

# FPGA-Based Development for Sophisticated Automotive Embedded Safety Critical System

Thang Nguyen  
Infineon Technologies Austria AG

Stuart Wooters  
TRW Automotive

## ABSTRACT

As software (SW) becomes more and more an important aspect of embedded system development, project schedules are requiring the earlier development of software simultaneously with hardware (HW). In addition, verification has increasingly challenged the design of complex mixed-signal SoC products. This is exacerbated for automotive safety critical SoC products with a high number of analogue interfaces (sensors and actuators) to the physical components such as an airbag SoC chipset. Generally, it is widely accepted that verification accounts for around 70% of the total SoC development. Since integration of HW and SW is the most crucial step in embedded system development, the sooner it is done, the sooner verification can begin. As such, any approaches which could allow verification and integration of HW/SW to be deployed earlier in the development process and help to decrease verification effort, (e.g.: accelerate verification runs) are of extreme interest.

In the described context, this paper addresses not only the design and verification challenges of such embedded systems but also proposes a new development, verification and validation workflow using an *FPGA-based SoC Emulation System with synthesizable analogue functional stubs and a risk minimizing analogue test chip*, which emulates and partially implements respective mixed-signal behavior of the ASIC SoC hardware. The workflow can also be called FPGA-in-the-Loop (FiL) and it is proven to help multi-disciplinary development teams to vastly improve the quality of HW/SW co-design at system level with as little time and effort as possible.

**CITATION:** Nguyen, T. and Wooters, S., "FPGA-Based Development for Sophisticated Automotive Embedded Safety Critical System," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 7(1):2014, doi:10.4271/2014-01-0240.

## INTRODUCTION

### **Modern Automotive Airbag System Application**

The increasing number of airbags in a vehicle, the requirement to comply with stricter safety requirements while costs must be reduced has forced automotive airbag system design to a new SoC design approach. As shown in [Figure 1](#), typically a modern airbag system is a heterogeneous sophisticated system comprised of sensors, an airbag ECU (Electronic Control Unit) - which consists of an airbag SoC chipset and an airbag embedded main micro-controller ( $\mu$ C) - and actuators. The airbag ECU is the hardware within a multiple airbag system that controls the deployment of airbags within a car.

During operation, sensors such as buckle switches, accelerometers or pressure sensors, mounted in key locations of the vehicle, continuously measure the positions of impact, the severity of the collision and other variables. The measured information is provided to the airbag SoC chipset in form of analogue communication signals, e.g.: modulated current, via the sensor interfaces. The airbag SoC chipset translates the analogue sensor communication, which is reported to the main  $\mu$ C via SPI (Serial Peripheral Interface) interface or other busing architectures. Based on this information the airbag ECU makes a determination if there is a crash event and if confirmed will activate the appropriate squib channels. This is done by providing a specified current and duration through the firing element resulting in deployment.

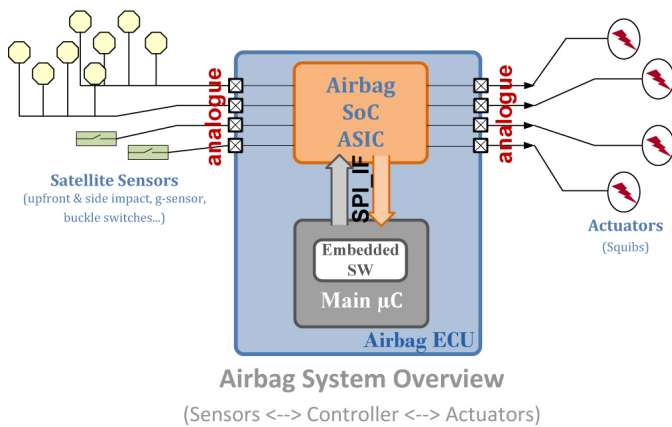


Figure 1. A typical airbag system from bird's eye view

### The Airbag SoC HW Functionalities and Design Complexity

As with all electronic systems, SoC hardware architectures consist of processing elements (embedded processors or state machines), IO devices, storage elements and interconnection structures linking all elements together. According to [1],[2],[3],[4] and [5] the most fundamental characteristic of an SoC is its complexity. Shown in Figure 2, integrating an automotive airbag electronic system into a single SoC controlled by a main  $\mu$ C makes the airbag SoC a complex device. The main functionalities of the airbag SoC chip include the following:

- Intelligent supply management units consist of Switched Mode Power Supply modules - SMPS (buck and boost converter), the internal Power Supply Generator - iPSG, and a CENTRAL power supply control logic, which controls the chip power up and power down stage and monitors the system supply integrity. It also provides logic for failsafe state diagnostics in case of integrity violation.
- Remote sensor interfaces supporting different sensor types ranging from acceleration to rotational sensors and compliance to various standards such as PSI5 (Peripheral Sensor Interface) [6] or DSI3 (Distributed System Interface) [7].
- Safing engine assisting the main  $\mu$ C in observing the sensor data transmission and confirming if a deployment event has occurred.
- Airbag deployment module - squib drivers and deployment enabling logic - allowing current for the ignition of airbags, covering requirements on deployment timing and current as well as load condition diagnostics.

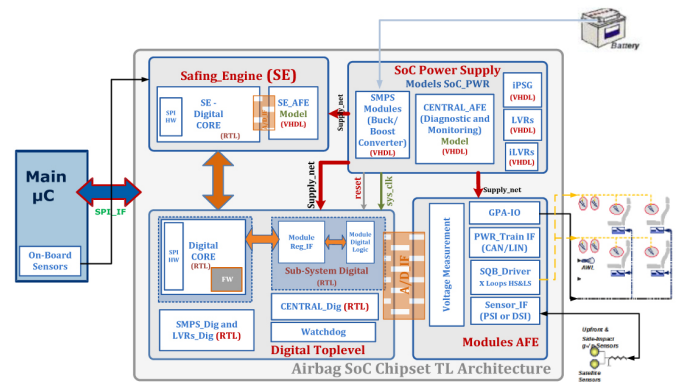


Figure 2. Overview of the airbag SoC chipset top-level implementation architecture

### CHALLENGES IN VERIFICATION, INTEGRATION, AND VALIDATION OF LATEST AUTOMOTIVE EMBEDDED SYSTEM DESIGN

Although airbag application systems are not new, the high integration of a large amount of functionalities into one single device with stricter safety requirements while still being cost-effective has brought the airbag system design to a new height of innovation. In addition, new sensor communication protocol developments have driven completely new platform development (in both HW/SW, e.g.: HW - airbag SoC chipset and SW - embedded SW of the main  $\mu$ C) to design teams. Especially, the verification of such a safety critical system becomes a real challenge mainly because of:

- Verification for the airbag SoC HW has to cover *real-time embedded mixed signal* domains.
- Most of the SoC functionalities can only be verified at the *top level* of the chip including many of the requested interactions requiring the embedded SW. The classical mixed-signal simulation approach becomes a bottle neck due to simulation performance issues.
- For new sensor protocol development, many verification scenarios of the sensor interfaces, such as, *long-term verification run* with checking of *millions of sensor data frames* are not suitable using computer-based simulation.
- As well, *reducing time-to-market and right-the-first-time* design in automotive electronics industry - one of the key requirements in projects to win customer and market share - have posed a great challenge to the design and verification team.

## PAPER MOTIVATIONS AND CONTRIBUTIONS

Generally, it is widely accepted that verification accounts a significant effort of the total SoC development. Additionally, since integration of HW and SW is the most crucial step in embedded system development, the sooner it is done, the better. However, there are many techniques to integrate hardware and software. The main goal should be to not waste time on debugging good software on faulty hardware or debugging good hardware running with faulty software [3],[5]. As such, the main motivation of the paper is to focus on approaches which could help:

- decrease verification effort, e.g.: accelerate verification runs or reuse of test setup between different disciplinary work groups
- allow verification and integration of HW/SW to be deployed earlier in the development process

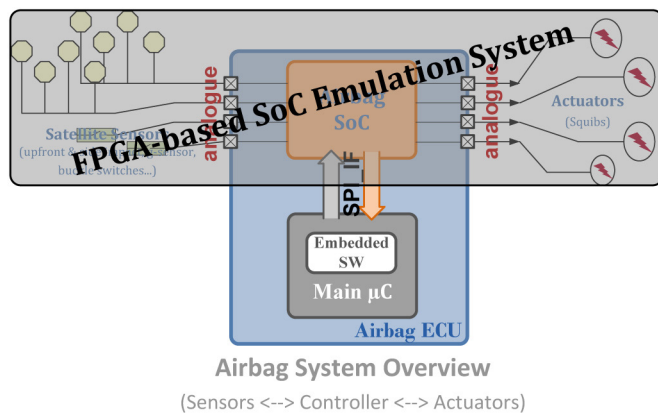


Figure 3. FPGA-based SoC Emulation System

As such, the paper contributes to the following:

- Presents the design of an embedded FPGA with synthesizable analog stub functions and analogue test chip, so called *FPGA-based SoC Emulation System*. As shown in Figure 3, the emulation system allows *not only* emulation of the mixed-signal behavior of the airbag SoC product *but also* the sensor and actuator hardware behavior.
- Proposing an innovative workflow for automotive embedded system/platform development, which uses the developed *FPGA-based SoC emulation system* as a vehicle to allow HW and SW to be tested/verified during their development cycle. The new workflow allows not only early system integration and validation but also acceleration of HW and SW verification.

## THE FPGA-BASED SOC EMULATION SYSTEM WITH ANALOGUE MIXED-SIGNAL TEST CHIP

Advanced development of emulation and FPGA technology in recent years allows high logic integration, thus early mapping of the design into an FPGA device is possible [8],[9]. One of

the main challenges in using FPGAs to map the complete system application is the capability of synthesizing analogue interfaces or analogue circuitry. Although, there have been “mixed-signal” FPGAs available in the market, such as SmartFusion from Microsemi, these devices are still limited for applications with sophisticated analogue interfaces. In order to tackle the challenge of integrating AFE circuitries into the emulation system, with minimum effort and implementation time, the authors applied the following two strategies:

- For relative simple AFE circuitries synthesizable models (implemented using VHDL) of these analogue circuits were created, called *stubs-functions*. These models are abstractions of the analogue HW functionalities which are necessary for verifying the design implementation of the SoC HW digital logic. These models have additionally a control input interface (*Stub-Ctrl*) which can be controlled by user of the system (e.g.: test engineers, SW developers, etc...) via a software GUI (graphical user interface), illustrated in Figure 4. As a result, different analogue HW statuses, including a faulty status (Analogue HW fault emulation), could be generated using the functional stub model with the control GUI. This strategy gives the advantage of having a low implementation effort in a reasonable time while it could be used not only for verification of HW implementation but also for the embedded SW development team. The main drawback is that the AFE stub-function model could only cover a sub-set of the HW functionality. This could become a risk for sophisticated and first time implemented circuits, e.g.: new sensor interface AFE circuits, as more critical mixed-signal and timing requirements need to be verified and validated.
- The motivation of the second strategy focuses on solving the limitation described in the first strategy. The entire AFE circuit (complex and first time) is designed in the typical flow but then implemented on a test chip, which is integrated together with the FPGA containing the RTL of the whole chip. This is also called a *risk minimizing analogue test chip (analogue TC)* as after the chip is validated, it will be migrated back to the SoC HW design team for final integration. Although, this strategy requires higher effort, its value realized as the complete implementation of certain important functions of the chip can be validated before the final chip is assembled and produced. Figure 4 illustrates setup of the airbag system emulation, focusing on verification and validation of the sensor data communication path, which is one of the safety critical paths in the design (Any errors occur during the sensor data transmission could consequence to endanger the driver life.) As such the second strategy has been used to support the AFE development, verification and validation of new sensors and transceiver.