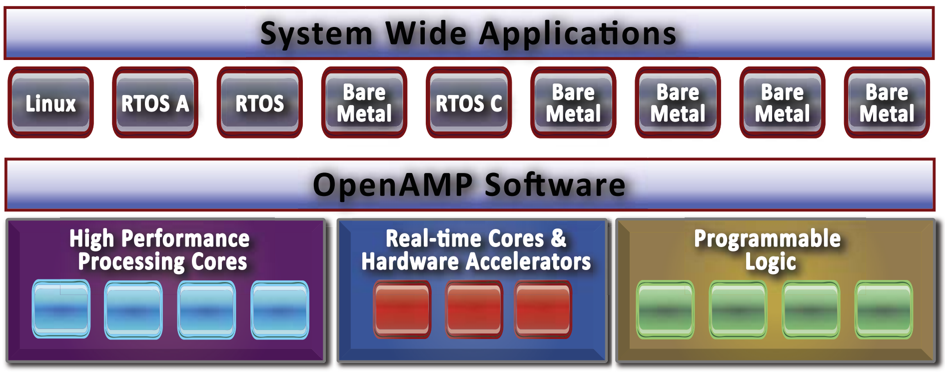
THE Multicore® ASSOCIATION

Join the Multicore Association and Contribute to the OpenAMP Standard

OpenAMP – The Open Source Framework for Complex Multicore Architectures



Operating systems can interact within a broad range of complex homogeneous and heterogeneous architectures.

Addressing Multicore Architectures and Hardware Accelerators

Complex SoC architectures are combining increasing numbers of application-class and microcontroller-class cores. As a result, consolidation of heterogeneous operating environments on a single device is more difficult to achieve – and more difficult for developers to utilize the underlying hardware.

While symmetric multiprocessing (SMP) operating architectures allow load balancing of the application across homogeneous processors within the multicore infrastructure, they do not scale to heterogeneous cores. Historically, there was a lack of accepted standards and software design paradigms to take full advantage of asymmetric multiprocessing (AMP), even on homogeneous multicore SoCs. Having mechanisms in place would enable AMP applications to leverage parallelism offered by the multicore configuration.

Introducing the OpenAMP Framework

To address these design complexities, the Multicore Association is expanding the development of OpenAMP, the body of work already available on GitHub and in the Linux distribution. This open-source software framework allows developers to configure and deploy multiple operating systems and applications across homogeneous or heterogeneous processors. This comprehensive framework enables developers to manage the challenges associated with inter-process communication (IPC), resource management and sharing, and process control. The framework also allows software developers to control the boot-up and shut down of individual cores on an SoC, thus allowing applications to maximize compute performance or minimize power consumption based on individual use case scenarios.

FEATURES

- Configure, deploy, and manage multiple OS's across homogeneous and heterogeneous cores
- Availability of open source Linux implementations and proprietary RTOS and bare metal implementations
- Android OS compatibility
- Inter-OS & inter-processor communication
- Shared memory protocol – Virtio/rpmsg
- Lifecycle APIs – remoteproc
- Proxy technologies emulate Linux processes

BENEFITS

- Simplify heterogeneous AMP design
- Collaborative definition and development process
- Standardizes OS interaction between Linux and RTOS/bare-metal
- Compatibility with Multicore Communications API (MCAPI) to support high-performance use cases and zero-copy
- Supported by broad ecosystem of operating systems and silicon hardware platforms
- Pre-ported OS support by Express Logic, FreeRTOS, Mentor, Micrium, NXP, Xilinx

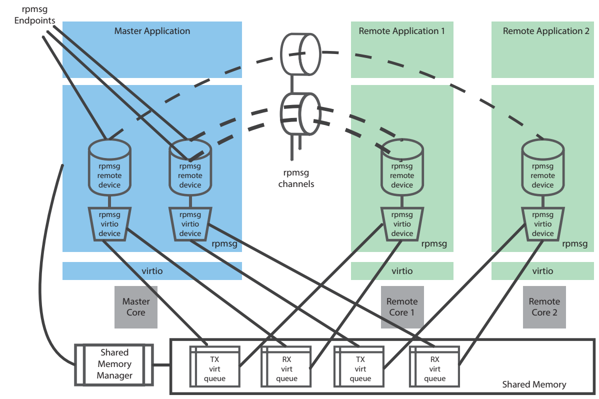
Inter-processor Communication (IPC)

IPC enables collaboration between heterogeneous cores running various software stacks. Since the software stacks vary from bare metal to fully configured Linux (or Android), one IPC mechanism would not address all use-cases. As such, the IPC mechanism utilized by a framework has 2 levels. For simple point-to-point communication, the virtIO framework widely adopted in Linux distributions to map physical memory directly into user spacer applications, have been enhanced to accommodate the multi-core and multi-OS paradigm. The virtIO becomes the transport layer that enables both remote processor lifecycle management and IPC. The second IPC level is provided by the remote processor messaging framework (rpmsg) to establish a communications channel, adding the messaging structure on top of the simple shared memory virtIO framework for the multiple applications running on various software stacks to communicate between. Both virtIO and rpmsg are found in many Linux distributions. The OpenAMP framework provides a cleanroom implementation for both levels to allow developers to communicate with non-GPL software stacks.

Remote Processor Life Cycle Management

Assuming control over a remote processor, and then starting or stopping an OS and/or application stack within that remote processor, is referred to as remote processor (remoteproc) lifecycle management. The Linux community has adopted a remote processor framework for managing this scenario and the most recent release is now part of the Framework. Remoteproc allows a master operating system to bring up other operating systems on other cores.

The remoteproc feature within the OpenAMP Framework allows remote processor interoperability between Linux, RTOS's, and bare metal environments. A key benefit to remote processor lifecycle management is reduced power consumption. The remote processor stays in a low power state when not in use. Only after remoteproc is used to bring up the remote processor and deploy the necessary firmware does the remote processor draw any notable power.



Use standard Linux process concept to manage other OS's

Simplified Booting

Booting a heterogeneous system is also not as simple as booting an OS on a dedicated processor. One needs a way to manage the booting of operating systems across the various cores, and to manage the applications that run on those processors. For example, performance requirements may dictate a certain boot order of the components. The Framework provides the capabilities to manage the booting of operating systems and applications across cores via the support of the remoteproc framework, which can be used for Mentor Embedded Linux, Nucleus RTOS, and even bare metal implementations.

ABOUT THE MULTICORE ASSOCIATION

The Multicore Association provides a forum in which all relevant multicore standardization issues are discussed and resolved. Our objective is to help developers achieve quicker time to market, and part of this is giving our members the ability to certify to their customers that their products are compatible with The Multicore Association standards.

Multicore Communications API (MCAPI) captures the elements of communication and synchronization required for closely distributed (multiple cores on a chip and/or chips on a board) embedded systems and widely-distributed systems (Internet of Things). Target systems span multiple dimensions of heterogeneity (e.g., core, interconnect, memory, operating system, and programming language). Join this working group to help expand the usefulness of MCAPI for Version 3.x (including 'zero copy' functionality), as well as official subsets.

The Multicore Programming Practices (MPP™) working group has collaboratively developed a multicore software programming guide for the industry that aids in improving consistency and understanding of multicore programming issues. Interested in contributing to the next version of this document?

SHIM™ is the Software-Hardware Interface for Multi-Many Core. It provides a common interface to abstract the hardware properties that matter to multicore tools. These hardware properties include the processor cores, the inter-core communication channels (in support of message passing protocols such as the Multicore Association's MCAPI), the memory system (including hierarchy, topology, coherency, memory size, latency), the network-on-chip (NoC) and routing protocol, hardware virtualization features, and more.

Questions about membership, participation in Working Group meetings, media, press releases, and about the infrastructure of this consortium:

Markus Levy, President

markus.levy@multicore-association.org

Phone: 530-672-9113

Mail: PO Box 4794 - El Dorado Hills, CA 9576