Software Infrastructure for Future Many-Core Embedded Systems

Yuki Kinebuchi, Wataru Kanda, Yu Yumura, Tatsuo Nakajima

Waseda University, JAPAN

ASMP and SMP multi-core processor

Today multi-core processor is widely used in embedded systems. Most common multi-core embedded systems are based on asymmetric multi-processor architecture (ASMP). This is suitable for running multiple different OSes on each core (Fig.1).

On the other hand, ways to leverage many-core processor based on symmetric multi-processor architecture (SMP) in embedded systems, are also discussed. The common purpose of using SMP is to let a single OS to support better parallelism (Fig.2).

SMP could be used asymmetrically to execute multiple OSes simultaneously. This could be done by simply modifying OSes not to share memory and peripheral devices. However, the model binding each OS to fixed core cannot utilize the benefit of SMP architecture (Fig.3).

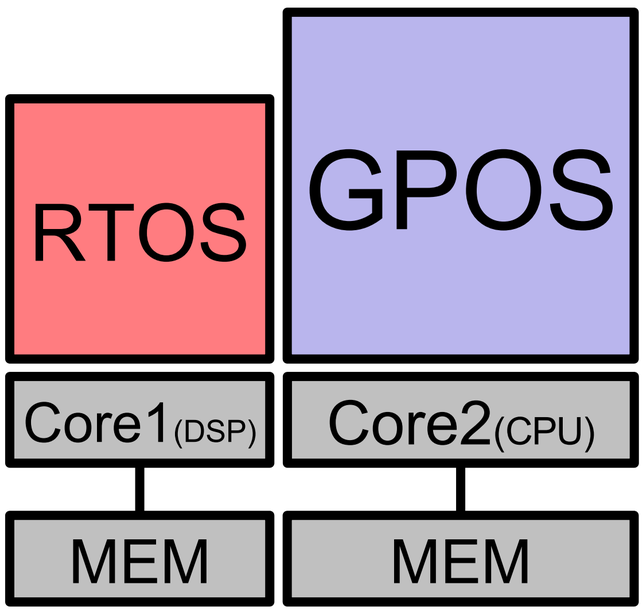

Fig.1 ASMP with multiple OSes

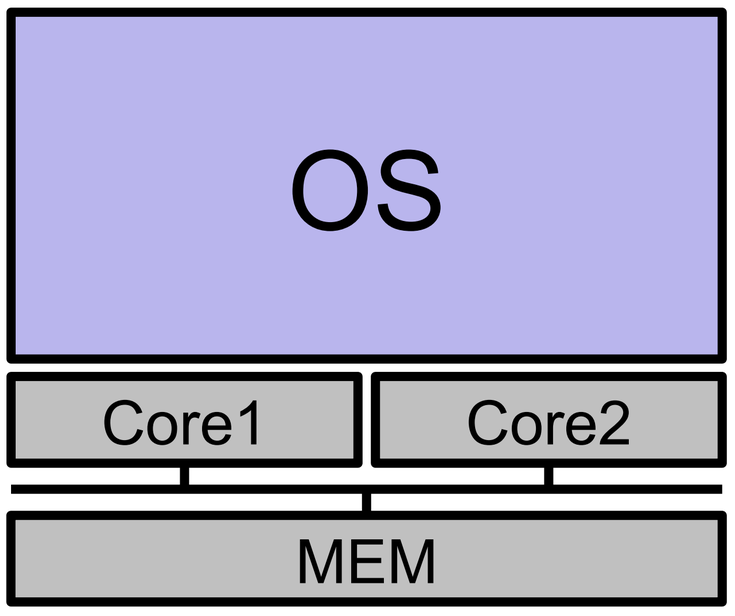

Fig.2 SMP with single GPOS

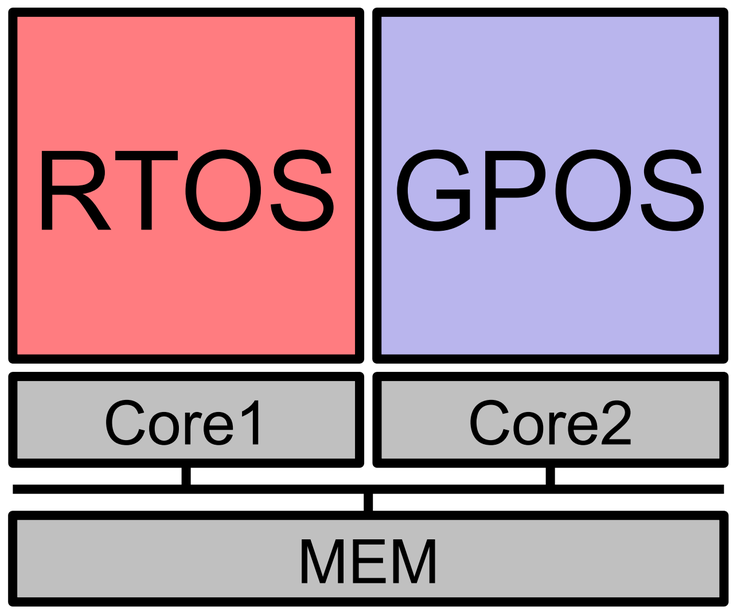

Fig.3 SMP model multiple OSes

Sharing Cores between OSes

We propose a software core multiplexer which enables to share cores between multiple OSes dynamically to benefit from the flexibility of SMP architecture (Fig.4). For instance, When the core utilization of RTOS is less than 100%, the remaining calculation power could be leveraged to execute GPOS code.

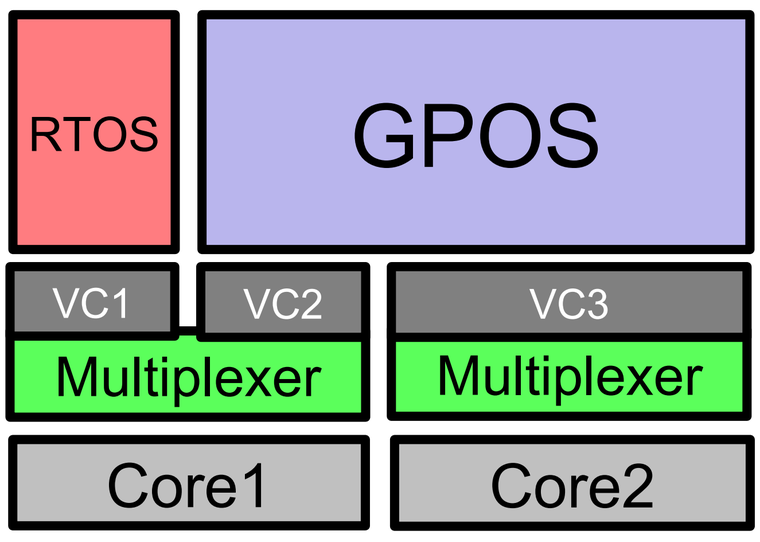

Fig.4 Executing two kernels on virtual cores with SMP

Saving Power Consumption

When the total amount of calculation between two OSes is small enough to be done on a single-core, and if the GPOS could dynamically stop using one of its cores, one of the real core could be stopped. This mechanism would help saving system power consumption (Fig.5).

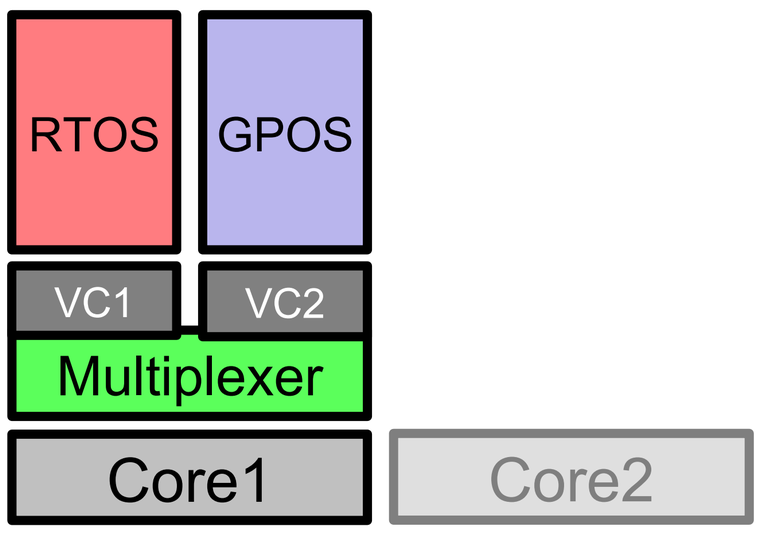

Fig.5 Stopping a core to save power consumption

Current Status

To achieve sharing cores between multiple OSes, we first started with implementing a software core multiplexer on a single-core device. So far, we are done with the prototype implementation. We used TOPPERS and Linux as a RTOS and GPOS. They are executed concurrently on a SH-4A architecture single-core processor. Since the multiplexer is designed to keep the original ABI of the SH-4A architecture, the modification is less than 20LOC for each OSes.

Summary

We proposed a software infrastructure to let multiple OSes to share cores provided by SMP. This functionality is expected to give more flexibility to an embedded system consists of multiple different types of OSes on SMP. When many-core embedded systems become more common, the dynamic assignment of cores among OSes could be leveraged to let processors to be used more efficiently.

We are done with the prototype implementation with a single-core processor. Currently we are adapting our infrastructure to a SMP multi-core platform.