特集

Virtualization Technology, and Applying to E-Gas monitoring Concept*

中川　祐　Yu NAKAGAWA
荒井総一郎　Soichiro ARAI
Riccardo MARIANI

We have developed a virtualization technology for automobile use that allows the multiple functions to be run in parallel without interfering with each other on a single CPU (Central Processing Unit). As an example use of a virtual CPU, we examined the E-gas Monitoring Concept. This application was verified from the standpoints of achieving safety targets as well as cost when implemented in three configurations including the standard microcontroller (MCU : Micro Control Unit) + IC (Integrated Circuit) one, the virtual CPU one and the multiple CPU one. In that case, we evaluated each configuration using our evaluation index. The results confirmed that the virtual CPU had the best performance.

Key words : Virtualization ; Function integration ; Functional safety ; Freedom from Interference ; Software partitioning

電子要素

1. INTRODUCTION

In the 1970s, the length of an engine control ECU program was about 4,000 lines of code. As a result of the recent explosive growth of electronic controls, which is rooted in the growing demands to improve the environment by reducing emissions and fuel consumption and to enhance vehicle safety through systems such as airbags and seatbelts, a luxury vehicle may contain more than 80 ECUs that are larger in scale and far more sophisticated than ever before. Excluding the navigation system ECU, the total amount of program (software) code now exceeds 7 million lines. **Fig. 1** shows the trends for ECUs installed in luxury vehicles since 1995.

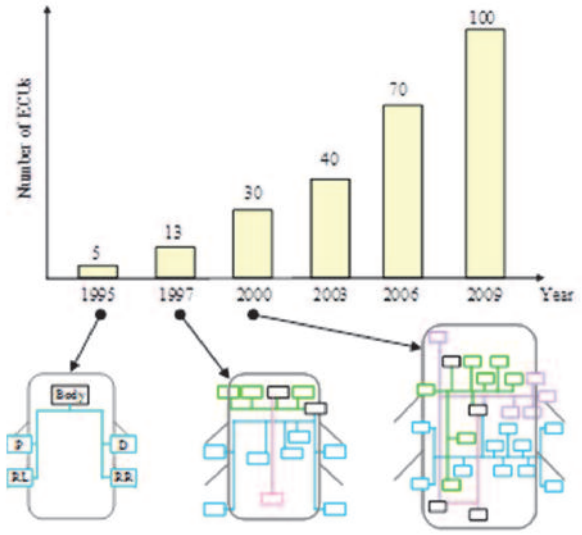

Fig. 1　Increasing number of ECUs installed in luxury vehicle

Along with such an explosive increase in the number of ECUs as seen in recent years, it has become increasingly difficult to secure spaces for installing ECUs in a vehicle from year to year. Therefore, in the field of vehicle development, there is a growing need for miniaturizing the component parts of the vehicle system and integrating those parts. However, if you are to develop an electro-mechanically integrated module that integrates an ECU and an actuator, you need to miniaturize the ECU to the size of an actuator ; otherwise, you will have an even harder time finding the space to mount the electronic components than you did previously.

On the other hand, the increase in the number of ECUs represents that highly-developed electronic control has come to play major roles in many aspects of vehicle control. Nevertheless, in the case in which system trouble occurs in a vehicle with such a complex electronic control system, it is difficult to identify how the vehicle is affected by the trouble ; in other words, it becomes very difficult to ensure the safety of the vehicle. Accordingly, in recent years, it has been required to respond to the functional safety defined in ISO26262. In the common technique for ensuring the functional safety of vehicle control, a double device method is being widely used, in which while one device is executing control, the other device that is designed for monitoring the

＊This paper is republished from FISITA 2014 paper F2014-AST-094, © 2014 KIVI .

controlling device is checking if everything is under control. However, as already mentioned, the miniaturizing of ECUs is required in order to make room for installation spaces. With that, we developed a technique that allows control function and monitoring function to be performed by a single device and the functional safety to be maintained in a smaller space.

2. VIRTUALIZATION TECHNOLOGY FOR AUTOMOBILE USE

Virtualization technology is widely known in the field of consumer products as the technique for integrating software that has different functions, such as above-mentioned control and monitoring functions while maintaining their separability. Virtualization technology is widely used for the purpose of constructing a mixed environment consisting from different OS (Operating System) or separating systems. We thought that this separability ensured by virtualization technology could be applied to substantiate the separability needed for ensuring functional safety.

On the other hand, there are specific challenges that the control software of the automotive field has to face, that is, how the high real-time property and durability of the software can be ensured. Those challenges made it difficult to apply the virtualization technology developed by the field of consumer products in the automotive field without making any changes to it. Therefore, we adopted an approach to optimize both hardware and software so that they could be used in the automotive field, that is to say, we have developed virtualization technology for automobile use. In addition, the unit separated by virtualization is hereinafter referred to as Virtual Machine (VM).

2.1 Hardware Approach - Virtualization assist CPU -

In the approach to optimizing hardware, we developed a CPU that is called a Virtualization assist CPU (Virtual CPU or VCPU) that strongly supports the virtualization with its hardware.

The VCPU is provided with a number of general-purpose registers and system control registers essential for program execution for each thread, an execution unit of the software. At the same time, the hardware is implemented with a scheduler that is capable of switching the threads that is to be executed in a unit of a CPU operating clock according to a preinstalled schedule. This makes approximate (pseudo) concurrent execution of multiple threads possible. More specifically, since the threads are switched over per CPU operating clock, adverse effects on the response of each thread is negligibly small. Additionally, the Memory Management Unit (MMU), which is responsible for allowing or disallowing the data access or instruction execution of a thread and also carrying out address virtualization, is able to be configured and operated for each thread in the VCPU.

Concurrently operating threads to which the VM is assigned in a pseudo manner by using the VCPU leads to achieving high performance real-time property.

2.2 Software Approach - Hypervisor for Automotive System -

Software called a hypervisor is widely known as the management software that provides functional convenience to the communications between VMs. As for software approach, we customized this hypervisor to be used in an automotive system. That is, we developed an automotive hypervisor.

The automotive hypervisor cooperates with hardware such as the MMU that is mounted on the VCPU described above. Configuring the software based on the assumption of utilizing the hardware makes it possible to implement the following function in a simple manner. That leads to reducing overhead in software, thus the real-time property will not be affected.

- Initialization and configuration of VMs
- Communication between VMs
- Synchronization and exclusion of VMs
- Management of VM shared memory

In addition, the automotive hypervisor was designed using the DRBFM (Design Review Based on Failure Mode), which is widely used in the automotive field. Thereby, even if one thread gets out of control, that will not affect the other threads, and thus, high consistency is achieved.

Fig. 2 shows the overall picture of the automotive virtualization technology configured in the VCPU and the automotive hypervisor.

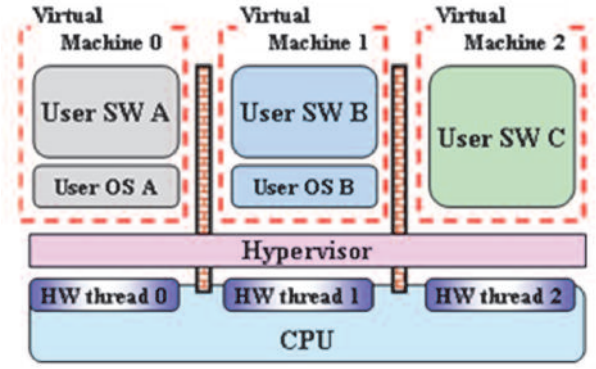

Fig. 2　Overview of virtualization technology for automobile use

3. EXPERIMENTAL PRACTICE : APPLYING E-GAS MONITORING CONCEPT

As a conventional functional safety-enabled application that is implemented by using two devices, we would like to refer to the E-gas Monitoring Concept. We analyzed the safety and cost associated in the realization of this application by a single device using the automotive virtualization technology described above.

3.1　E-gas Monitoring Concept

The E-gas Monitoring Concept describes the monitoring structure to ensure safety in automotive engine control. This concept consists of three levels (L1, L2, and L3) as shown in **Fig. 3**. Each level is outlined below.

L1 (MF : Main Function) : This is the primary functions of engine control. For example, processing such as calculating the required torque, monitoring the input and output sensors, and processing the fail-safes during abnormalities, are done at this level.

L2 (FML : Function Monitor Level) : This is the monitoring function for Level 1, and it detects software abnormalities in Level 1. For example, this compares the required torque calculated in Level 1 to the allowable values and monitors its suitability. If the calculated value is abnormal, the failsafe process is performed. Level 1 and Level 2 are typically run by the same controller.

L3 (CML : Controller Monitor Level) : This is the monitoring function for the function controller run on Levels 1 and 2, and straddles the function controller and monitoring controller. For example, when the monitoring controller makes a query, the answer from the function controller is checked by the monitoring controller, which detects any abnormalities with the function controller, and performs the failsafe process independently from the function controller.

Based on the functions of each level described above, it can be seen that the E-gas Monitoring Concept is made up of not one, but two controllers. Of these two controllers, it is customary to use a microcontroller (MCU) for the function controller, and an ASIC (Application Specific Integrated Circuit) or small-scale MCU for the monitoring controller.

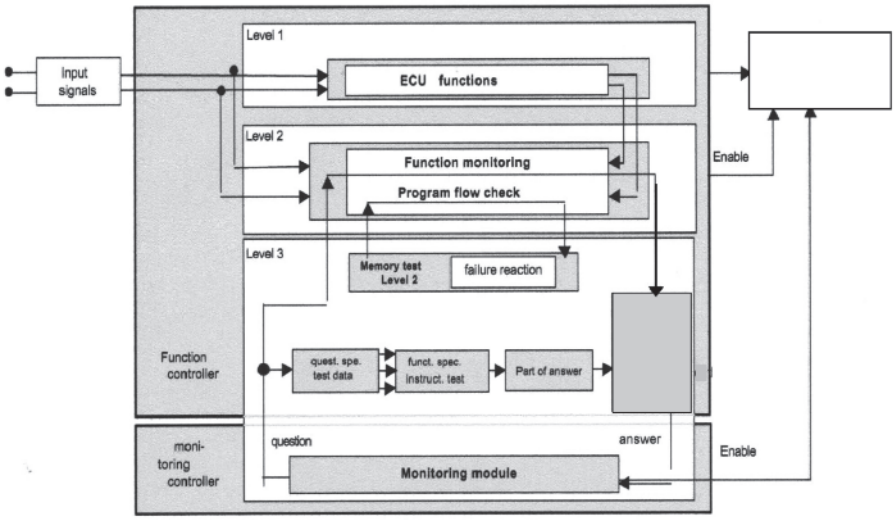

Fig. 3　Block diagram for E-gas Monitoring Concept

3.2 Case Study

We analyzed the E-gas Monitoring Concept by comparing the three cases that are implemented with the following hardware configuration described in **Fig. 4**. Evaluation indexes used in the comparison will be described in the next section in detail.

Case 1) is the standard approach with MCU and ASIC, where the CPU is allocated to the function controller and the ASIC is allocated to the monitoring controller.

Case 2) is a single MCU, in which both L1, L2 and L3 E-gas layers are executed in the same CPU. It is assumed that each L1/L2/L3 is in a separated VM.

Case 3) is a single MCU with a multi-core architecture, in which L1, L2 and L3 E-gas layers are executed in two different CPUs.

3.3 Analysis Approach

In this article, we evaluated each case individually to determine whether they could meet either of the ASIL (Automotive Safety Integrity Level, it can take the level of ASIL A, ASIL B, ASIL C or ASIL D) specified in the ISO26262, and also examined them from qualitative aspects relating to achieving the safety goal. After adding the aspect of hardware cost, we performed a comprehensive evaluation. Approaches taken for evaluating each case are discussed below.

Achievable ASIL for each case was determined after obtaining hardware metrics for each case according to the definition of the ISO26262. The hardware metrics involves SPFM (Single Point Failure Metrics), LFM (Latent Failure Metrics) and PMHF (Probabilistic Metrics of Hardware Random Failures). More specifically, the hardware metrics were obtained through calculation using FMEDA (Failure Modes Effects and Diagnostics Analysis) after assuming the safety mechanism.

Then, with regard to qualitative aspects related to the achievement of the safety goal, the following four aspects were specifically evaluated. Those aspects were individually examined in the light of the know-how we have established and the number of "+" was counted : the results were judged by the number of "+." In other words, the more the number of "+" is, the better the result was judged.

- Coherency between safe tasks and unsafe tasks
- Separability between L1, L2 and L3 tasks
- Avoidability and detectability of dependent failures
- Processing performance

Lastly, regarding the hardware cost, chip-size of each MCU and IC were calculated on the assumption that the process rule, memory size and functional requirements of semiconductor in each case are the same. Chip size is determined by taking into consideration the size of a mounting area and power consumption of the ICs.

We put these three evaluation indexes together and made the comprehensive index as defined below, by which the final judgement regarding the comparative merits and demerits of the three cases was made. The ASIL score referred to here is a value set for each achievable ASIL. The value is in accordance with the value of SPFM obtained in each ASIL.

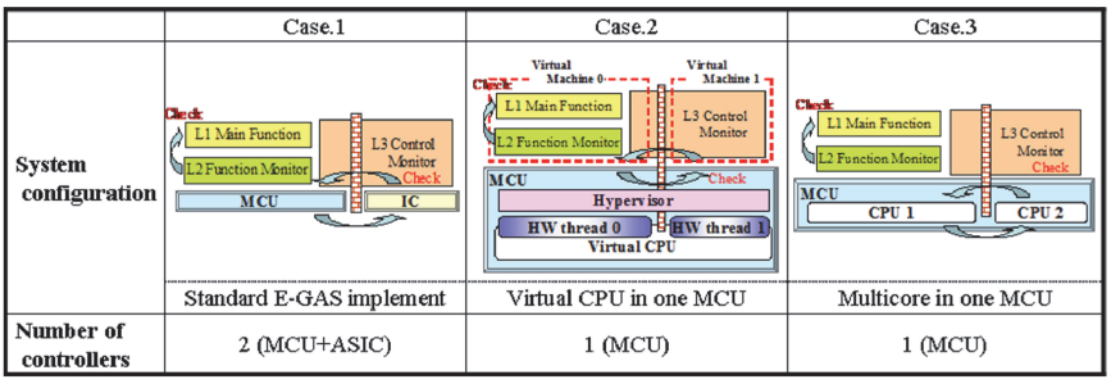

Fig. 4　Block diagrams for the three analysis cases

Table 1　Analysis results table

			Case 1 MCU+ASIC	Case2 VCPU	Case3 Multicore
Quantitative index	Achievable ASIL		ASILB	ASILC	ASILC
		SPFM (Permanent Faut) - %	95.0	98.9	98.9
		SPFM (Transient Fault) - %	99.0	98.9	99.45
		LFM (Permanent Fault) - %	95.4	97.8	97.1
		PMHF (Permanent Fault) –FIT	8.6	1.3	1.6
		PMHF (Transient Fault) – FIT	19	21	20
	Chip size –mm2		76	29	43
Qualitative index		Coherency between safe tasks and unsafe tasks	0	++	0
		Separability between L1, L2 and L3 tasks	+++	++	+++
		Avoidability and detectability of dependent failures	+++	0	++
		Processing performance	+	+	++++
	the number of "+"		7	5	9
Comprehensive index			2.2	4.0	3.6

(Comprehensive index) = [ASIL score / chip size] + [the number of "+" normalized in case1]

4. RESULT AND DISCUSSION

Table 1 shows the analysis results. Please note that these results were obtained by putting focus only on the microprocessor, and actually, the ASIL for ECUs must be determined after obtaining the failure rate and detection rate of the whole ECU, including the other components.

When comparing Case1 and Case2, there is no decline in the achieved ASIL. That is to say, it was confirmed that the functional safety configuration achieved by two devices can be achieved in a single integrated device by using the virtualization technology for automobile use, without causing any disadvantage in terms of the hardware metrics. In addition, the chip area of the semiconductor of Case2 was reduced to about 0.4 times the size of that of Case1. Therefore, it is possible to achieve the functional safety equal to the conventional one with a smaller mounting area.

On the other hand, comparison of the virtualization technology with the multi-core, that is, comparison between Case2 and Case3 shows that Case2 is slightly better in the overall evaluation ; however, there is no remarkable difference therebetween. However, the chip area of Case2 is about 1.5 times as large as that of Case3. Considering the mounting area, such a difference is something to be reckoned with. That is, if the functional safety is to be achieved by a single device, there is more advantage in a solution using the virtualization technology than that of using the multi-core, especially when considering the mounting space.

From these results, it was confirmed that the automotive virtualization technology is useful in ensuring the functional safety in a limited mounting space.

However, virtualization technology is not necessarily superior in all aspects. As shown in **Table 1**, in the evaluation of the qualitative aspect involved in achieving the safety goal, the number of "+" serving as an indicator in Case2 is the lowest. Particularly, the number of "+" in the item, avoidability of dependent failures, in Case2 is much lower than those of the other cases. The following is one of the reasons why the avoidability of dependent failures is not high in Case2. In the functional safety configuration achieved by the virtualization technology in Case2, a single CPU is shared by all the features of the L1, L2 and L3. Therefore, in a case that a CPU gets out of control, for example, there is a possibility of causing a malfunction even in L3 that is monitoring the entire control system. As a result, L3 might not be able to work properly as a control system to ensure safety. Such a failure, which is due to a single cause that produces damage to multiple functions is referred to as Common Cause Failure (CCF). When we develop the functional safe-

ty configuration using virtualization technology like in Case2, how to deal with the CCF will pose a more challenging issue than when developing other cases.

4.1 Response to CCF

If a single resource shared between multiple functions is found to have an abnormality, it might result in the CCF. In the case of single-core configuration such as in Case2, clock signal, power supply, peripheral functions or the CPU can be considered as shared resources. However, the recent microcontrollers are normally provided with a mechanism for automatically detecting abnormality as standard hardware with respect to a clock, power source and important peripherals such an A/D converter. On the other hand, it is also possible to detect the abnormality in a CPU to some degree by using a simply designed watchdog mechanism. Accordingly, it is possible to develop a design that can maintain a safe state by detecting most of the abnormalities in the shared resources, without employing special devices.

However, you cannot completely detect CPU abnormalities with the watchdog. For example, when a CPU experiences a livelock state, which is the abnormality where the entire system repeats a busy state and the control system ceases to work, the watchdog seems to be operating properly ; however, such an abnormality cannot be detected by a simple watchdog. Such an abnormality might be possibly solved by providing a device with the capability of monitoring the processing flow of the CPU ; however, such a design change might lead to the increase in the size of chips. When it is difficult to solve problems of a system by adding hardware in connection with the installation of ECUs, as discussed in this article, it is desirable to perform detailed safety analysis and safety verification including the implementation of fault injection in the development process and confirm the safety of the system.

5. CONCLUSION

The vehicle electronic system is increasingly becoming complex, in which it is imperative to achieve functional safety, and at the same time, there is a growing need for the miniaturization and integration of system components. This time, as a technique for achieving functional safety in a space-saving manner, we have developed the automotive virtualization technology. We newly developed two techniques including the VCPU that supports the virtualization with its hardware and the automotive hypervisor that cooperates with it, and finally reached the achievement of the automotive virtualization technology.

As a functional safety application, we addressed worked on the E-gas Monitoring Concept, through which the advantageous effect brought about when this application is realized with a single device by using the automotive virtualization technology was evaluated by comparing with other hardware configurations. As a result, it was confirmed that when the automotive virtualization technology is employed, it is possible to maintain the conventional level of functional safety, while reducing the size of chips to the minimum. Therefore, the automotive virtualization technology has been proved to be useful for obtaining the needed functional safety in a limited mounting space.

REFERENCES

1) Gräter A. Safety of Electric Vehicles During Their Life Cycle. ATZ autotechnology, 2011, volume 11.
2) Schäuffele, J. Automotive software engineering principles, processes, methods and tools. USA : SAE International, 2005.
3) Kato, M. Automotive Electronics : Systems. Tokyo : Nikkei Business Publications, Inc., 2010
4) Kato, M. Automotive Electronics : Basic Technologies. Tokyo : Nikkei Business Publications, Inc., 2010
5) M. Baleani, et al. “Fault-Tolerant Platforms for Automotive Safety-Critical Applications”. In Proc of International Conference on Compilers, Architecture and Synthesis for Embedded Systems, 2003, 170-177
6) Arbeitskreis EGAS. Standardisiertes E-Gas-Überwachungskonzept für Motorsteuerungen von Otto- und Dieselmotoren. Verband der Automobilindustrie, 2005, Report Version 2.0.
7) R. Mariani, M. Baumeister, P. Fuhrmann. “A single channel, fail-safe microcontroller to simplify SIL3 safety architectures in automotive applications”. Electronic Systems for Vehicles VDI Conference, 2007.
8) R. Mariani, F. Colucci, and P. Fuhrmann. Safety integrity of memory sub-systems in automotive microcon-

trollers. SAE 2007 World Congress, 2007, 2007-01-1494

9) S. Brewerton, R. Schneider, and D. Eberhard, “Implementation of a Basic Single-Microcontroller Monitoring Concept for Safety Critical Systems on a Dual Core Microcontroller” . SAE 2007 World Congress, 2007, 2007-01-1486

10) Yoshimura T. Toyota prevention method GD3. Tokyo, JUSE Press, Ltd., 2002

11) Mariani, R. “The impact of functional safety standards in the design and test of reliable and available integrated circuits”, Test Symposium (ETS), 2012 17th IEEE European, 2012, page 1.

12) Yukihide N. et al. Virtualization technology and using virtual CPU in the context of ISO26262 : The E-gas case study. SAE 2013 World Congress, 2013, 2013-01-0196

＜著　者＞

中川　祐
（なかがわ　ゆう）
基盤ハード開発部
電子基盤技術の企画・開発に従事

荒井　総一郎
（あらい　そういちろう）
基盤ハード開発部
電子基盤技術の企画・開発に従事

Riccardo MARIANI
YOGITECH SPA CTO Ph.D.
半導体の機能安全検討などに従事