and with the pre-existing design flow.

5.2.1 Integration of ForSyDe into the workflow

As shown in Figure 29, the workflow in Intecs Telecommunications previously had *no upfront application modelling activities*.

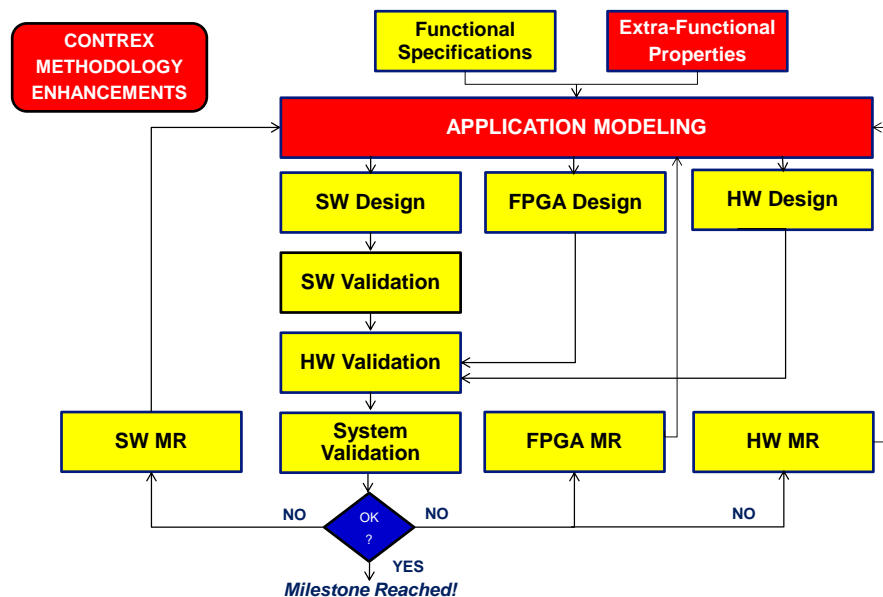


Figure 29: Introduction of Application Modelling capability into Telecom Workflow

This led in many cases to an expensive set of iterations on the application design in order to discover optimal / appropriate allocation of tasks in the given environment, which can be very complex due to the presence of software, hardware, and digital elements such as FPGAs. The ForSyDe tool fills this gap with a methodology that is very well suited to telecom applications: a network of actors, each with its own memory space, interacting through signals. Each actor is effectively defined as a finite state machine, and various models of computation are available depending on the nature of the application. In the specific case of Use Case 3, the Synchronous Data Flow (SDF) model of computation was generally employed, although hybrid models are expected to be of interest as experience is gained with the tool and methodology.

ForSyDe comes in two versions. The original version was developed in Haskell in order to provide a pure functional programming environment. It is in fact the most natural environment in which to use the tool and methodology. However, in order to provide support for application designers who are used to working with more industrial languages, a second version using the SystemC environment was created.

Both versions were used by Intecs in the experimentation activities of Use Case 3. Each version was appreciated for its particular strengths. In the end, however, the demonstrator application was implemented in the SystemC version in order to demonstrate its applicability in a typical industrial environment. There were no major problems in learning the system and in bringing the modelled application into a stable state.

5.2.2 Integration of the Virtual Platform

Since partner OFFIS was in charge of actually creating the Zynq platform abstraction over the Cadence VP, there was a tight interaction between the two partners for this step of integration into the overall telecom development workflow.

OFFIS, working independently, first implemented the Zynq abstraction, and also included additional CONTREX relevant facilities for providing power and thermal measurement traces over the simulated Zynq platform components. Parameters implemented included

- CPU clock frequency,
- memory clock frequency
- AXI clock frequency
- bitwidth of AXI interface (32 or 64)
- IO clock frequency

Once this was accomplished, Intecs provided source code for the Telecom demonstrator application and in a series of interactions, the application was successfully run on the simulated platform.

At that point, OFFIS began to provide traces from both the power and thermal simulation facilities showing the results of the simulation.

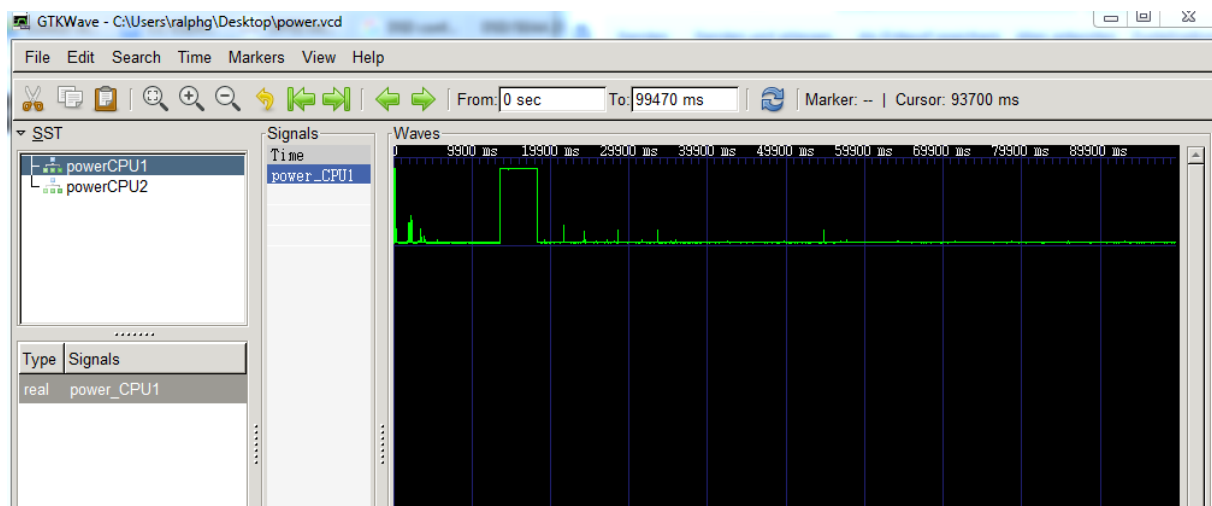


Figure 30: Power traces of telecom demonstrator application in simulated Zynq environment

Some experimentation involving frequency scaling of the Zynq processor in order to vary simulated loads was also done in Intecs.

5.2.3 Integration of custom IP blocks in the standard virtual platform

Although the Virtual Platform as integrated by OFFIS provides an important basis for raising the abstraction of design in the Telecom development environment, it is unable to completely

Report on evaluation of the integrated design flow (final)

capture and model the full extent of many of the telecom applications, because they inevitably contain custom IP that is not modelled off the shelf in the OVP platform.

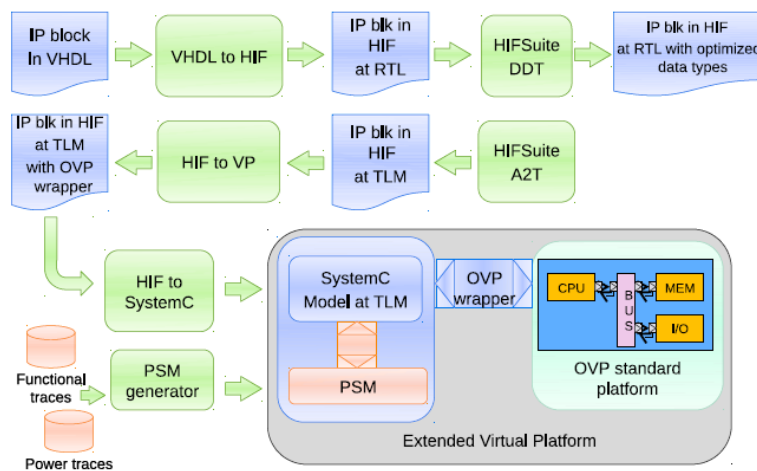


Figure 31: Integration of HIFSuite into Telecom VP Workflow

These custom IP blocks are generally modelled in Intecs Telecom at Register Transfer Level using VHDL which cannot be efficiently co-simulated with the standard virtual platform. Thus, we took the EDALab's HIFSuite tool as a solution that abstracts and integrates the custom IP blocks in the standard virtual platform. Another EDALab's tool named ODEN takes functional traces and power traces from the RTL model of the custom IP block and generates the corresponding high-level power state machine (PSM) to simulate power behaviour together with functional behaviour in the final extended virtual platform. The corresponding workflow is depicted above.

5.2.4 Integration of Intel® Docea™ Thermal Profiler in the toolflow

Intel® Docea™ Thermal Profiler was installed directly on laboratory laptop computers in the telecom development environment, with support from the Docea/Intel personnel.

It was used to replace current tools used for that purpose, adding more capability and more possibilities to work with not only the real-world environment of the Zynq platform but also with the power traces arriving from the simulated Virtual Platform environment.

Report on evaluation of the integrated design flow (final)

```

$timescale 1ms $end

$scope module powerCPU1 $end
$var real 1 a power_CPU1 $end
$upscope $end

$scope module powerCPU2 $end
$var real 1 b power_CPU2 $end
$upscope $end

$enddefinitions $end

#0
r0.61424 a
r0.61424 b
#10
r0.94934 a
r0.614241 b
#20
r0.95349 a
r0.61424 b
.
.
.

```

Figure 32: Typical Telecom Power Trace file from Virtual Platform

It is a standalone tool and the integration into the overall workflow was entirely straightforward. As an example of the work being undertaken, the figure below represents the thermal profile of the Zynq platform at startup.



Figure 33: Zynq startup thermal profile

Then, the following figure represents a snapshot of the thermal profile of the same platform at full regime. As can be noted from the scales, the temperature ranges from around 24C up to circa 48C. (All intermediate points were also profiled, of course, but are not included here for reasons of space.)

Report on evaluation of the integrated design flow (final)

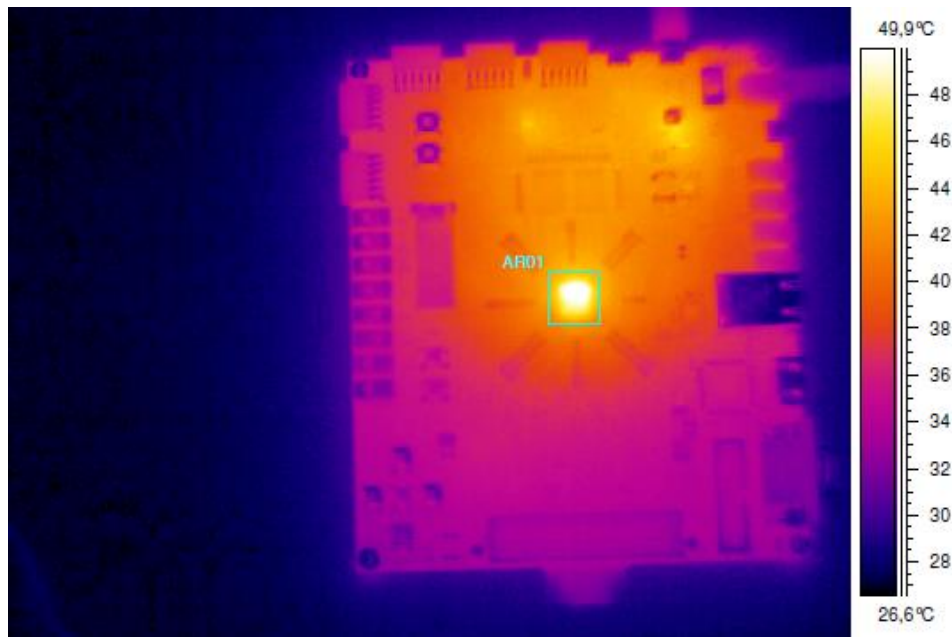


Figure 34: Zynq thermal profile at full regime

The tools were used on both the real platform and the simulation platform to verify the fidelity of the simulation, with satisfactory results. Below are simulated temperature maps and power/temperature graphs that were extracted from Intel® Docea™ tool (from left to right: 222 MHz, 333 MHz, 666 MHz, so increasing power and temperature). The maps are for the simulation timing of circa 600s, and they use the same temperature scale (from 25°C to 55°C) as the real temperature readings above to ease the comparison between the cases.

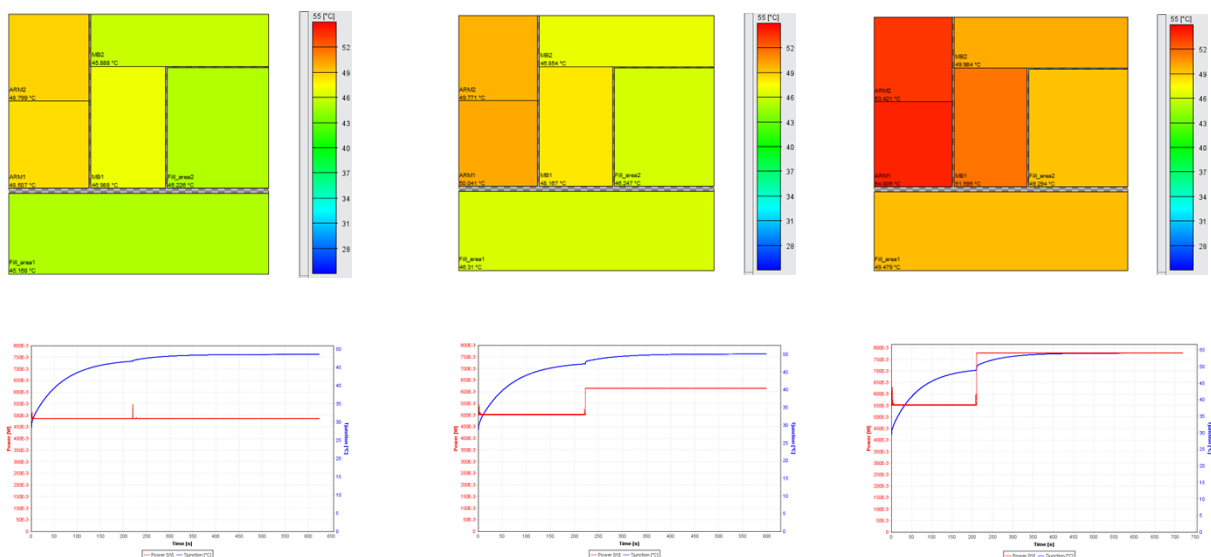


Figure 35: Zynq simulation thermal and power maps / graphs

*Report on evaluation of the integrated design flow (final)***Evaluation Summary**

This section summarizes the result of the evaluation.

Integration goal	ForSyDe SystemC version
Status	Completed, some DSE currently underway
Evaluation Results	The ForSyDe modelling tool and methodology were used to produce a complete, running model of the identified kernel subset of the Telecom application. There remains some Design Space Exploration activity to perform together with KTH personnel.

Integration goal	Virtual Platform
Status	Completed, in testing
Evaluation Results	<p>The telecom demonstrator application was successfully ported onto the virtual platform, and power traces were also successfully generated and analysed.</p> <p>The original nature of the application was such that it was not constructed for parallelism (since it was originally written for a single core processor, the PowerPC). For this reason, it became impractical to go through simulated load-balancing on the virtual platform to examine the effects in terms of power and in terms of thermal characteristics. Nevertheless, it was possible to circumvent this by varying other parameters, including the frequency scaling of the CPU, and thus be able to examine the effects on both the simulated platform (which was, of course, quite easy to implement) and on the real platform, which involved some intervention in the Zynq processor management.</p> <p>Overall, the comparison of simulated results and results on the real platform was strongly aligned, and the integration into the workflow was smoothly accomplished after clearing up the needs on both the simulated and the real-world platform environments.</p>

Report on evaluation of the integrated design flow (final)

Integration goal	HIFSuite
Status	Completed
Evaluation Results	<p>Intecs decided to use as its evaluation subject its network interface, which is a legacy component currently described in VHDL and needed to be integrated into the overall simulated application in order for a faithful simulation to be possible. It was abstracted first, and then HIFSuite was used to obtain the translation into SystemC++ (The original model was in VHDL). It was then successfully integrated into the overall simulation environment on the virtual platform by using a bridging element automatically generated by HIFSuite starting from the IP-XACT description of the legacy component.</p> <p>We were also able to obtain the necessary bridging elements automatically that were needed to do the co-simulation of the OVP part of the application and the part that was generated by HIFSuite. HIFSuite also generated these elements for us. Overall, no major obstacles were encountered over the whole process, and it integrates smoothly into the overall workflow. In particular, there are no major interferences between the HIFSuite workflow and the OVP workflow sub-branches, but rather they work together, thanks to the facilities for HIFSuite for supporting their interaction.</p>

Integration goal	Intel® Docea™ Thermal Profiler tool
Status	Completed, in use for testing purposes
Evaluation Results	<p>The Telecom Demonstrator personnel is using the Docea thermal tool to generate comparative power and thermal information both from the simulated power traces arriving from the virtual platform environment. A particular strength of the Docea tool is that it is aligned with the overall CONTREX methodology and, for example, is therefore able to process the power and thermal characteristics being managed not only in the real-world Zynq board but also the simulated environment, for a seamless integration into the overall methodology.</p>

6 References

- [1] "Description of Work". CONTREX –Design of embedded mixed-criticality CONTRol systems under consideration of EXtra-functional properties, FP7-ICT-2013- 10 (611146), 2013.
- [2] D.1.3.2
- [3] CONTREX Consortium, D5.1.1: Report on evaluation of modelling, design, validation, and platform service abstraction techniques (preliminary), 2015.
- [4] CONTREX Consortium, D5.2.1: Report on evaluation of design tools (preliminary), 2015.
- [5] CONTREX Consortium, D1.2.1: Definition of industrial use-cases, 2013
- [6] CONTREX Consortium, D2.3.2: System Modelling, Analysis and Validation Tools (final), 2016
- [7] CONTREX Consortium, D4.1.2: Implementation of demonstrator's applications (final), 2015
- [8] CONTREX Consortium, D4.2.2: Implementation of use-case execution platforms and run-time systems (final), 2015
- [9] Xilinx Inc. (2014) Zynq-7000 Platform Devices. [Online]. Available: <http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/index.htm>
- [10] Trenz Electronic GmbH. [Online] Available: <http://www.trenz-electronic.de>
- [11] Xilinx Inc. (2016) Extending 28nm Leadership. [Online]. Available: <https://www.xilinx.com/products/silicon-devices/28nm-extensions.html>
- [12] Embest (2016) Mars Board. [Online]. Available: <http://www.embest-tech.com/shop/star/marsboard.html>