

Migrating Medical Devices to Multicore

Justin Moon

As medical devices become more complex and offer increasingly sophisticated user interfaces and capabilities, these devices are inevitably moving to multicore processing. Today, ultrasound, MRI, CT, PET, and digital X-ray systems all imply multicore processing. It is not a question of if, but of when. In addition to all the other challenges they must face, medical device manufacturers must ensure that their multicore systems obtain FDA and other relevant certifications. This requirement has made many wary of multicore systems, concerned that their complexity might jeopardize certification. OS characteristics and features that can facilitate migration to multicore systems, as well as the question of validating multicore systems for certification are discussed.

The Case for Multicore Processing

The growing need for medical devices with new and more sophisticated capabilities, including complex user interfaces and high definition imaging (which both have voracious appetites for computational power) is feeding an increasing demand for multicore processing in medical systems.

While supporting these advanced features, many must also satisfy requirements for low weight, low power consumption, and low heat dissipation. These requirements are very difficult to meet with only one core, as they conflict with the laws of physics.

A generation ago, silicon vendors increased computing power by introducing bigger processors running at higher clock rates. This approach brought single-core development to three insurmountable limitations:

- 1) power consumption and heat generation increase more rapidly than clock rates,
- 2) memory speed does not keep up with processor speeds, and
- 3) resource, control and data dependencies limit the parallelism possible with a single instruction stream.

Multicore processors were developed to directly address these limitations. Since they provide much greater processing capacity per ounce, per watt, and per square inch than single-core processors, they have taken center stage in the product line-ups of major semiconductor vendors, who now offer chips with two, four, or even greater numbers of integrated cores.



With growing demands for computing power, even a relatively small medical device may benefit from multicore processing

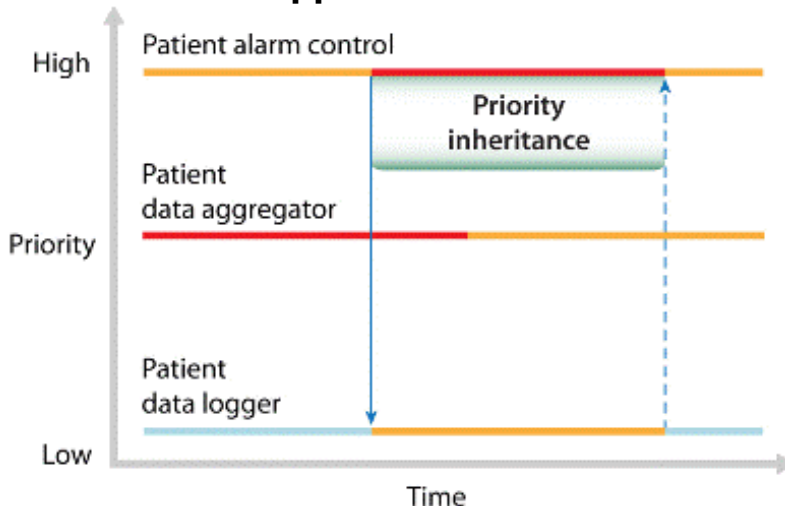
Implementing Multicore Processing

Device manufacturers implementing multicore systems are not limited to the mutually exclusive choices of either

- a) migrating an existing single-core system to multicore, or
- b) designing a new system from scratch.

In practice, it may be more prudent and more efficient to migrate some parts of an existing system, while re-designing others. Whatever the specific strategy selected, careful OS selection will be crucial to the project's success.

OS Multicore Support



Priority inheritance ensures that the data aggregator cannot block the alarm control when the alarm control calls the data logger

Any medical device for which reliability and availability are requirements will most likely require a real-time OS (RTOS). The RTOS should support symmetric multiprocessing (SMP), since running a single OS that can distribute processes across multiple cores is easier to work with than asymmetric multiprocessing (AMP),

which uses distinct OS instances (and even different OSs) on each processor.

The selected RTOS must prevent faults from permeating through the system, and be able to restart components when faults do occur. A microkernel architecture, where applications, device drivers, file systems, and networking stacks all reside outside the kernel in separate address spaces and are thus isolated from both the kernel and each other, is probably the optimal choice. Also of importance, due to the high number of processes and threads in multicore systems, is a fault-recovery (or high availability) watchdog to restart failed processes.

To ensure system dependability, the RTOS should also support:

- time partitioning (preferably adaptive) so critical processes cannot be starved of CPU cycles,
- priority inheritance to prevent priority inversions, and
- processor affinity, especially bound multi-processing, or BMP1, which allows designers to restrict critical processes and their child processes to run on particular, designated cores.

Designers may decide exactly how they want to use the multiple cores available to the system. They may, for instance, assign non-critical but computational-intensive processes, such as HMI and high-resolution image rendering, to run on any available core; or they may choose to run them only on a limited subset of cores, while restricting critical components to one core to simplify design, debugging, and validation of these components.

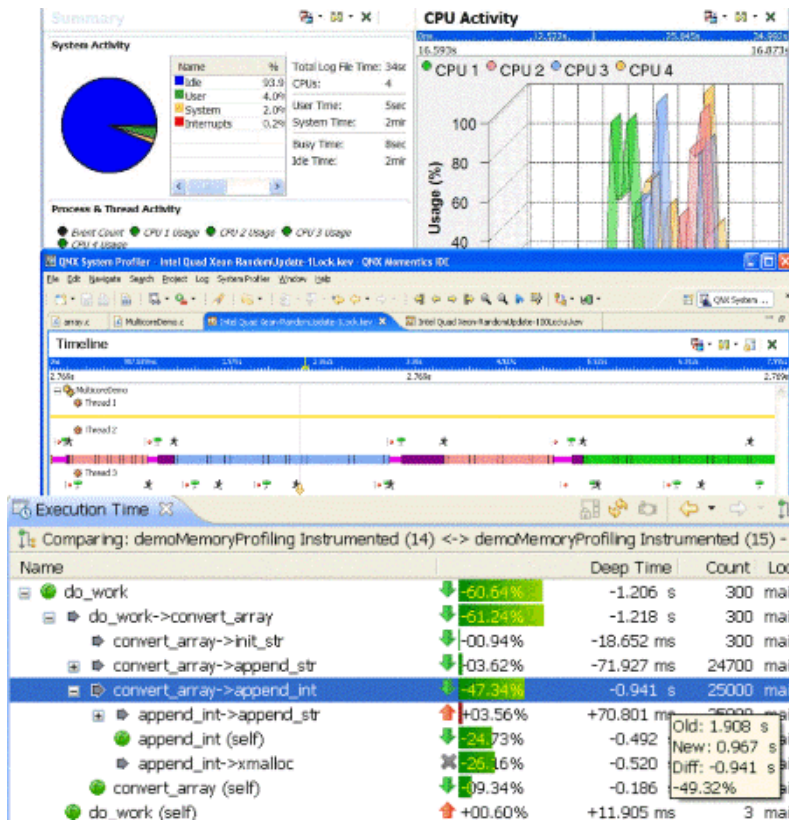
Multicore Development and Validation Tools

Developers need tools designed to support multicore development, debugging, and optimization. The multicore RTOS should be supported by a tool set that allows them to

- assess load balancing
- identify bottlenecks
- identify opportunities for concurrency
- characterize and debug interactions between threads running on multiple CPUs
- move threads or processes between cores, and
- measure the resulting impact on the system.

Timing and blocking can be thorny issues, and the tool set should provide developers the means of pinpointing exactly what processors and threads are doing.

Certifying a Multicore Medical Device



Screen shots showing how a system profiler can analyze software behavior on a multi-core system

Compliance is a necessary condition for getting a medical device to market, a [sine qua non](#) condition that must be met before investments can be transformed into revenue. Therefore, in addition to OS characteristics and tool set features, a medical device needs a well-planned certification strategy.

Concern that the complexities associated with multicore systems might hinder certification has influenced some manufacturers' hesitations to adopt multicore processing. It is worth noting, however, that the inherent complexity of single-core as well as multicore systems means that, in practice,

- it is impossible to validate these systems simply by testing them, and
- the same techniques can and should be used to validate single-core and multicore systems.

These techniques are well known to anyone who has taken a system through a certification process. They include:

Testing—though testing can only prove the existence of faults and not their absence, testing is still a key element in any system validation strategy.

Statistical analysis—applied to in-use data for devices or components with a history in the field, statistical analysis can be used to support dependability claims. It can also be used with fault injection during validation to estimate the number of faults remaining in a system.

Fault tree analysis—starting with potential or actual failures, fault tree analysis can be used to deduce possible causes for the failures.

Design verification—undertaken during development and retrospectively with tools such as SPIN and NuSMV, design verification can provide strong evidence to

Migrating Medical Devices to Multicore

Published on Medical Design Technology (<http://www.mdtmag.com>)

[1]. He has worked on the Custom Engineering Team, specializing in BSP and driver development, and on the Automotive Team, where he worked with a number of Tier 1 and OEM automotive companies. Moon can be reached at 1-800-676-0566 or jmoon@qnx.com [2].

Source URL (retrieved on 03/06/2015 - 4:02am):

<http://www.mdtmag.com/articles/2011/07/migrating-medical-devices-multicore>

Links:

[1] <http://www.qnx.com/>

[2] <mailto:jmoon@qnx.com>