

The yin and yang of multicore

Once they start working with multicore devices, developers are faced with significant choices.

There are two apparently opposing forces in multicore architecture: one to make software running on groups of processors work together better; the other is to isolate them from each other. Both come together in markets like automotive, where manufacturers need the performance of multicore architectures to deliver advanced driver assistance systems (ADAS) and, ultimately, autonomous driving.

As well as trying to obtain more instructions per cycle, automotive OEMs want to reduce the costs associated with the seemingly relentless proliferation of MCUs across the vehicle. Mentor Graphics embedded virtualisation product manager Felix Baum says: “They are consolidating functions for ADAS and other systems, bringing together and running multiple pieces of software. One part may do sensor fusion and another handling communication with ECUs.

“All these pieces that used to run on distinct CPUs are now running on one SoC. When you do that, there are certain things that you have to deal with. How do they communicate and how you start and stop them. How do we make sure this communication is synchronised and works efficiently?”

As soon as they start working with multicore architectures, developers face difficult choices. The most

fundamental is: where will each task run? Most multicore SoCs are asymmetric multiprocessors, with processors tuned for different tasks, which can guide allocation. But the higher-performance SoCs may also contain clusters of symmetric multiprocessors. The decision then becomes one of whether to let a multiprocessor-aware operating system balance the workload dynamically across the available symmetric cores or lock certain groups of tasks to specific cores to ensure they can be scheduled quickly when needed.

One issue with allocating tasks is that it might be better to change where they are allowed to run over time depending on circumstances. For its military and aerospace user base, Green Hills Software developed a technology called time-variant unified multiprocessing (tuMP) that allows processors to be re-allocated as they become spare. The Integrity-178B operating systems acts as a hypervisor for the entire SoC. It coordinates communications between tasks that may run on symmetric clusters in some modes but, to guarantee their performance during other modes, switch some of those processors into a asymmetric architecture.

In other cases, a static allocation seems obvious. Tomas Evensen,

embedded software CTO at Xilinx and chair of the Multicore Association’s OpenAMP working group, points to the way in which the internet of things (IoT) affects embedded systems: “People using our devices in these spaces have real-time needs but they have to connect to services running in the cloud using Microsoft Azure or Amazon. They don’t want to use an RTOS for handling those communications. They want to use Linux for that.”

The Multicore Association is developing a number of standards to manage resources and streamline communication between cores running Linux and others running various flavours of RTOS on the same SoC. Rather than reinvent the wheel, OpenAMP borrows the interprocessor communications model developed by the Linux community and applies it to other operating systems.

“A lot of projects use Linux for at least one operating system in their embedded systems. Going to Linux folks and telling them to transition to some proprietary method for embedded? That’s really hard. So we looked at Linux and picked capabilities that are already present in a mainstream Linux kernel,” Baum says.

The group is developing cleanroom implementations of the open-source code and supplying them under



BSD licence to avoid the problems associated with the Gnu General Public Licence (GPL) that demands developers make publicly available their own improvements and linked code.

As well as interprocessor communications, OpenAMP supports remote procedure calls to let tasks on one core request services from those running on another. Baum says Mentor has built a number of reference designs around OpenAMP.

“We have built an ethernet switch. Another is a shared graphics virtualisation framework, where you can have two or more cores sharing a GPU. You might have screens in a car dashboard that you want to be shared. One half may be owned by a subsystem running Android; the other half by Linux. Then at the top and bottom you show status information like battery level and signal strength. That’s controlled by an RTOS perhaps. We have real customers using that kind of framework,” Baum claims.

Although the APIs are common and are designed to work on SoCs that may only have shared memory to use for sending messages to and from processors, silicon and operating systems vendors have the option to make their own enhancements to speed up communication.

“What happens under the hood is up to each vendor. If they have a fast [onchip network] fabric they can use that. Shared memory instead? They can use that. Any silicon goodness they have they can use for optimisation,” Evensen says.

The complex interactions between tasks found in mainstream Linux systems, such as those running on cloud servers, have demonstrated how sensitive to hardware implementation multicore performance can be.

Developers of low-level software such

as mutual-exclusion semaphores (mutexes) that lock tasks out from sections of memory while another performs work on it have to take into account the multiple levels of storage hierarchy in today’s non-uniform memory architecture (NUMA) computers. It is a hierarchy that has fast local caches served by larger caches shared between cores in a cluster that talk to offchip main memory.

Samy Al Bahra, CTO at Backtrace I/O, points out: “With NUMA, the cost to access memory varies. If you have a mutex that is NUMA oblivious you can end up with performance mismatches. That can become starvation and even livelock under extreme load.”

As using locks to coordinate threads can damage performance, the developer community is working on mechanisms to speed up their operation and, in some cases, moving towards lockless synchronisation. “But non-blocking synchronisation is not a magic bullet; there is a lot of hype around these data structures. Getting the most out of them requires a deep understanding of your workload,” says Al Bahra.

Although synchronisation for communicating tasks needs to be efficient, in embedded systems running multiple operating systems, particularly those that need to provide safety guarantees, a key requirement is preventing some tasks from interacting with others. “The bigger picture is how we run apps side by side in a safe and secure manner,” Baum says.

ARM product marketing manager Phil Burr notes: “What we are ending up with is multiple pieces of software with different criticality needs running on one SoC or the same processor core.”

In a system without a mechanism



As soon as they start working with multicore architectures, developers face difficult choices. The most fundamental is: where will each task run?



to isolate tasks from each other and prevent them from overwriting the data of others, the mixing of code leads to a major headache for guaranteeing safe behaviour. A small update to graphics code for controlling the audio systems leads to the need to revalidate software that handles the rear-view camera – a function that must be built into US cars from May 2018. “Separation is vital for reducing the amount of certification,” says Burr.

Processors such as ARM’s recently launched Cortex-R52 add hardware protections and support for hypervisors to control how tasks running in different memory partitions talk to each other. Baum says OpenAMP was designed to work with this level of separation: “We are not dictating to customers their architecture. Maybe they run operating systems natively side by side or they want to use a hypervisor. Some are using ARM’s Trustzone to provide hardware-enforced separation. But the communication between tasks where it needs to happen is the same: that’s what OpenAMP offers.”

newelectronics

web: www.newelectronics.co.uk