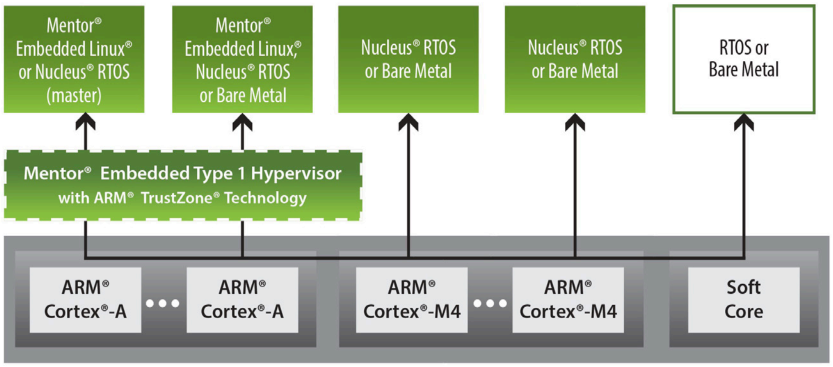
MENTOR EMBEDDED
Multicore Framework

DATASHEET



The Mentor Embedded Multicore Framework enables a broad portfolio of runtime environments on complex homogeneous or heterogeneous cores.

Addressing Hardware complexity and Advanced Multicore Architectures

Today's complex system-on-chip (SoC) architectures are combining more application-class and microcontroller-class cores than ever before. As a result, consolidation of heterogeneous operating environments on a single device is much harder 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. Further, there is a lack of accepted standards and software design paradigms to take full advantage of asymmetric multiprocessing (AMP), even on homogeneous multicore SoCs. Having certain mechanisms in place would enable AMP applications to leverage parallelism offered by the multicore configuration. Embedded virtualization allows the mixing and matching of SMP and AMP environments and supervises some of the operating systems while managing payload on the cores.

Introducing the Mentor Embedded Multicore Framework

Acknowledging the design complexities of AMP architectures, the Multicore Association (MCA) has created the OpenAMP™ standard. Mentor Graphics contributed the initial software to the OpenAMP open source project and created the Mentor® Embedded Multicore Framework, the first commercial implementation of the OpenAMP standard, allowing developers to configure and deploy multiple operating systems and applications across homogeneous or heterogeneous multicore processors. This comprehensive framework enables developers to manage the many challenges associated with inter-process communication (IPC), resource management and sharing, process control, debugging, and application optimization within a multicore environment by supporting native, virtualized, and trusted configurations of multiple operating systems. The Framework allows software developers to control the boot-up and shut down of individual cores on a SoC, thus allowing applications to maximize compute performance or minimize power consumption.

SOLUTION FEATURES:

- Configure and deploy multiple operating systems across homogeneous or heterogeneous processor cores
- Consolidation of the system – including discrete components
- Supports native, virtualized, and trusted configurations of multiple operating systems
- Simplified booting
- System-wide communication
- System level visualization of heterogeneous system behavior
- Common toolchain supports the entire solution enabling heterogeneous system development, debug, and analysis

BENEFITS:

Enables consolidation
Leverage multiple operating systems within a multicore design while easily managing payloads across processor cores

Manage compute performance and power consumption
Developers are given control over boot-up and which cores to shut down on a SoC, also helping to manage power consumption

Locate and fix problems deep in the system
Powerful tools allow developers to see deep inside multiple cores and fix problems quickly on a single common timeline

Part of a larger multicore ecosystem
Mentor's broad portfolio of runtime environments, integrated tools, services, and partner technologies allow projects to start quickly and innovation to begin earlier

Inter-processor Communication (IPC)

Once the remote processor OS and application stack are running, many use cases will require communication with other parts of the system. The Framework provides a cleanroom implementation of a remote processor messaging framework feature called *rpmsg* to establish a communications channel between the master operating system and the remote operating systems. In this way, data can be passed back and forth between the two in an inter-processor communication channel.

The transport layer that enables both remote processor lifecycle management and inter-processor communication is *VirtIO*. VirtIO is a virtualization standard for high performance input/output device drivers widely adopted in virtualized Linux® environments.

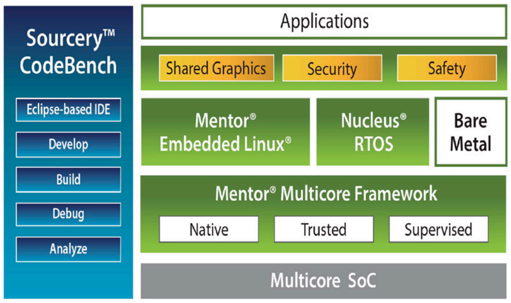
Remote Processor Life Cycle Management

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 Framework allows remote processor interoperability between Mentor® Embedded Linux®, Mentor® Embedded Nucleus® RTOS, 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.

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.



The Mentor Embedded Multicore Framework is integrated with Mentor® Embedded Sourcery™ CodeBench.

Visualization Deep into the System

Developers must be able to visualize how the heterogeneous components interact with each other in the consolidated heterogeneous systems. Because the systems are consolidated on shared hardware, the chances of running into resource contention and bottlenecks are greatly increased. Developers require tools that can help them identify those contentions and bottlenecks, and enable them to quickly find solutions to the problems. Sourcery™ CodeBench with built-in Sourcery™ Analyzer have been integrated into the Framework allowing various OS and virtualized guest runtimes, along with the applications, to be visualized on a single common timeline.

More about Mentor Embedded

The Mentor Graphics® Embedded Systems Division comprises the Mentor Embedded™ family of products and services, including embedded software IP, tools, and professional services to assist developers and silicon partners to optimize their products for design and cost efficiency.

The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive license of Linus Torvalds, owner of the mark on a worldwide basis.

For the latest product information, call us or visit: www.mentor.com/embedded

Corporate Headquarters
Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777
Phone: 503.685.7000
Fax: 503.685.1204

Sales and Product Information
Phone: 800.547.3000
sales_info@mentor.com

Silicon Valley
Mentor Graphics Corporation
46871 Bayside Parkway
Fremont, CA 94538 USA
Phone: 510.354.7400
Fax: 510.354.7467

North American Support Center
Phone: 800.547.4303

Europe
Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim
Mentor Graphics (Taiwan)
11F, No. 120, Section 2,
Gongdao 5th Road
HsinChu City 300,
Taiwan, ROC
Phone: 886.3.513.1000
Fax: 886.3.573.4734

Japan
Mentor Graphics Japan Co., Ltd.
Gotenyama Garden
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140-0001
Japan
Phone: +81.3.5488.3033
Fax: +81.3.5488.3004

MGC 01-16 1033810-w