

ECN Roundtable - Designing with Multicore Devices

Edited by Alix Paultre



This month's Roundtable question is, "What do you feel is the most challenging aspect of incorporating a multicore processor into a design for the engineer?"

Hers' an answer from Kim Hartman, VP Sales & Marketing, TenAsys Corporation .
www.TenAsys.com [1]



Multicore processors present both opportunities and challenges to today's computer system designers. One of the challenges that designers must face is how to partition applications to make the best, most efficient use of the available processing power. As microprocessors become available with larger numbers of processing units: 4, 16, 32 or more, this challenge will become greater for real-time applications.

General purpose operating systems such as Windows have taken the approach of abstracting the underlying multicore architecture using symmetrical multi-processing (SMP), with schedulers built into the OS that take responsibility for distributing applications across the available computing resources. Given the wide variety of algorithms that the scheduler would need to manage in order to fully optimize the usage of each processor core, SMP algorithms don't do a perfect job of this, frequently leaving some available processing bandwidth unused. Furthermore, the execution of some applications can be interrupted or slowed down while the scheduler adjusts the computing resources as necessary. These periods of uncertainty may not cause problems in commercial applications running on a server, but in a system that requires real-time, deterministic responsiveness, these types of "mishaps" are not acceptable, and they can be catastrophic.

Designers of real-time deterministic applications need to have control of how the available computing resources are assigned. In addition, as real-time applications span across the multiple cores of a multi-core processor, or even span across a network of distributed processor platforms, the application segments need to interact with each other in a predictable manner. The solution is to employ a distributed RTOS that supports the ability of distributed application threads to communicate with each other to provide a seamless environment, as if the application were running on one processor. At the same time, the distributed RTOS should enable parts of the application to be completely segregated in order to ensure their integrity.

Source URL (retrieved on 11/23/2014 - 8:32pm):

<http://www.ecnmag.com/blogs/2010/12/ecn-roundtable-designing-multicore-devices>

Links:

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