Grant agreement no. 100265

Artemis Project

ASAM

Automatic Architecture Synthesis and Application Mapping

D1.3: Flow integration, verification and acceptance requirements.

Due Date of Deliverable	**1st November, 2011**
Completion Date of Deliverable	
Start Date of Project	**1st May, 2010 – Duration 36 Months**
Lead partner for Deliverable	**M. Lindwer (SH/IBBV)**
Author(s)	**G. Notarangelo (STM)**

Revision: v1.3

Project co-funded by the Artemis Joint Undertaking Call 2009		
Dissemination Level		
PU	Public	
PU/COA	Public with confidential appencices	X
PP	Restricted to other program participants (including Commission Services)	
RE	Restricted to a group specified by the consortium (including Commission Services)	
CO	Confidential, only for members of the consortium (including Commission Services)	

Table of Contents

Table of Figures

Table of Tables

1 Scope of this document

This document specifies the verification and validation strategy for the full flow of ASAM framework.

The validation plan should lead to operative feedback of the quality of the ASAM tools, their integration, and of the quality of the final results, based on the use-cases provided.

A further objective is to deliver industrial directives, specific to each use-case, covering the proposed design flow, such that ASAM methodology and tools can be validated. Application providers will concentrate their efforts on the development of tools, and/or integration of existing tools, referring to ASAM base technologies.

The follow up of this document will consist in:

- a continuous feedback provision, in order to keep a constant link between industrial and academic partners so as to ensure the applicability of the proposed methods and tools;
- improvement of the flow with mapping refinement in order to improve the results as obtained on the use-cases, if necessary.

This document does not include the characteristics of each tool and the structure of the use cases: this information is exhaustively reported in many of the first year deliverables.

All the developed in ASAM tools and toolchains will be just initial prototypes. They should thus provide the required functionality in the sense of producing their required input-output relationships and acceptable quality of results, and this will be checked. However, they will not be required to have any of the other features that are expected from the actual final commercial tools, as e.g. appropriate interfaces, reliability, documentation etc.

Therefore, even if further in some parts of this report not all needed details will be specified in relation to the above, it is assumed that this policy is the general policy of the Flow Verification and Acceptance Requirements.

Moreover, the formulated requirements will be subject to a careful review and possible refinement in the third year of the project, and will only be used for the actual flow verification and acceptance after their final acceptance by all the ASAM project partners.

2 Verification strategy

D1.2 “Final Design Methodology, Flow, and Tool Requirements” provides the specification and requirements for the ASAM design flow. The resulting framework covers a complete front-end design trajectory for multiprocessor systems: from high-level application target specifications to the mixed hardware/software implementation.

The implementation goal is to combine a complete set of tools, all properly interacting with each other, with the aim of, as much as possible within the project, automating the design space exploration that is needed to direct the design environments for building optimized multi-ASIP systems, where these processes currently comprise mostly manual tasks. The integration pertains in particular to the Intel / Silicon Hive multi-ASIP System-on-Chip design environment. However, the to-be-developed tools will not necessarily be specific for this environment.

This tool integration will also need a complete chain of several feedback loops, which establish logical and operative relationships among each of the different abstraction levels: from DSE targeted towards the HW/SW partitioning of the specific application, down to the final ASIC RTL, passing through the HDL generation and prototyping.

Besides specifying ASAM’s generic methodology and flow, document D1.2 also specifies several usage scenarios. These usage scenarios are defined to explicitly stipulate how the ASAM methodology should support user teams with different levels of expertise, and what results to expect from that level of support. Next to that, document D6.2 provides three application use cases. Each of the application use cases comprises application code, architecture platform, and current-day performance characteristics. This current document specifies for each application use case, what the result is expected to be when applying a certain usage scenario.

For example, the Default usage scenario for Novice Users specifies that applying the ASAM flow on a non-optimized application and platform, should result in a solution of which the quality level is on-par with 2010 state-of-the-art designs for the same application. Yet, if the flow is used by an experienced team, then these users should be able to quickly arrive at a solution which is on-par with 2013 state-of-the-art. This current document aims to quantify these statements for certain combinations of application use cases and usage scenarios.

Establishing completeness of the ASAM framework is another complex matter. As a consequence, the verification plan will have to follow common general rules, application independent but reproducible for each of the application use cases, in order to ensure that all aspects are covered consistently.

To cope with evaluation complexity, this document discusses a four-state verification approach, which will offer the possibility to progressively go from the entry point of an application use case, through the standalone use of the tools, till to a global flow vision composed by tools interaction and final result analysis (quality of results checkpoint).

This progressive approach, described in the following section, will provide both more clarity on the tool interaction mechanisms and a realistic metric on the development and implementation of the application use cases, which are the most relevant design flow experiments.

3 Four-state common verification flow.

Figure 3-1 shows a simple picture of the verification plan composed of four states. For each state, its elements of major interest are then explored in more detail.

The states are distributed in a sort of timeline logic flow and the feedback contribution chain is reported in terms of feedback branches for iteratively tuning both the application and the tools implementations and heuristics.

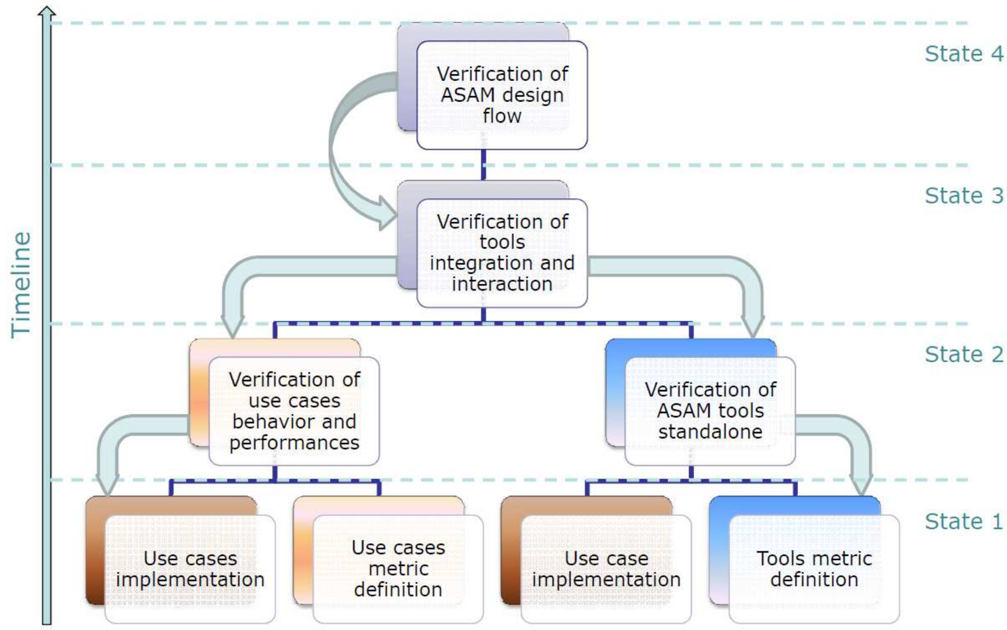

Figure 3-1 – Four-state verification strategy.

3.1 Verification strategy: state 1.

The verification of ASAM tools and design flow work-in-progress is, of course, tightly linked to application use cases, provided by industrial partners. However, the basic timeline should be application independent and reproducible for relevant applications.

Moreover for the different usage scenarios, the flow should lead to technically satisfying implementations of the application use cases.

Under these considerations, the first state of verification strategy consists of evaluating existing state-of-the art implementations of application use-cases, by using application-specific metrics. The objective of this phase is to evaluate the efficiency of the proposed solution and check whether it meets the system requirements provided in D6.2.

In this case, the metrics are values that can be monitored and that characterize the requirements of the implementation. When evaluating the application implementations, the metrics reflect the intrinsic performance of the design (ratio timing performance / power consumption for example); when evaluating the design tools, the metrics reflect the way in which today's design techniques converge to an adequate solution (number of exploration paths and parameters usable by the exploration tool).

The tables below describe those metrics, as proposed for the ECG, MPEG4 SP, and Hearing Aid application use cases. In following State 2, these metrics will be compared to the industrial results already provided in D6.2: "Definition of requirements, industrial use-cases and evaluation strategy", in order to verify the behaviour and the performance of these application use cases.

Essentially, State 1 is composed of two phases that are different but really linked to each other through the use-case implementation task. These phases are:

1. Application use case metric definitions: phase qualifying for each application use case the industrial objectives of the final implementation of the application;
2. Tool metric definitions: phase qualifying the quality of results of each tool, already existing or to be implemented, integrated into the ASAM flow.

ECG Metric	Description
Power Consumption	The power consumption of the overall circuit is the most important result to be controlled in a low-power system. Both for a complex systems such as the REISC SoC and for simpler platforms, a power-optimized system will have to include power saving mechanism able to manage the right system configuration fit to the application to map on. Power measures should be reported, where possible, outlining the real configuration mode in order to give an add value to the result.
Area occupation	In medical markets, cost often comes second to safety, dependability and performance. However, Less area means, in general, lower power (both static and dynamic). So this metric is closely related to the power consumption metric. REISC SoC is a relatively large design both for ECG or other medical applications. An area reduction of about 20%, by utilizing the results of the project, should be feasible.
Wake up time	Is related the above power consumption result metric. If a low-power system includes a power saving mechanism, there needs to be a fast wake up mechanism able to bring the power-switched parts of the system up as soon as possible. The power-on latency depends both on the applied switching technology (pMOS/nMOS switches) and on the nature of the low-power solutions. REISC SoC achieves a wake-up time of 1.1 µs: this is considered as a good result. If the latency were even lower, this would be very good. However, specific power switching features are not an objective of the ASAM project. Therefore, wake-up time is also not measured specifically.
Fault tolerance	A platform for medical applications has to be fault-tolerant. There is no simple measure for fault tolerance. The concepts include (i) robustness of the software, (ii) reliability of the technology, and (iii) redundancy of the system resources. However, most of these concepts are outside the scope of the ASAM project. Application software robustness is an aspect of the input specification. The project aims to be agnostic to the choice of middleware layers and to the choice of other implementation technologies. Redundancy is an architectural aspect and can, as a specification aspect/constraint be taken into account, at least in order to be able to compare other quality aspects based on comparable architectural constraints.
Real-time acquisition performance	Health monitoring only becomes viable once the system is able to acquire results in real-time. Real-time performance implies the capability, from the system, at least to capture data in the time range in which they are generated. This means an efficient streaming flow from the source to the buffer/memories where they will be stored

	waiting for the post-processing stage.

Table 3-1 - ECG Metrics.

MPEG4 SP Metric	Description
Power Consumption	The overall circuit power consumption is the most important result to be controlled in multi-processor systems, if targeted for the mobile market. Power optimizations take place both at the circuit level and at the architecture level. Specific power saving mechanisms which manage system power configurations based on application monitoring fit are part of the first category, which is not a specific target of the ASAM project, since the ASAM project deals with architectural improvements which reduce overall system power (regardless of the state of the application). Power measurements should be reported, if possible taking into account the actual configuration mode, in order to give an added value to the result. Moreover image/video compression has important applications in specific low-power fields (i.e: gastrointestinal endoscopy). With some simplified algorithms (and of course at the cost of lower compression ratio), it is possible to attain much lower energy dissipation. For a multimedia application mapped on a network architecture, power consumption is also linked to the memory access rate and the interconnect infrastructure between the processing elements.
Area occupation	In mobile and consumer markets, cost is one of the most important metrics, which is directly related to area occupation. In addition, lower area means, in general, lower power (both static and dynamic).
Real-time acquisition performance	Real-time performance is truly a system capability, and is of the highest importance in any video capture or video display application. When encoding video, it means that, data needs at least to be captured in the time range which they are generated: this means an efficient streaming flow from the source to the buffer/memories where they will be stored waiting for the post-processing stage.
Bus usage	In systems though for multimedia applications, the bus subsystem is one of the bottlenecks that limits the performance of the overall system. The communication through the system bus has to be reduced to a minimum level in order to keep the bus complexity and cost low, while keeping performance high.
Latency	In video and audio-oriented applications, one of the main factors to be taken into account is the latency of the transmission of a given communication context. For this reason, this parameter has to be maintained at a low level or hidden from intelligent data access mechanisms (at least within the requirements).
Throughput	As for the latency case, the throughput of the overall system (how much information can the system transmit in a given time) is one of the key parameters that will have to be optimized.
Scalability	An architecture design targeted for multimedia algorithms should be highly scalable to support a rich range of applications, including them that require standard formats and higher performances. Scalability is very hard to certify and measure, yet can to some degree be expressed through the number of applications supported with at their proper performance levels. Scalability of ASIC designs is very poor while processor based design typically shows high scalability.
Performance/power	The best tradeoff between performance and power consumption implies combining the flexibility and scalability of microprocessors, while providing the performance/power ratio comparable to that of ASIC designs.
Interconnect structure and	This represents a critical design parameter for high-density platforms. The basic topologies of processor interconnection include bus/ring, crossbar and mesh.

parameters of NoC or bus segmentation	Bus/ring architecture is simple, but its throughput is quite limited. Especially, when the node number is large, the throughput is not enough to maintain performance. Crossbar can ensure maximum bandwidth, and its architecture is simple. However, its implementation cost (area and power dissipation) increases exponentially with the number of processor nodes. Thus, it typically not used for larger processor networks. Mesh (NoC) and segmented bus are reasonable trade-offs between bus and crossbar: they ensure scalability but their architectures are more complex.

Table 3-2 - MPEG4 SP Metrics.

Hearing Aid Metric	Description
Power Consumption	In the hearing aid market, the overall circuit power consumption, including analog parts, is the most important metric. A dynamic power saving mechanism is important. However, it is not a topic of the ASAM project. And since a hearing aid must be always switched on and since, under certain circumstances, it must be guaranteed that no sound is lost, a hearing aid cannot be fully powered down.
Area occupation	Currently, cost is not one of the most important metrics in the hearing aid market. Several types of hearing aids exist, some of which fit within the human ear. The aesthetic aspects of hearing aids dictate that they are very small. Therefore, the total chip area cannot exceed 5 mm^2. Also, die size is often related to power consumption (both static and dynamic), the other main aspect of a hearing aid design.
Real-time acquisition performance	Real-time performance is truly a system capability, and is of the highest importance in a hearing aid application. When processing audio, it means that data needs to be captured and processed in the time range which they are generated; this means an efficient streaming flow from the microphones to speakers, through both analog and digital parts. The audio sampling rate has to be maintained under all circumstances.
Latency	In hearing aid applications, one of the main factors to be taken into account is the latency of audio processing. This latency should be very low, in order for the user to be able to respond immediately to sound impulses.
Throughput	As for the latency case, the throughput of the overall system (how much information can the system transmit in a given time) is one of the key parameters that will have to be optimized. Since the real-time capability needs to be maintained at all times, the throughput is determined by the complexity of the algorithms. On systems with high-performance audio processing capabilities, real-time processing can be guaranteed for complex algorithms, while maintaining high throughput.
Scalability	Scalability is an important feature for hearing aids, because the audio processing algorithms are constantly being updated. In addition, different users have different kinds of audio processing requirements. Therefore, audio DSPs have traditionally been applied extensively in hearing aids.
Performance/power	The best tradeoff between performance and power consumption implies combining the flexibility and scalability of microprocessors, while providing the performance/power ratio comparable to that of ASIC designs.

Table 3-3 – Hearing Aid Metrics.

3.2 Verification strategy: state 2.

The application use cases described in D6.2 (and mentioned above) are considered good targets for design flow experiments. As such, they are also considered invariant elements of the tool implementation. Depending on its structure and characteristics, each application use case targets a specific path in the design flow and focuses on a specific tool subset.

For example, the MPEG4 SP application use-case focuses on the high-level specifications, while the ECG and hearing aid application use cases more specifically focus on implementation and optimization steps. The questions to be answered in this step are:

1. Are the tools usable in an industrial environment?
2. Do they meet the user's needs?
3. Do they lead to required improvements?

To be usable in an industrial environment, tools have to follow the requirements of efficiency of output results, including portability of generated code.

Industrial partners therefore have to evaluate their performance and their degree of adaptation to the industrial needs. To enable a precise and fair evaluation, all partners have participated in the definition of a set of metrics applicable to each tool and capable to measure its performance and quality. Specifically, as the complete list of tools will be defined, all tools metrics will be accompanied by associated baselines which should be references to quantify the quality of results.

State 2 checks also the time to obtain the DSE results on application use case behaviour and performance under standalone tool runs.

Both tools and application use case verification actions in state 2 will provide the final evaluation in tuning application use case implementation and re-definition of tools metrics per case.

3.3 Verification strategy: state 3.

State 3 is essentially based on the verification of tool integration.

Also in this state, the application use cases are the invariant elements. However, the main action is focused on common tool interfaces and communication. This state also verifies the operative delay time in making tool communication available during the entire flow (changeover of input data formats, definition of max number of feedback loops from one tool to another during the optimization phase, and so on). However, as defined in the technical annex and as a consequence of the fact that the ASAM final tools are prototypes, some inter-tool communications can require the final user intervention and be not automatic.

3.4 Verification strategy: state 4.

The last step of the approach consists of verification of the integrated design flow and calculating the improvement it brings compared to current design practices.

As the flow is, at present, only partially implemented, this state will not be developed here. Indeed, State 4 can be detailed only when all tools and the flow will be available to all partners.

4 Verification Strategy and Evaluation Criteria based on ISO/IEC 9126.

The verification method should be integrated in the evaluation activity based on the BS ISO/IEC 9126-1:2001 and integrated in the Task T6.3: "*Evaluation of design flow and its particular tools*".

Industrial use-cases, whose development relies on the use of the proposed tools and methodology, are checked against the key properties identified for each tool family. The evaluation aims also at verifying that the proposed flow follows industrial standards, e.g. in terms of adhering to standards for RTL languages, and provides real improvement over actual design practices.

In this document, a first attempt of requirements on the implementation design is produced classifying the requirements by application use cases. The tables should be modified according to the results of use-case implementation and they will be re-proposed with the final structure in the D6.5: "*Evaluation of the design flow.*"

4.1.1 Re-iteration of the objectives of the project

This section briefly repeats the set of high-level and technology-level objectives, as they are stated in the technical annex of the ASAM project. In subsequent chapters, a complete set of application use case requirements/criteria is provided, which covers each of below project operational, functional, technical, and verification requirements.

High-level goals:

- Integrated chain of tools
- Complete process flow for development of embedded systems
- Support architecture exploration
- Supporting concurrent validation and verification at different abstraction levels
- Constructs heterogeneous computing architectures instantiated from generic platform
- Flexibility and adaptability to trade off performance, resources, and power usage
- Generated platforms enable massive real-time data processing
- Enable composition of platform-independent software on concurrent systems
- Relevant for a broad range of applications: consumer, multi-media, telecom, imaging

Derived from these high-level goals, the technical annex provides a number of detailed operational targets, functional requirements, technical objectives, and verification requirements. These are listed here below. The list contains identification tags for future reference to these targets and requirements.

<u>Operational targets</u> are (not all simultaneously):

O1. Reduce productivity gap for designing ASIP-based embedded systems, by reducing effort levels for system design, relative to 2005, by 10%. In D1.2, this requirement was made more strict and more specific, through the specification of a set of usage scenarios:

O1.1. Using the (as much as possible) automated flow: a novice team of engineers should achieve results that are on-par with 2010 state-of-the-art.

O1.2. Using an automated flow, but possibly exercising some manual controls: an expert team of engineers should achieve results which are on-par with 2013 state-of-the-art, yet with 25% lower effort.

O2. Or the ASAM results enable an expert team to manage an application complexity increase of 25% with similar effort levels as when using traditional flows.

O3. Or the ASAM results enable solution efficiency improvement of 20% to 30% in terms of power, area, or performance (PPA). In D1.2, this requirement was made more specific:

O3.1. Using manual controls, an expert team of engineers should exceed by 20% the 2013 state-of-the-art, at similar effort levels as when using traditional flows.

Functional requirements are:

F1. Input is an embedded system specification with the following qualifications/components:

F1.1. high-level,

F1.2. algorithmic,

F1.3. constraints (PPA numbers and implementation technology)

F2. Output is an embedded system with the following qualifications/components:

F2.1. correctly executing the algorithmic input specification,

F2.2. heterogeneous,

F2.3. ASIP-based,

F2.4. satisfying the input constraints

Detailed technical objectives:

T1. Technology-aware, rapid, multi-objective design space exploration, taking into account micro- and macro system trade-offs

T2. Estimators will be used to drive fast design space exploration

T3. Identify or construct a common input formalism for parametric requirement definitions for embedded systems

T4. Identify or construct common system platform template formalism

T5. Design space covers full range of parameters of multi-ASIP systems, including physical parameters

T6. The impact of the floorplanning on the power consumption distribution on the die will be evaluated

T7. Automatically construct and synthesize application-tailored system architecture

T8. System and processor analysis for performance, area, and power consumption

T9. Support purely hardware-implemented system modules

T10. At micro-architecture level, synthesize ASIPs, accelerators, communication structures, and memory structures, by identifying, generating, and instantiating IP blocks, such as:

- sequences of operations to be replaced by new instructions,
- parallel issue slot clusters, register files, memories,
- and other datapath elemements

T11. Synthesize power islands that allow the switch-off of parts of the processing and storage elements[1].

[1] In the absence of requested funding for TUBS, this requirement cannot be fully met.

T12. Identify or construct a common input formalism for application behavior, which is standard-based, near to industry practice, and have powerful (a. o. abstract) constructs to express parallelism

T13. Support application analysis, parallelization, and partitioning, by analyzing how application domain specific computation and communication patterns interact in the interconnections at the runtime

T14. (Semi-)automatic application mapping of embedded software, originating from a single source, on the resulting platform

Verification requirements:

V1. Design technology will be demonstrated and evaluated using standard and generated benchmarks

V2. Demonstration will be based on (parts of) real-life industrial applications provided by the consortium's industrial partners.

V3. Showing applications running in (near) real-time on actual implementation platforms (Because of practical and commercial constraints, the target system cannot have the same complexity as a complete product.)

V4. prove that the proposed new technology is capable of producing working systems, showing mapping both hardware and software systems to existing FPGA boards

V5. resulting systems can be constructed with less effort or with higher quality (i.e. improved PPA) than when using traditional methodologies

   a. SH and ST will measure how much effort currently is being spent on designing similar systems; methodology will be used to design an ultra low-power hearing aid platform. Area and power consumption should be 20% to 30% lower than state-of-the-art

   b. methodology will be used to design a high-performance video codec platform decoding HD streams within the power budget of a mobile phone.

V6. The accuracy of results of estimators will be compared against the results of more traditional measurement systems[2]

V7. the flow documentation will be checked on timely availability and quality

V8. the prototype models will be checked for existence of their code, functionality, and quality of their execution

V9. Demonstration on prototype tools of the applicability and effectiveness of the design methodology, flow, and tools.

4.1.2 Application Use Case I: ECG System Requirements and Evaluation Characteristics.

In Table 4-1 the System Requirements for the design of the ECG application use case are listed. They are aligned with the main industrial features for a low-power platform.

In this phase of the project, the System Requirements have to be considered a pure proposal and can be modified, cancelled, replaced and/or confirmed in the final ASAM evaluation plan.

A brief rationale on how the requirement is applicable to the ASAM design flow is also included.

System Requirement	Applicability to the ASAM Design flow
SR-01 (relates to T3+T6+T12): Must be aligned with the pre-existent power	The success and exploitation of new methods and tools depends on the ability to answer the design's needs that are not covered by existing design

[2] E.g. full-blown power simulation at netlist level.

	format related to different abstraction levels.	flows. A new architecture explorer and ASIP implementation tool only generates real interest from semiconductor companies, primarily through the possibility to be easily integrated in the existing industrial design flows. The ASAM flow should therefore respect and facilitate the RTL-to-layout flow. In particular, if ASAM is to generate power control mechanisms, there are already formats to express power-saving techniques in the design flow, like UPF (Unified Power Format) and CPF (Common Power Format). In that case, having the same format to represent power-mechanism, shall be an objective, taking into account constructing a UPF/CPF representation in the RTL model of the SoC.
SR-02	(relates to T11): May allow representation of certain aspects of existing implementations of power control mechanisms.	Power mechanisms are already well exposed in D6.1 and D6.2 when describing the Reisc SoC architecture which includes the ECG application use case global definition and properties. The hardware representation of general industrial power mechanisms at all levels of abstraction (retention states, multi-supply voltage techniques, dynamic voltage and frequency scaling techniques, power domains, power gating clock gating etc.), may also be described at the macro and micro architecture level.
SR-03	(relates to T13+T14) Must provide an efficient SW representation of application blocks.	Since Silicon Hive's Software Development Kit (HiveCC) includes a full ANSI-C compiler, a well defined methodology that allows mapping onto micro and macro-architecture models a pure functional C model is required.
SR-04	(relates to T9): Must allow the use of predefined HW blocks.	When dealing with generic platform exploitation and related design generator (as ASAM should be), at the interface between synthesis final steps of the implementation flow, it shall be possible to replace independent architecture nodes by already existing IP models within the resource library, even if modelled, for example, only in RTL.
SR-05	(relates to F2.1+T2+T8+V4+V6): Must provide accurate results, as specified in the application use cases.	The accuracy of characterization results is necessary to take the appropriate decisions in all usage scenarios (power dissipation for power management implementation, for example). The accuracy and reliability of the results have to be as expected in the requirements and have to allow the correct analysis of each application use case.
SR-06	(relates to F1.3+F2.4+T1): Must be able to take into account the characteristics of the technology.	When adding power/timing information to the HW generated block, all tools should be driven by generic parameters reflecting the characteristics of the technology (90nm, 65 nm, etc.).
SR-07	(relates to T7+T10+T12): Should use and follow open standards.	A platform generator fully compliant with open standard (i.e.: SystemC TLM 2.0) could be a plus good feature for industrial flow inheritability and should be considered a truly IP reuse follow up. The open standard compliance requires a heavy effort that should be evaluated in the next timeline review.
SR-08	(relates to V2+O1+O2+O3+T4+T5+T8+V5): Must provide a complete simulation analysis of the whole system.	ASAM environment should be able to support the simulation of communication engines among embedded parts of the system since this aspect is significant in the assessment of the performance of the design solutions and therefore in the design-space exploration and optimisation.
SR-09	(relates to T6+T11): May provide characterization of power modes for DSE framework and runtime management.	Since (or if) the best solution of low power HW platform for ECG application will provide different power modes, exploration take into account power management strategies. Impact of software-driven power management strategies may be considered during power consumption measurement.
SR-10	(relates loosely to T4+T5+T10): Memory subsystem exploration phase should include on-chip and off-chip memory models.	The memory hierarchy modelling and memory access profiling developed in design exploitation and implementation should consider dimensioning of the whole memory subsystem, both on-chip and off-chip. For a low power use-case definition (as ECG) the use and traffic exploitation with memory subsystem is a key point for power dissipation final results.

SR-11	(relates to SR-01, T3+T6+T12) Should provide standard interface implementation.	The interfaces between the flow and the application use case and between tools in the flow have to be as standard as possible. Using tools built to standards, which are non-proprietary, increases the ability to incorporate those tools into a diverse array of new or existing flows and minimizes the risk of "data lock-in" in the future. For more details on the interface definition between tools refer to D1.2.
SR-12	(relates to O1+O2): Flow should have a proper simulation time.	In particular the system environment should have an order of magnitude gain of efficiency in term of simulation time respect to RTL level. This is crucial for Design Exploration step, where many simulations have to be performed.
SR-13	(loosely relates to O1+O2): Flow may be able to perform a RTL to a generic behavioural language transformation.	SR-04 requirement can lead to improper simulation time if RTL blocks are present. To avoid this, RTL blocks may be replaced, in an easy and low-effort way, by representations in a generic behavioural language
SR-14	(related to O1.1) Novice users should be able to achieve 2010 state-of-the-art results	Using ASAM's largely automated flow and starting from non-optimized code and system design, novice users should be able to achieve the same power and performance results as mentioned in ASAM deliverable D6.2, section 2.9.
SR-15	(related to O3.1) Expert users should be able to exceed 2010 state-of-the-art results by 20%.	Starting from 2010 state-of-the-art, expert users should be able to design a system that has 20% better PPA numbers (either power, performance, or area 20% improved) than mentioned in ASAM deliverable D6.2, section 2.9.

Table 4-1 – Application Use Case I: ECG requirements.

4.1.3 Low Power Industrial Evaluation Strategy.

Provided application use cases, and in particular the ECG low-power application, rely on a physical implementation of the platform that can allow evaluating hardware and software timing, as well as, power information provided by ASAM flow with respect to RTL and Netlist simulations.

In particular a Low-Power RTL-to-GDSII Flow is implemented in STMicroelectronics and illustrated in

3.

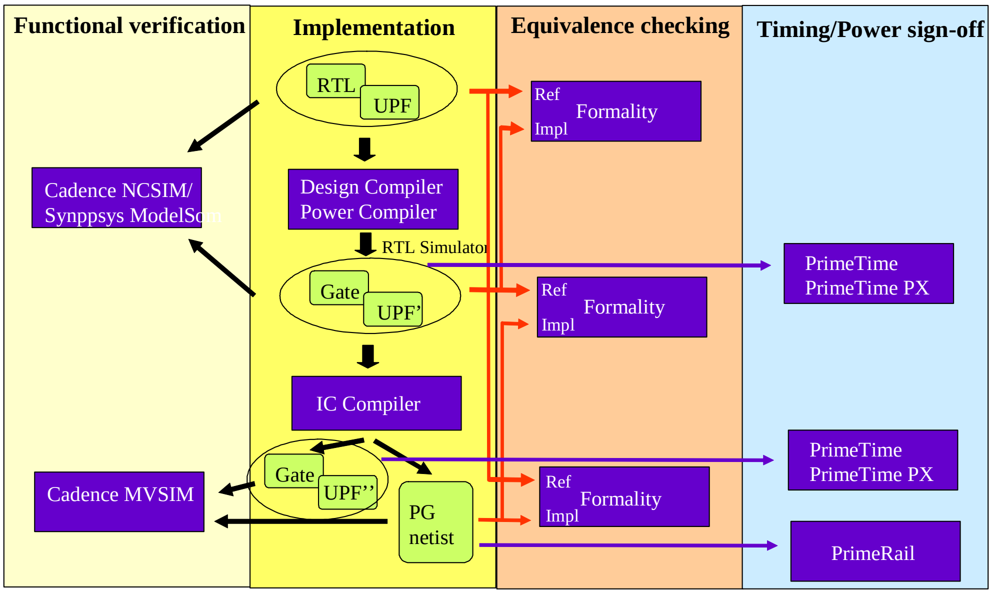

Figure 4-3 - STMicroelectronics RTL to GDS-II flow

The ASAM flow will provide tools and methods to evaluate performance and power consumption of the system in order to adapt applications to the hardware platform and find the best architecture of the complete system. As a consequence, reliability of evaluations is the main factor to evaluate in order to validate the efficiency of the proposed flow.

To check the validity of timing performance and power measurements provided by ASAM tools, D6.1 and D6.2 already provided the Reisc SoC physical system results on ECG application mapping according to a set of different low-power operative modes.

Although STMicroelectronics doesn't provide the implementation board and test-chip to the consortium, during the evaluation phase STMicroelectronics is willing to check its own results versus ASAM framework evaluation board/simulator in a coherent way, in order to have a right measures comparison.

4.1.4 Application Use Case I: ISO 9126 factors to evaluate for ECG application.

The Table 4-2 summarizes the evaluation strategy of the software and hardware produced by the ASAM tools and methods or (of the software implementing the ASAM tools) with respect to the ISO 9126 characteristics and sub-characteristics taking into account the requirements on the design flow described in previous sections.

Please refer to ASAM deliverables D1.2 ("Final Design Methodology, Flow, and Tool Requirements") and D6.1 ("Definiton of requirements, industrial use-cases, and evaluation strategy") for elaborate explanation of the different aspects of the ISO9126 quality framework. Since this framework is specifically meant for software which is produced completely under industrial/commercial usage requirements, not all of the characteristics are completely relevant to the results of the ASAM project.

Deliverable D1.2 therefore also indicates where the ISO9126 characteristics are different from the ASAM requirements. Where this is the case, in below table, the cell in column "Requirement" will state "N/A" or "not traced". When the cell states "N/A", this indicates that the requirement is not applicable to the ASAM project results, either because the requirement is not relevant to the kind of tooling that ASAM produces, or because the requirement is not applicable to typical hardware systems. When the cell states "not traced", this means that the requirement would be applicable to the ASAM tools, if those tools would need to be commercially viable. However, since the results of the ASAM project are prototypes originating from a research project, these requirements are relevant to a much lesser extent. Thus, the associated metrics are, many cases not specifically measured.

Unless explicitly specified differently, below requirements relate to ASAM's tools, not to the results generated by these tools.

.

	ISO 9126 Characteristic	ISO 9126 Description of Characteristic		ISO 9126 Sub-characteristics	ISO 9126 Description of Sub-characteristics	Requirement	Rationale
1	Functionality	The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.	1,1	Suitability	The capability of the deliverables (prototype tools) to provide an appropriate set of functions for specified tasks and user objectives.	SR-02 SR-04 SR-08 SR-09 SR-10 SR-14 SR-15	The tools might provide or generate system simulation, power modes, runtime memory management and memory subsystems. These features are indeed generally used to reduce power consumption in autonomous sensor node systems. However, they are not part of the kind of architecture improvements from which the ASAM project expects to extract much higher gains.
			1,2	Accuracy	The capability of the software product to provide the right or agreed results or effects with the needed degree of precision.	SR-05 SR-09 SR-10	Accuracy can be measured by comparison with RTL and netlist models provided or generated for characterized implementation technologies.
			1,3	Interoperability	The capability of the software product to interact with one or more specified systems.	SR-01 SR-11 SR-07	The use of open standards facilitates guarantees this interoperability between different software tools

			1,4	Security	The capability of the software product to protect information and data so that unauthorised persons or systems cannot read or modify them and authorised persons or systems are not denied access to them.	N/A	No security requirements.
			1,5	Functionality Compliance	The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions relating to functionality.	SR-01	ASAM framework should take care of pre-existing de-facto standards for power supply information in SoC architecture.
2	Reliability	The capability of the software product to maintain a specified level of performance when used under specified conditions.	2,1	Maturity	The capability of the software product to avoid failure as a result of faults in the software.		The ASAM framework toolset shall be stable to the extent that the three application use cases can be executed under at least one of the usage scenarios.
			2,2	Fault Tolerance	The capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.	Not traced	Workarounds for known points of failure should be available.
			2,3	Recoverability	The capability of the software product to re-establish a specified level of performance and recover the data directly affected in the case of a failure.	Not traced	If a failure occurs while designing the system, models shall be recoverable and information about the failure kept in a log file.
			2,4	Reliability Compliance	The capability of the software product to adhere to standards, conventions or	SR-05 SR-08	In the case of ASAM tools, reliability is strongly related to accuracy of complete system

					regulations relating to reliability.		analysis. Only if system-level characterisation results are accurate, users can rely on the DSE results. When analysis is accurate, the DSE tools should then be reliable to take the right decisions.
3	Usability	The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.	3,1	Understandability	The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use.	N/A	Outputs from ASAM framework toolset will be clear, and possibly compliant to pre-existent outputs. Analysis from DSE should provide clear information regarding best trade-off.
			3,2	Learnability	The capability of the software product to enable the user to learn its application.	Not traced	Documentation of ASAM framework toolset should be clear and complete.
			3,3	Operability	The capability of the software product to enable the user to operate and control it.	Not traced	Different usage scenarios require different levels of operability. When operated by expert users, the tools shall provide detailed controls to steer their operation. When operated by novice users, the tools shall execute mostly automatically.
			3,4	Attractiveness	The capability of the software product to be attractive to the user.	N/A	N/A

			3,5	Usability Compliance	The capability of the software product to adhere to standards, conventions, style guides or regulations relating to usability.	SR-01 SR-03	Where applicable, the use of open standards implies this usability aspect.
4	Efficiency	The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions.	4,1	Time behaviour	The capability of the software product to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions.	SR-12	Virtual and emulated systems should have simulation times (several) orders of magnitude less) than RTL models.
			4,2	Resource Utilisation	The capability of the software product to use appropriate amounts and types of resources when the software performs its function under stated conditions.	Not traced	ASAM framework toolset will be tested on a computer farms in order to check the efficiency in a typical industrial framework. The measurements will assume single Intel I7 (2.5GHz) machines.
			4,3	Efficiency Compliance	The capability of the software product to adhere to standards or conventions relating to efficiency.	N/A	N/A
5	Maintainability	The capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in	5,1	Analysability	The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for identification of the parts, which need to be modified.	N/A	Tools will be supplied with debug symbols. Exceptions will be thrown with proper source code references. Information on tool execution and potential failures will be kept in a log file.
			5,2	Changeability	The capability of the software product to enable a specified modification to be implemented.	SR-04 SR-06 SR-13	Replacing an HW IP with a pre-existent one or modifying it and its parameters according to different technology should be an easy operation for the

		environment, and in requirements and functional specifications.					ASAM framework user.
			5,3	Stability	The capability of the software product to avoid unexpected effects from modifications of the software.	N/A	No changes are expected to be performed in the ASAM framework toolset.
			5,4	Testability	The capability of the software product to enable modified software to be validated.	N/A	No changes are expected to be performed in the ASAM framework toolset.
			5,5	Maintainability Compliance	The capability of the software product to adhere to standards or conventions relating to maintainability.	N/A	No changes are expected to be performed in the ASAM framework toolset.
6	Portability	The capability of the software product to be transferred from one environment to another.	6,1	Adaptability	The capability of the software product to be adapted to different specified environments without applying actions or means other than those provided for this purpose to the considered software.	SR-11 SR-01	The configuration of the ASAM framework toolset via open format files and scripting languages will facilitate this sub-characteristic. The effort needed should be previously defined.
			6,2	Installability	The capability of the software product to be installed in a specified environment.	Not traced	Standard platforms (i.e. certain Linux distributions) will be specified on which the tools will be tested. The easiness of the installation of the ASAM framework toolset will be evaluated.
			6,3	Co-existence	The capability of the software product to co-exist with other independent software in a common environment sharing common resources.	N/A	N/A

			6,4	Replaceability	The capability of the software product to be used in place of another specified software product for the same purpose in the same environment.	SR-11	The use of open standards facilitates the replaceability of a particular tool by a similar one performing the same function.
			6,5	Portability Compliance	The capability of the software product to adhere to standards or conventions relating to portability.	SR-11 SR-01	Standard platforms (i.e. certain Linux distributions) will be specified on which the tools will be tested. The use of open standards/existent format for output reinforces portability in different environments.
7	Safety	The capability of the software product to meet applicable safety requirements.		Output Integrity	The capability of the software product to supply outputs which are safe.	Not traced.	The repeatability of the results is a potential issue. The tools depend on genetic algorithms for new design space points. The final result depends on initial population. However, we should expect comparable improvements, regardless of the differences.
				Qualifiability	The capability of the software product to be qualified from a safety point of view.	Not traced.	N/A

Table 4-2 - ISO 9126 characteristics for ECG application.

4.1.5 Use Case II: MPEG4 Simple Profile System Requirements and Evaluation Characteristics.

In the Table 4-3 the System Requirements on the design for the MPEG4 encoder application are listed. They are aligned with the main industrial features for a high-performance platform.

In this phase of the project, the System Requirements have to be considered a pure proposal and can be modified, cancelled, replaced and/or confirmed in the final ASAM evaluation plan.

A brief rationale on how the requirements are applicable to the ASAM design flow is given here: The design flow proposed by the ASAM project deals both with hardware and software parts of a design. The goal is to offer the possibility to find the best mapping between application and platform. As an application provider, STMicroelectronics will concentrate its efforts on the development of the application and its mapping onto the platform.

Following STMicroelectronics' industrial executable specification is the entry of the flow. This part is highly important, because it is at the interface between algorithms description and mapping on the platform.

Requirement	Applicability to the ASAM Design flow
SR-16 (loosely related to F1+F2+T12+V2): Must allow access to shared data and to the application code.	The application might need to access some shared constants or arrays of reference for example. As a consequence, it is important to design a way to access shared data from the software side. Accesses may be implicit or explicit, but a clear method to access data located in memory from the software side should be defined.
SR-17 (loosely related to T1): Should allow access to data located in the heap or stack.	If code integrated in a module is definitely intended to be implemented in CPU software (as opposed to ASIP software), we would have the possibility to use dynamic memory allocation as in pure software. We need to control the growth of the stack and heap and spread data in different memory sections when implemented on the final device.
SR-18 (loosely related to F1+F2+T12) Should provide a methodology to ensure ANSI-C compliance.	Application developers usually develop applications in pure ANSI C. Because, even if optional, some specific ANSI-C extensions need to be used to achieve code which is optimally targeted to the platform (vector operations, HiveCC pragmas, etc., see D1.2 and D5.1).
SR-19 (loosely related to F1+F2+T12): Should use standard C tools.	To create a C specification that can be handled by the ASAM framework tools, designers will most probably have to rely on specific libraries. However, libraries developed in the scope of the project should not restrict the use of commercial ANSI-C tools.
SR-20 (loosely related to T12): Should use last ANSI-C compiler (gcc) version.	The majority of the tools of the market offer the possibility to handle designs based on gcc kernel latest version. This makes ASAM developers to be able to use the latest debugging facilities.
SR-21 (related to T12+T13): Must provide a way to express parallelism in the application SW.	This will be directly offered by ASAM tools, but we need to be aware of any restriction on the number of potential parallel threads. We also need to know if the number of parallel threads would have any impact on the efficiency of

	the timing annotation tool.
SR-22 (related to F2.3+F2.4+T2+T8+T13): Should provide a clear analysis during HW/SW task separation.	The entry of the MPEG4 encoder application is composed of several signal processing algorithms that might be mapped onto hardware accelerators in order to reduce the load on CPUs and ASIPs. The "HW/SW task separation" step of ASAM framework should provide an analysis of the potential implementations: hardware or ASIP software or CPU software. This analysis should integrate precise metrics such as timing performance and power consumption.
SR-23 (related to F2.2+T7+T10): May integrate automatic HW generation.	Studying two implementation variants by developing all components from scratch would be too expensive. As a consequence, the methodology proposed may integrate a synthesizer to implement software elements in hardware. The intervention of the designer should be reduced as much as possible. This is not part of the ASAM project proposal.
SR-24 (related to F2.2+F2.3+T10+T13+T14+V2): Must provide communication management.	The implementation of a task into hardware has to integrate the management of communications. Indeed, exchange of data between hardware and software elements has to respect the communication protocol defined by the hardware platform. The separation of hardware and software tasks should propose a way to handle communications efficiently by proposing for example an API of functions that can be used by the designer.
SR-25 (loosely related to T11+T13+T14): May support control of power management by software.	Since the evaluation study platform of the MPEG4 encoder application provides different power (see D6.2) contribution visibility, exploration may focus on power management strategies. Implementation of SW power management would in that case be a requirement.
SR-26 (loosely related to O1+O2): Should change the structure of code as little as possible.	Generated/transformed code shall resemble the original as much as possible. The overall structure (top-level functions, global variables, modules of functions etc.) should be preserved and should be recognisable in output code.
SR-27 (related to F1.3) Must allow using pre-characterized technologies.	When adding power information to the HW generated block, it is important that the tools are driven by the characteristics of the hardware technology (ASIC process node, FPGA, etc.).
SR-28 (related to O1+O2+F1.1+T14): Should provide tracking of code transformations.	When it is not possible to preserve the original structure of the code, as required in SR-26, and transformations are needed, the generated code should give information to the developer on the structure and how the original structure has been modified. It should be, for example, possible to track the input and output of functions.
SR-29 (related to O1+O2+T2+T7): System generator should be autonomous.	One advantage of ASAM platforms is the ability to use them without any cost expensive hardware: evaluation board, development board, JTAG connectors, etc. To favour exploitation, the generated virtual platform should be able to run in an autonomous way. It should be possible to run the simulation on a PC without attached specialised hardware components.
SR-30 (related to O1+O2+T8+V3): Virtual system generator should guarantee simulation performance.	The performance of the simulation should be sufficient to offer the possibility to debug applications. Simulation time should be efficient enough to allow many simulations in reasonable time.
SR-31 (related to F2.1+F2.4+T2+T8+V3+V4+V6):	All quality-related measurements, such as power, area, and performance will be estimated with different trade-offs between accuracy and

Should guarantee an adequate level of information.	analysis/simulation speed, as needed for a particular stage of the ASAM flow. The project is mostly concerned with power reduction through architectural features of the ASIPs, their constituent components and other architectural blocks, than with explicit power management, which is not directly in the scope of the project. The ASAM framework will analyze the implications of power management, by assuming that certain power control features are in place and adjusting power estimates accordingly. The most detailed quality-measurements will be sufficiently accurate such that users can determine whether the results of the ASAM flow satisfy the usage scenarios for this application use case. ~~The flow will automatically improve the performance of the system in terms of power without affecting its performance. The optimizations should be detailed to the user by printing the different power reduction techniques implemented and the optimization of the software code.~~
SR-32 (related to F1.3+F2.4+T5+V6): Should show parameter's effects.	It is also important for the user to observe directly the impact of the modification of one parameter. As a result of changing a parameter, quality-related measurements, such as power, area, and performance can be estimated with different trade-offs between accuracy and analysis/simulation speed, as needed for a particular stage of the ASAM flow.
SR-33 (related to O1.1) Novice users should be able to achieve 2010 state-of-the-art results.	Using ASAM's largely automated flow and starting from non-optimized code and system design, novice users should be able to achieve the same power and performance results as mentioned in ASAM deliverable D6.2, section 3.2.
SR-34 (related to O3.1) Expert users should be able to exceed 2010 state-of-the-art results by 20%.	Starting from 2010 state-of-the-art, expert users should be able to design a system that has 20% better PPA numbers (either power, performance, or area 20% improved) than mentioned in ASAM deliverable D6.2, section 3.2.

Table 4-3 – MPEG4 encoder usage scenario requirements.

4.1.6 High Performance Industrial Evaluation Strategy.

The application of MPEG4 Simple Profile provided by STMicroelectronics is a part of the standard video compression algorithm. This application use case intends to utilize hardware and software modelling and estimation parts of the ASAM design flow.

The ASAM framework should offer a way to find the best mapping between application and platform. As an application provider, STMicroelectronics will concentrate its efforts on the development of the application and its mapping onto the platform, while other partners will support this application use case by the providing configurable, executable, SW/HW platforms.

These two aspects, Hardware/Software partitioning and efficient application to virtual platform mapping, are crucial in the evaluation process. Moreover, based on reference numbers provided by the virtual platform provided for this application, it will be possible to compare the consistency of the ASAM flow estimated power and timing properties with the virtual behaviour.

4.1.7 MPEG4 encoder application: ISO 9126 factors to evaluate.

The Table 4-4 summarizes the evaluation strategy against the ISO 9126 characteristics and sub-characteristics taking into account the requirements on the design flow described in previous sections.

	ISO 9126 Characteristic	ISO 9126 Description of Characteristic		ISO 9126 Sub-characteristics	ISO 9126 Description of Sub-characteristics	Req.	Rational
1	Functionality	The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.	1,1	Suitability	The capability of the software product to provide an appropriate set of functions for specified tasks and user objectives.	SR-16 SR-17 SR-18 SR-21 SR-23 SR-24 SR-25 SR-28 SR-30 SR-33 SR-34	Memory management and memory subsystems are key features. Equally important are ANSI-C SW entry and support for analysis of parallel SW and effects of design decisions/parameters on quality aspects of the design. Explicit power management is important.
			1,2	Accuracy	The capability of the software product to provide the right or agreed results or effects with the needed degree of precision.	SR-31 SR-22 SR-30	Accuracy can be measured by the amount and quality of information provided by ASAM toolset.
			1,3	Interoperability	The capability of the software product to interact with one or more specified systems.	SR-20 SR-29	The use of open standards at the appropriate version facilitates interoperability between different software tools.
			1,4	Security	The capability of the software product to protect information and data so that unauthorised persons or systems cannot read or modify them and authorised persons or systems are not denied access	N/A	No security requirements.

					to them.		
			1,5	Functionality Compliance	The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions relating to functionality.	SR-20	ASAM should use standards.
2	Reliability	The capability of the software product to maintain a specified level of performance when used under specified conditions.	2,1	Maturity	The capability of the software product to avoid failure as a result of faults in the software.	Not traced	The ASAM toolset shall demonstrate stability, albeit it is only a prototype toolset.
			2,2	Fault Tolerance	The capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.	Not traced	If a failure occurs while designing the system, models shall be recoverable and information about the failure kept in a log file.
			2,3	Recoverability	The capability of the software product to re-establish a specified level of performance and recover the data directly affected in the case of a failure.	Not traced	If a failure occurs while designing the system, models shall be recoverable.
			2,4	Reliability Compliance	The capability of the software product to adhere to standards, conventions or regulations relating to reliability.	N/A	N/A
3	Usability	The capability of the software product to be understood learned, used and	3,1	Understandability	The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use.	SR-26	The output code should be as human readable as possible in order to favour maintainability and debugging.

		attractive to the user, when used under specified conditions.	3,2	Learnability	The capability of the software product to enable the user to learn its application.	Not traced	ASAM flow has to propose a very sharp learning curve: main aspects of the flow should be learned in few weeks.
			3,3	Operability	The capability of the software product to enable the user to operate and control it.	Not traced	Two aspects should be considered: ASAM should simplify the use of HW platform and help the designers and architects to configure HW platforms.
			3,4	Attractiveness	The capability of the software product to be attractive to the user.	N/A	N/A
			3,5	Usability Compliance	The capability of the software product to adhere to standards, conventions, style guides or regulations relating to usability.	SR-20	The use of open standards enforces this usability of the tool.
4	Efficiency	The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions.	4,1	Time behaviour	The capability of the software product to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions.	SR-30	Multiple simulations of the use case shall be performed in reasonable time.
			4,2	Resource Utilisation	The capability of the software product to use appropriate amounts and types of resources when the software performs its function under stated conditions.	SR-29	
			4,3	Efficiency	The capability of the software product to adhere to standards or conventions	N/A	N/A

				Compliance	relating to efficiency.		
5	Maintainability	The capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.	5,1	Analysability	The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified.	N/A	N/A
			5,2	Changeability	The capability of the software product to enable a specified modification to be implemented.	SR-27	Replacing an IP with a different technology should be possible.
			5,3	Stability	The capability of the software product to avoid unexpected effects from modifications of the software.	N/A	No changes are expected to be performed in the ASAM toolset.
			5,4	Testability	The capability of the software product to enable modified software to be validated.	N/A	No changes are expected to be performed in the ASAM toolset.
			5,5	Maintainability Compliance	The capability of the software product to adhere to standards or conventions relating to maintainability.	N/A	No changes are expected to be performed in the ASAM toolset.
6	Portability	The capability of the software product to be transferred from one environment to another.	6,1	Adaptability	The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered.	SR-19	The configuration of the ASAM toolset via open format files will enforce this sub-characteristic.
			6,2	Installability	The capability of the software product to be installed in a specified environment.	Not traced	The easiness of the installation of the ASAM toolset will be

							evaluated.
			6,3	Co-existence	The capability of the software product to co-exist with other independent software in a common environment sharing common resources.	N/A	N/A
			6,4	Replaceability	The capability of the software product to be used in place of another specified software product for the same purpose in the same environment.	Not traced	The use of open standards enforces the replaceability of a particular for a similar one performing the same functions.
			6,5	Portability Compliance	The capability of the software product to adhere to standards or conventions relating to portability.	Not traced	The use of open standards/existent format for output reinforces portability in different environment.
7	Safety	The capability of the software product to meet applicable safety requirements.		Output Integrity	The capability of the software product to supply outputs that are safe.	Not traced.	The repeatability of the results should be a proof.
				Qualifiability	The capability of the software product to be qualified from a safety point of view.	Not traced	Repeatability an accuracy of the results should guarantee a qualified product.

Table 4-4 - ISO 9126 characteristics for MPEG4 Simple Profile usage scenario evaluation.

4.1.8 Use Case III: Hearing Aid application use case: System Requirements and Evaluation Characteristics.

In Table 4-5 the System Requirements for the design of a hearing aid application are listed. They are aligned with the main industrial features for an ultra low-power special-purpose DSP platform.

In this phase of the project, the System Requirements have to be considered a pure proposal and can be modified, cancelled, replaced and/or confirmed in the final ASAM evaluation plan.

Silicon Hive / Intel provided this application as an example of a typical industrial application. The application is provided as a non-optimized straight-line software implementation, characterized on a typical embedded RISC processor. The application is also provided as consisting of optimized code, mapped onto an appropriately configured/optimized DSP platform. Silicon Hive / Intel will monitor the capabilities of the ASAM flow by applying it on the non-optimized application. Silicon Hive / Intel will verify that the flow achieves results that are similar to the supplied optimized platform.

Requirement	Applicability to the ASAM Design flow
SR-35 (related to O1.1+O1.2+T12+T13): Must process straight-line ANSI-C code.	The application code is provided as regular non-optimized ANSI-C code. The application analysis phase of the project must autonomously partition such code, according to different processing characteristics found in the code.
SR-36 (related to T12): Should use last ANSI-C compiler (gcc) version.	The majority of the tools in the market offer the possibility to handle designs based on Gnu Compiler Collection (GCC). The ASAM flow should incorporate functional verification of application code, based on compiling (and executing) applications using the latest GCC version. This gives the ASAM developers the possibility to use the latest debugging facilities for initial straight-line ANSI-C development.
SR-37 (related to F2.2+F2.3+T2+T8+T13+T14): Must provide a clear analysis to drive optimization strategies for code partitions	The hearing aid application is composed of several signal processing algorithms that might need to be mapped onto different types of accelerators or other architectural components, in order to improve processing efficiency. The macro-level design space exploration step of ASAM framework should provide an analysis of the potential implementations: hardware or ASIP software or CPU software.
SR-38 (related to O1.2): Should process partially optimized ANSI-C code and suggest further improvements.	The application code is also provided as optimized ANSI-C code, expressing different kinds of parallelism and mapped onto an optimized platform. The ASAM flow should be able to process this code and, if opportunities exist to further optimize this code, the flow should identify these opportunities or indicate to the experienced user that these opportunities might exist. The flow should also assist the experienced user in evaluating possible further optimizations.
SR-39 (related to O1.1+O1.2+O2+F2.2+F2.3+T4+T7) Must automatically generate an appropriate macro architecture design.	In order to map partitioned applications, the ASAM flow must generate those micro architectural components that are suitable to map the different partitions on. The ASAM flow must generate a macro architecture which instantiates those components.
SR-40 (related to O1+O3+F2.2+F2.3+T1+T5+T8+T10): Must analyze and potentially optimize	The ASAM flow must provide analyses about the performance, power, and area characteristics of the different micro architectural components, when they execute their assigned application partitions. These

architectural components.	application partitions may originate from initial straight-line code or from –partially- optimized code. These analyses must be sufficiently accurate to allow comparison of different possible micro architectural optimizations and to analyze the effects of such optimizations.
SR-41 (related to O1.2+O3+T4+T7+T8): Must analyze and potentially optimize macro architecture.	The ASAM flow must provide analyses about the performance, power, and area characteristics of macro architectures, which instantiate (optimized) micro architectural components. These analyses must be sufficiently accurate to allow comparison of different possible macro architectural optimizations and to analyze the effects of those optimizations.
SR-42 (related to T4+T5+T7+T9): Macro-level analyses should take into account the characteristics of ‘external’ components.	In the hearing aid application use case, piezoelectric devices (microphones, speakers) and AD/DA converters consume a certain amount of system-level area and power. These devices are not generated by the purely digital ASAM flow. However, the area and power impact of these devices does partially depend on the processing characteristics of the application (e.g. sample rate). Even though the ASAM flow cannot influence the characteristics of these devices, when analyzing overall system-level PPA characteristics, the influence of such components on maximum achievable parallelism should be taken into account (Amdahl’s law)
SR-43 (related to F1.3+F2.4+T1+T3+T5+T6): Should support power-aware technology choices and strategies synthesis	Hearing aids are always on devices. During operation, the device continuously processes the flow of incoming sound. Therefore, the choice of silicon implementation technology should result in the lowest leakage current possible. In order to reduce dynamic power consumption, the flow should use implementation technology and synthesis strategies which make use of very low voltages, low clock speeds, and specific (clock) duty cycles. On the other hand, the current optimized application does not make use of dynamic power control mechanisms and for this application, this is also not expected of the ASAM flow.
SR-44 (related to F2.2+F2.3+T10+T13+T14+V2): Must provide communication management between different architectural components.	Even when the complete hearing aid application is mapped onto a single ASIP, the flow will have to model the communication behaviour between the ‘external’ components and the ASIP. Indeed, exchange of data between such ‘external’ hardware and software elements has to respect specific communication protocols. When the application is mapped onto several ASIPs, the communication between these ASIPs has to be facilitated as well. The separation of hardware and software tasks should propose a way to handle communications efficiently by proposing for example an API of functions that is automatically used to facilitate different application partitions and that can also be used by the designer to implement communication with ‘external’ components.
SR-45 (loosely related to O1+O2): Should change the structure of code as little as possible.	Generated/transformed code shall resemble the original as much as possible. The overall structure (top-level functions, global variables, modules of functions etc.) should be preserved and should be recognisable in output code.
SR-46 (related to F1.3) Must allow using pre-characterized technologies.	When adding power information to the HW generated block, it is important that the tools are driven by the characteristics of the hardware technology (ASIC process node, FPGA, etc.).
SR-47 (related to O1+O2+F1.1+T14): Should	When it is not possible to preserve the original structure of the code, as

provide tracking of code transformations.	required in SR-45, and transformations are needed, the generated code should give information to the developer on the structure and how the original structure has been modified. It should be, for example, possible to track the input and output of functions.
SR-48 (related to O1+O2+T2+T7): System generator should be autonomous.	One advantage of ASAM platforms is the ability to use them without any cost expensive hardware: evaluation board, development board, JTAG connectors, etc. To favour exploitation, the generated virtual platform should be able to run in an autonomous way. It should be possible to run the simulation on a PC without attached specialised hardware components.
SR-49 (related to O1+O2+T8+V3): Virtual system generator should guarantee simulation performance.	The performance of the simulation should be sufficient to offer the possibility to debug applications. Simulation time should be efficient enough to allow several seconds of real-time operations to be simulated in reasonable time, i.e. at most one hour of simulation time.
SR-50 (related to F2.1+F2.4+T2+T8+V3+V4+V6): Should guarantee an adequate level of information.	All quality-related measurements, such as power, area, and performance will be estimated with different trade-offs between accuracy and analysis/simulation speed, as needed for a particular stage of the ASAM flow. Since the project is mainly concerned with power reduction through architectural features of the ASIPs, their constituent components, and other architectural blocks, explicit power management is not directly in the scope of the project. The ASAM framework will analyze the implications of power management, by assuming that certain power control features are in place and adjusting power estimates accordingly. The most detailed quality-measurements will sufficiently accurate such that users can determine whether the results of the ASAM flow satisfy the usage scenarios for this application use case. The flow will automatically improve the performance of the system in terms of power without affecting its performance. The optimizations should be detailed to the user by printing the different power reduction techniques implemented and the optimization of the software code.
SR-51 (related to F1.3+F2.4+T5+V6): Should show parameter’s effects.	It is also important for the user to observe directly the impact of the modification of one parameter. As a result of changing a parameter, quality-related measurements, such as power, area, and performance can be estimated with different trade-offs between accuracy and analysis/simulation speed, as needed for a particular stage of the ASAM flow.
SR-52 (related to O1.1+F2.4+V5+V6): When used by novice users, effort should be 20% less than using traditional methods.	ASAM Deliverable 6.2 reports that effort level for a single novice user to achieve a 2010 state-of-the-art design and mapping was 2 person-months. Using the new ASAM flow, this should be reduced to 7 man-weeks. Refer to D6.2 for PPA numbers of 2010 state-of-the-art.
SR-53 (related to O1.2+O3.1+F2.4+V5+V6) When used by expert users, 20% additional complexity should be managed.	ASAM Deliverable 6.2 requires that it should take expert users 2 man-months to extend the optimized version of the application with a feature such as Acoustic Scene Classification or with sampling frequency increased to 20 KHz. The ASAM flow should construct a new optimized application and system design to include this feature, yet still fitting the same PPA budget as 2010 state-of-the-art.

Table 4-5 – Hearing Aid application use case requirements.

4.1.9 Ultra low-power Industrial Evaluation Strategy for hearing aid application.

The application provided by Silicon Hive / Intel is a complete hearing aid DSP software stack. In terms of its functionality, it was verified to be consistent with state-of-the-art designs. Detailed power and area estimation experiments (using industrial EDA tools) were published and were verified to exceed state-of-the-art for these kinds of designs. This application use-case is intended to show that novice users can achieve state-of-the-art designs using the largely automated ASAM design flow.

4.1.10 Hearing Aid application use case: ISO 9126 factors to evaluate.

The Table 4-6 summarizes the evaluation strategy against the ISO 9126 characteristics and sub-characteristics taking into account the requirements on the design flow described in previous sections.

	ISO 9126 Characteristic	ISO 9126 Description of Characteristic		ISO 9126 Sub-characteristics	ISO 9126 Description of Sub-characteristics	Req.	Rational
1	Functionality	The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.	1,1	Suitability	The capability of the software product to provide an appropriate set of functions for specified tasks and user objectives.	SR-35 SR-38 SR-39 SR-40 SR-41 SR-44 SR-47 SR-48 SR-51	Memory management and memory subsystems are key features. Equally important are ANSI-C SW entry and support for analysis of parallel SW and effects of design decisions/parameters on quality aspects of the design. Explicit power management is important.
			1,2	Accuracy	The capability of the software product to provide the right or agreed results or effects with the needed degree of precision.	SR-37 SR-39 SR-43 SR-47 SR-50 SR-51	Accuracy can be measured by the amount and quality of information provided by ASAM toolset.
			1,3	Interoperability	The capability of the software product to interact with one or more specified systems.	SR-36 SR-43 SR-44	The use of open standards at the appropriate version facilitates interoperability between different software tools.
			1,4	Security	The capability of the software product to protect information and data so that unauthorised persons or systems cannot read or modify them and authorised	N/A	No security requirements.

					persons or systems are not denied access to them.		
			1,5	Functionality Compliance	The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions relating to functionality.	SR-35 SR-43 SR-46	ASAM should use standards.
2	Reliability	The capability of the software product to maintain a specified level of performance when used under specified conditions.	2,1	Maturity	The capability of the software product to avoid failure as a result of faults in the software.	Not traced	The ASAM toolset shall demonstrate stability, in the limit possible for a prototype toolset.
			2,2	Fault Tolerance	The capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.	Not traced	If a failure occurs while designing the system, models shall be recoverable and information about the failure kept in a log file.
			2,3	Recoverability	The capability of the software product to re-establish a specified level of performance and recover the data directly affected in the case of a failure.	Not traced	If a failure occurs while designing the system, models shall be recoverable.
			2,4	Reliability Compliance	The capability of the software product to adhere to standards, conventions or regulations relating to reliability.	N/A	N/A
3	Usability	The capability of the software product to be understood	3,1	Understandability	The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and	SR-37 SR-38 SR-45 SR-47	The output code should be as human readable as possible in order to favour maintainability

		learned, used and attractive to the user, when used under specified conditions.			conditions of use.	SR-50	and debugging.
			3,2	Learnability	The capability of the software product to enable the user to learn its application.	SR-52	ASAM flow has to propose a very sharp learning curve: main aspects of the flow should be learned in few weeks.
			3,3	Operability	The capability of the software product to enable the user to operate and control it.	SR-39 SR-40 SR-41 SR-42 SR-45 SR-47	Two aspects should be considered: ASAM should simplify the use of HW platform and help the designers and architects to configure HW platforms.
			3,4	Attractiveness	The capability of the software product to be attractive to the user.	N/A	N/A
			3,5	Usability Compliance	The capability of the software product to adhere to standards, conventions, style guides or regulations relating to usability.	SR-35 SR-36	The use of open standards enforces this usability of the tool.
4	Efficiency	The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated	4,1	Time behaviour	The capability of the software product to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions.	SR-48 SR-49 SR-52 SR-53	Multiple simulations of the use case shall be performed in reasonable time.
			4,2	Resource Utilisation	The capability of the software product to use appropriate amounts and types of resources when the software performs its function under stated conditions.	SR-48 SR-49	

		conditions.	4,3	Efficiency Compliance	The capability of the software product to adhere to standards or conventions relating to efficiency.	N/A	N/A
5	Maintainability	The capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.	5,1	Analysability	The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified.	N/A	N/A
			5,2	Changeability	The capability of the software product to enable a specified modification to be implemented.	SR-53	Parameterization of the implemented application should be supported (i.e. adding algorithm extensions or increasing the sample rate).
			5,3	Stability	The capability of the software product to avoid unexpected effects from modifications of the software.	N/A	No changes are expected to be performed in the ASAM toolset.
			5,4	Testability	The capability of the software product to enable modified software to be validated.	N/A	No changes are expected to be performed in the ASAM toolset.
			5,5	Maintainability Compliance	The capability of the software product to adhere to standards or conventions relating to maintainability.	N/A	No changes are expected to be performed in the ASAM toolset.
6	Portability	The capability of the software product to be transferred from	6,1	Adaptability	The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for	SR-38 SR-42 SR-51 SR-53	The configuration of the ASAM toolset via open format files will enforce this sub-characteristic.

		one environment to another.			this purpose for the software considered.		
			6,2	Installability	The capability of the software product to be installed in a specified environment.	Not traced	The easiness of the installation of the ASAM toolset will be evaluated.
			6,3	Co-existence	The capability of the software product to co-exist with other independent software in a common environment sharing common resources.	N/A	N/A
			6,4	Replaceability	The capability of the software product to be used in place of another specified software product for the same purpose in the same environment.	Not traced	The use of open standards enforces the replaceability of a particular for a similar one performing the same functions.
			6,5	Portability Compliance	The capability of the software product to adhere to standards or conventions relating to portability.	Not traced	The use of open standards/existent format for output reinforces portability in different environment.
7	Safety	The capability of the software product to meet applicable safety requirements.		Output Integrity	The capability of the software product to supply outputs that are safe.	Not traced.	The repeatability of the results should be a proof.
				Qualifiability	The capability of the software product to be qualified from a safety point of view.	Not traced	Repeatability an accuracy of the results should guarantee a qualified product.

Table 4-6 - ISO 9126 characteristics for Hearing Aid application use case evaluation.

5 Conclusions

The current document provides a comprehensive strategy to verifying all aspects of the proposed ASAM methodology and flow for building and programming digital SoC systems for embedded applications.

Since the ASAM project will result in a flow which consists of a set of tools, the strategy consists of four stages, related to several levels of integration of the ASAM toolset.

At the last stage, the three industrial application use cases are applied according to three different usage scenarios. For each of the application use cases, a specific set of usage criteria is provided. These usage critera are cross referenced with the project requirements and with the applicable ISO9126 common software quality criteria.

When combined, the usage criteria for the three different application scenarios cover all operational, functional, technical, and verification requirements for the project. They also cover all applicable ISO9126 software quality criteria.