DDC-I

TECH NOTE

SafeMC Technology for Deos

A Solution for Multi-core Environments

SafeMC™ Technology for Deos™, a safety critical real-time operating system for multi-core processors, provides the resource and scheduling mechanisms that enable developers to bound and control the interference patterns which occur whenever processor cores share resources (e.g., cache, memory, or I/O). These patented capabilities enable Deos developers to have full utilization of all cores for safety critical operation, as opposed to artificially forcing the user to designate a single core for safety critical tasks.

Key Features Overview

- Safe Scheduling
- Memory Pooling and Cache Partitioning
- Fine Grain Locking on Kernel Interface Objects
- Safety Nets
  - Memory Throttling

Deos is a time, space and resource partitioned RTOS designed for certifiable, safety critical applications. It has been certified to DO-178 DAL-A since 1998 and is used in avionics functions on aircraft from FADEC's to displays. The Deos multi-core technology provides a solution to the resource contention issues by bounding and controlling interference patterns with some key technologies such as safe scheduling, memory pooling, cache partitioning, file grained locking, and safety nets.

Safe Scheduling

Deos's safe scheduling includes patented technology for the bounding and control of interference patterns created by the shared resources within multi-core processors. Deos uses a two level scheduling model which employs a scheduler per core, which are synchronized across all cores. Using a scheduler per core bounds or eliminates cross core contention when scheduling tasks on individual cores. At the first level, time is managed in windows which are aligned across all cores. Within a time window (second level) schedulers (i.e., ARINC-653, Deos RMA, POSIX) are assigned to each core, and then applications are assigned to one, or more, of the schedulers. Each one of these schedulers can schedule multiple tasks in multiple processes/partitions. The goal of this architecture is to allow the developer to orchestrate the scheduling of applications running on different cores and different windows such that they have limited interference with each other. Safe Scheduling offers a configurable approach of a single RTOS instance managing all the cores and provides user control over the co-scheduling of tasks among the cores. As shown in Figure 3, this scheduling environment allows high-DAL applications to run simultaneously across all cores by enabling the system integrator to configure the system such that resource contentions are minimized (and bounded) between processor cores. For example one could place a memory intensive application on one core and compute bound applications on other cores (which then limits, and bounds, memory resource contention).

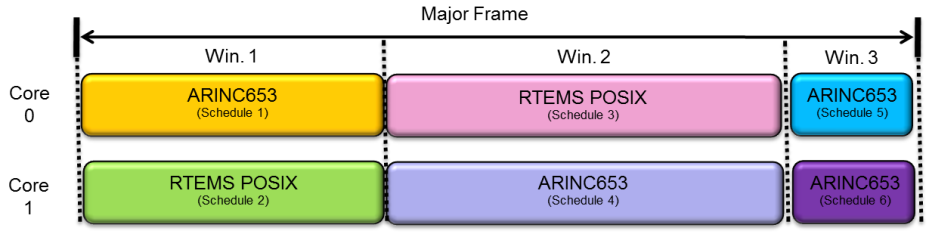

Figure 3 - Safe Scheduling

The Deos SafeMC safe scheduling includes several other features that enable higher processor utilization and flexibility, such as:

- **Slack Scheduling Through Sliding Time Windows –** This feature enables time windows to compress, expand or slide based on if the scheduled tasks in the partition within the window allows it.
- **Hardware Performance Monitor Events –** This capability enables the developer to take advantage of performance monitors that are available on some hardware architectures. The developer can setup threshold values for activity on the memory bus, thermal profile, etc., and have Deos perform scheduling activities that limit a particular applications behavior.

Memory Pooling and Cache Partitioning

Memory pooling and cache partitioning is a configurable cache partitioning capability that enables developers to isolate cache at the application or partition level (not just at the core level) via XML based configuration files (not based on hardware features). This patented Deos feature substantially reduces one of the major sources of resource contention, namely access to cache and system memory.

Deos cache partitioning is a capability that enables the user to isolate separate areas of cache for different applications. Since no other application will use this area of cache it negates the need to flush cache between partition context switches (which impacts Worst Case Execution Time WCET and is often required for high DAL systems in both mono-core and multi-core systems).

The Deos memory pooling feature allows developers to define where applications run in physical memory. Depending on the processors and architecture considered, memory pooling can be setup to use on-chip high speed RAM and run specific critical applications from that memory for applications that need to maximize performance.

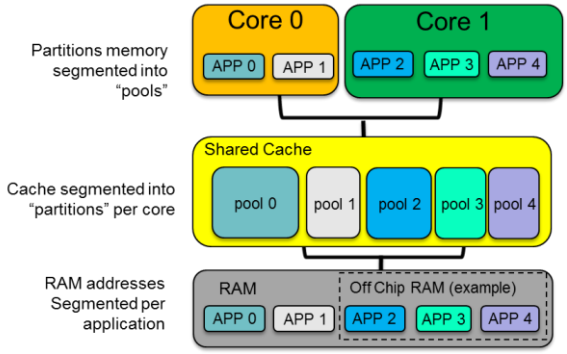

Figure 2 - Memory Pools and Cache Partitioning

Fine Grain Locking on Kernel Interface Objects

Deos is designed to not have any cross core couplings (e.g., contention) by using multiple schedulers and employing fine grain locking on kernel interface objects. Since kernel resources are not shared across cores, there is no cross core locking or the associated impacts on performance.

Safety Nets (Memory Throttling)

A safety net is required according to CAST-32A to contain unintended functionality. Deos implements a method of contention throttling that uses the resources available on the hardware platform designed to monitor hardware behavior.

Many of today's MCP processors have a performance monitor built into the chip that allows the application developer to setup and monitor activity that occurs on the device. Deos supports the ability to interface with the performance monitor and based on the behavioral limits that the system developer defines, disable the scheduling of a misbehaving partition.

Performance Monitoring is a hardware specific capability provided by the individual hardware manufacturers, and thus constrains the application to the hardware specific capabilities of these chips. Unlike the other features of Deos SafeMC (cache/memory pool partitioning, safe scheduling, etc), it is hardware dependent.

Note: The details of meeting the objectives on CAST-32A are available in DDC-I's Certifiable Safety Critical Multi-core Solution for Avionics white paper.

Summary

To summarize, MCPs have become ubiquitous. However, they present significant challenges for the developers of certifiable, safety-critical applications. If one wishes to use an MCP in such applications, bounding and controlling interference patterns on shared resources, and effectively managing CPU utilization are essential. Without the first capability, certification of safety-critical software is impossible. Without the second, much of an MCP's increased computing power is wasted.