

AUTOSAR Based Multicore Software Implementation for Powertrain Applications

Ralph Mader, Armin Graf, and Gerd Winkler
Continental Automotive GmbH

ABSTRACT

The combustion engine will be the dominant drive for motor vehicles despite all the advances in the electrification of the drive train, for many years. The greater are the challenges for the automotive industry, especially in fuel consumption (CO₂) and the environmental impacts of other emissions. From the fuel supply to the engine, up to the exhaust after treatment, new or improved functions are needed, which are integrated into increasingly powerful control electronics. This modern electronic engine management and powertrain controller will remain key components in the vehicle.

As most of the micro controllers for future applications will be MultiCores, this paper gives an overview on how PowerSAR® supports this kind of architectures. It shows the concepts applied in the basic software area as well as for the applicative software. Further it will show the impact on the development process as well as the integration support for software delivered by the OEM.

This work will give an outlook on upcoming micro controller architectures and the new challenges in the future in this domain.

CITATION: Mader, R., Graf, A., and Winkler, G., "AUTOSAR Based Multicore Software Implementation for Powertrain Applications," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 8(2):2015, doi:10.4271/2015-01-0179.

MULTICORE: A CHALLENGE FOR THE ENGINE MANAGEMENT APPLICATION

Performance Needs

The main drivers for innovation and new functionalities in the engine management domain are first the legislation demands for low fuel consumption and clean air and second the consumer demand for fun-to-drive. Continuously stricter exhaust gas limits - such as EU 6c or EU 7 - combined with downsizing activities of the engines are driving the push for more and more computing power for functionalities implemented in software. Closed-loop control strategies, with an online evaluation of the opening and closing behaviour of the injector needle, are only one example. An end to this trend is not in sight. As current technology has already reached a high level of complexity, the effort to improve fuel consumption and emission is much greater than in the past. This will also impact the required computational power of the micro-controllers used in this area.

Needed Versus Available Performance

In the past, the performance of micro-controllers was mainly increased by operating the core at a higher clock speed. A micro-controller in 2003, when 32-bit architecture started to penetrate the market, was clocked with 40 MHz. Meanwhile, the clock frequency has now reached 300 MHz and the silicon structures decreased from 130nm down to 40nm.

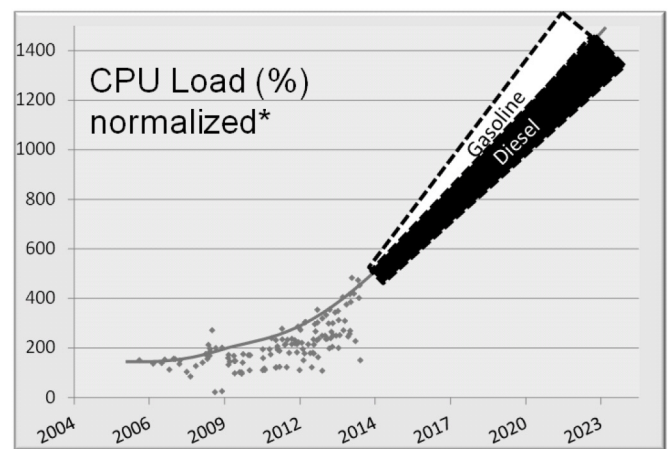


Figure 1. CPU utilization of high-end engine management applications from 2005 to 2023

Figure 1 shows the evolution of the needed performance. Each dot in the graph gives the used CPU load of a high-end application. High-end in the domain of engine management application is not necessarily equal to a higher number of cylinders. Even three or four cylinder combustion engines with a complex air path and injection system are considered in this class. 100% gives the reference, with a 100 loaded Freescale MPC563 @56MHz (Green Oak). Today's high-end applications are using approximately 4.5 times more computing power.

300 MHz seems to be also the limit for the silicon suppliers in terms of power losses of the device. Increasing the clock speed in today's micro controller designs beyond 300 MHz would require other cooling concepts, which would increase the cost for the ECU and set new constraints for their placement in the vehicle. This would not be accepted in the market in addition to questions being raised concerning technical feasibility. Therefore the answer to the request for additional computing power is the introduction of additional cores on the controller; while keeping the clock frequency constant or even decreasing it.

Micro-Controller Architecture

The micro-controllers, which typically used in engine management context, have two to three cores that can independently execute software. The cores share the same instruction set. All the controllers have shared flash memory, with size scales from 2.5 MB to 8 MB for code. The RAM size varies from 192 KB to 576 KB and is divided into system RAM - connected to the crossbar and core local memory for data and code, to increase the performance of the cores [1]. At least one of the cores is equipped with a lock step core dedicated to support safety requirements.

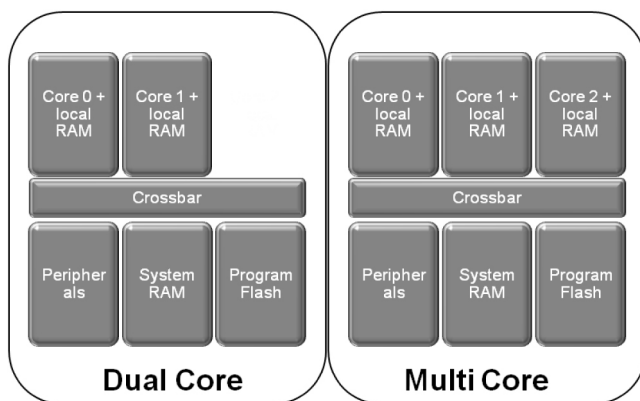


Figure 2. Used Micro-controller architecture

Consequences for the Software Architecture

The performance estimations for the future shows (see Figure 1), that every high-end engine management application with a production start of 2015 onwards will have to use a multi-core controller to fulfil performance needs. As in the past, mid-range applications will follow, as they inherit functionalities from the high-end applications, with a delay of two to three years. It is expected that each mid-range and high-end applications starting production in 2016 will have to use a multi-core controller as well. The challenge in engine management is to gain sufficient performance, even when having an application with closely coupled functions that has to be distributed across cores. The clear target from the beginning was to achieve a performance gain of factor 1.5 with three cores. As a result, the software architecture has to support the distribution of the software on the different cores in an efficient way; to reduce the inter-core communications overhead to an acceptable value. In the following chapters this paper will provide an overview of the features of the PowerSAR® software platform available at Continental Powertrain.

This software platform is flexible, in order to support legacy software which was developed on single cores and migrated into a “multi-core ready” style; as well as supporting application software which follows the AUTOSAR 4.0.3 - or higher - standard. Furthermore, it has to support different business models - starting from a “box business” where only low-level software parts are delivered via shared application software; up to a full-system business developed by Continental.

THE SOFTWARE ARCHITECTURE

Overall Architecture

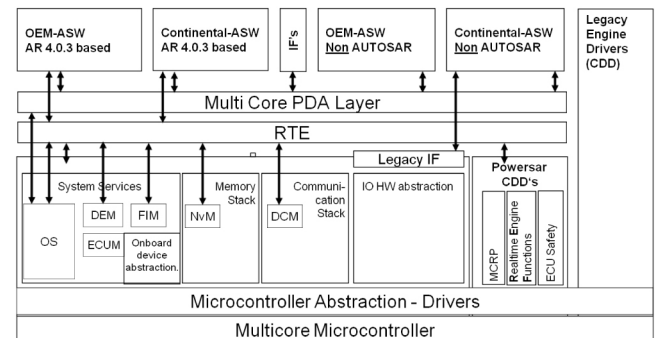


Figure 3. PowerSAR® static architecture

Static Architecture View

The architecture is based on the AUTOSAR 4.0.3 standard [2]. If necessary, it extends the AUTOSAR definitions to fulfil the requirements of a powertrain application [3]. Examples are the boot and reprogramming concept or the support of OBDII features in the DEM/FIM area. In this domain, features that were identified as not being required, such as the “Per Instance Memory Concept”, were dropped. The overall architecture is shown in Figure 3.

To support multi-core and ISO 26262 in an efficient way for AUTOSAR SWCs and legacy software, a so-called “multicore PowerSAR® Dynamic Architecture (PDA) layer” [4] has been introduced as an add on to the RTE. The basic software has to provide legacy interfaces, in addition to the AUTOSAR interfaces. As most of the applicative software delivered by the OEMs today is still non-AUTOSAR, the multicore communication layer is also in charge of the data protection of this kind of third-party code, to allow distribution to cores.

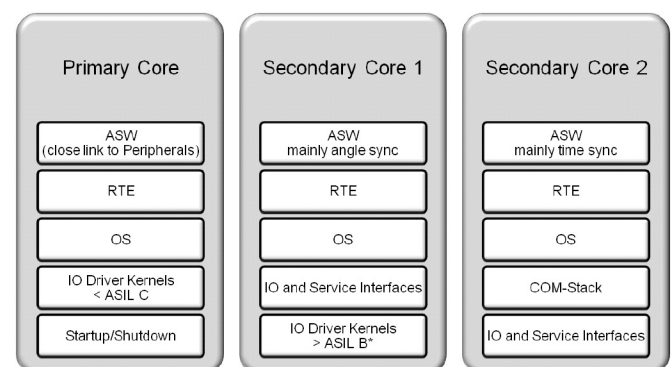


Figure 4. Core partitioning example