**Proceedings of 2013 International Symposium on Embedded Technology(Interactive Session Only)**
**(ISET 2013)**
*Daegu Metropolitan City, 23-24 May, 2013.*

# Inter guest OS Communication for ARM TrustZone virtualization on multi-core processor

MooWoong Jeon[*], Se-Won Kim[†], SukYoung Rho[††], HyunWoo Lee[††], Chuck Yoo[†]

[*]Graduate School of Convergence IT
Korea University, Seoul, Republic of Korea
E-mail: moongstyle@korea.ac.kr

[†] Collage of Information and Communication
Korea University, Seoul, Republic of Korea
E-mail: {swkim, chuckyoo}@os.korea.ac.kr

[††] Hyundai Motor Company
Seoul, Republic of Korea
E-mail: {stoneno, lhwpro}@hyundai.com

**Abstract**

Virtualization architecture for appending additional system into existing operating system causes extra hardware and software costs and requires communication method between the separated two systems. To solve this problem, ARM TrustZone provides execution of two guest OSes on a physical processor. This paper presents an inter guest OS communication for ARM TrustZone virtualization on multi-core processor.

**Keywords:** ARM TrustZone, Multi-core, Inter Processor Interrupt (IPI)

## 1. Introduction

In order to add extended functions to existing operating system, a traditional approach is to append additional system into existing system, each with its own CPU, memory, and peripherals. A new operating system that contains extended functions runs on the appended system. This solution causes extra hardware and software costs and requires communication method between the separated two systems: appended and existing system. To reduce these costs and additional facilities, virtualization is a way to run several operating systems on same hardware. Due to the lack of virtualization support of ARM processors, virtualization in embedded system has been considered as an unrealistic solution. However ARM TrustZone provide easy way to virtualize ARM embedded processors. Each of the physical processor cores supports ARM TrustZone has two virtual processor execution modes, Secure World and Normal World. Each World runs its own guest operating system separately and TrustZone provides inter World context switching mechanism, known as monitor mode[1]. SafeG[2] is typical example to use this technology. In this paper we present an inter guest OS communication for ARM TrustZone virtualization on multi-core processor.

## 2. A communication mechanism using ARM Trustzone

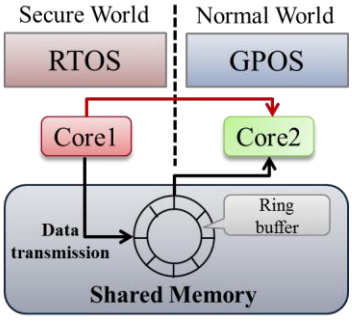


**Figure 1.**

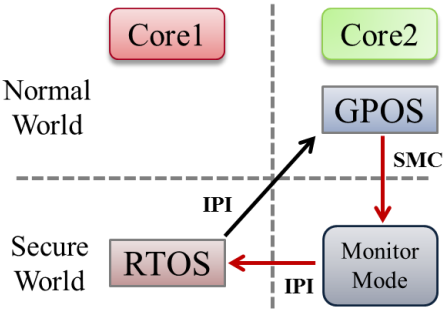


**Figure 2.**

Figure 1 shows an inter guest OS communication mechanism. Communication process consists of *data transmission* and *notification*. *Data transmission* can be read and written from the ring buffer in the shared memory of the two worlds. In order to transfer data, a guest OS write the data to ring buffer and notify corresponding guest OS. When the corresponding guest OS received a notification, it reads the data from the ring buffer.

As Figure 2, *notification* was designed to use Secure Monitor Call(SMC) instruction and Inter Processor Interrupt(IPI). IPI can be passed selectively to the core which a communication application runs on. By IPI, even though our communication is designed for inter-guest OS communication, it also supports inter-process communication (IPC). Secure World can directly send IPI to Normal world but Normal World cannot send directly. So Normal World should call SMC instruction to change Secure World and then send IPI to the Secure World of corresponding core.

Because of providing communication APIs per guest OS, a guest OS can be communicated the other guest OS although an application does not know the communication mechanism: data transmission and notification. An application can directly communicate with an application in the other guest OS through abstraction called communication port. Communication APIs are equipped with message buffer, callback function caller, communication port management. We use Linux as the operating system of Normal World. Linux controls all I/O through file operations such as open(), close(), read(), wirte(), and ioctl() functions. TrustZone driver provide an interface between Linux file operations and communication APIs to follow the I/O way of Linux.

## 3. Experiments

We have ported a Linux as Normal World and RTOS as Secure World over ARM Cortex-A9 MPCore board. Table 1 shows the average time of notification between two worlds. Experiments were performed 1000 times each, depending on notification rate.

| Notification rate (msec) | From Normal to Secure (nsec) | From Secure to Normal (nsec) |
|---|---|---|
| 10 | 190 | 1082 |
| 30 | 202 | 1082 |
| 50 | 190 | 1081 |
| 100 | 220 | 1092 |
| 300 | 223 | 1081 |
| 500 | 234 | 1084 |

**Table 1.**

## 4. Conclusions

This paper shows an inter guest OS communication for ARM TrustZone virtualization on multi-core processor. Communication process consists of *data transmission* and *notification*. *Data transmission* can read and write the data to ring buffer in shared memory of two worlds. *Notification* was designed to use SMC instruction and IPI. Secure World can directly send IPI to Normal world but Normal World should call SMC instruction to change Secure World and then send IPI to the Secure World of corresponding core. Communication APIs provide a communication of the other guest OS although an application does not know the communication mechanism: data transmission and notification.

**Acknowledgement**

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No.2010-0029180) with KREONET.

**References**

[1] ARM Limited, "ARM Security Technology: Building a Secure System using TrustZone Technology", http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.prd29-genc-009492c/index.html, 2009

[2] Daniel Sangorr´ın, Shinya Honda, and Hiroaki Takada. Dual operating system architecture for real-time embedded ystems. In Proceedings of OSPERT 2010, July 2010.