eVM™ Virtualization Platform for Windows®

Host your Embedded OS and Windows on a Single Hardware Platform using Intel® Virtualization Technology

April, 2008
Executive Summary

Starting from scratch is a luxury most embedded projects cannot afford. Development schedules are short, feature requirements are many, and the complexity of embedded systems continues to grow. Solutions built on a base of proven software are often the fastest and most reliable road to success. But how does a developer add features to existing real-time software, code built from years of effort in refinement and testing, without disturbing the underlying reliability and performance of that embedded software?

Embedded virtualization manager provides the solution

The TenAsys eVM™ platform, an embedded virtual machine manager, built on Intel® Virtualization Technology (Intel® VT), hosts your legacy application and OS in an embedded virtual machine while simultaneously supporting the Microsoft® Windows® OS, on a single multi-core PC-compatible platform.

By running embedded guests on dedicated CPU cores, the eVM system provides a means to easily migrate from outdated hardware to modern embedded platforms. To ease the transition, legacy I/O can be emulated, in order to minimize rewriting proven software. For example, an obsolete ISA system can be migrated to a smaller and less expensive single-board computer by redirecting access to ISA peripherals to equivalent on-board PCI devices.

The eVM platform accommodates your hardware (access to I/O) and timing (interrupt latency) needs by giving the embedded guest direct access to time-critical hardware. This is done by assigning I/O exclusively to each guest OS—accommodating the use of existing native device drivers.

The notion of assigning I/O exclusively to a virtual machine is essential to guaranteeing real-time responsiveness.

Multi-core Intel VT processors support real-time virtualization

By utilizing multi-core Intel VT processors to host your embedded guest alongside Microsoft Windows, eVM software is able to partition resources, such as CPU cores, RAM, interrupts, and I/O, between multiple operating systems. Only shared and emulated resources need to be virtualized. Every operating system running on the platform has direct and full-speed access to its dedicated physical I/O, interrupts, RAM, and CPU cycles.

Low interrupt latency, direct access to specialized and critical I/O, and guaranteed ownership of a CPU core deliver near native performance to embedded guests using the TenAsys eVM solution—without compromising the performance and features of Windows and its applications.

Partitioning resources insures that only the authorized OS will have access to its time-critical I/O. For example, the eVM system can insure that the main operator display is only accessible to Windows and that a fieldbus interface is only accessible to the guest RTOS.

Consolidation reduces cost and increases performance

The net gains for hosting your embedded application on the eVM platform, alongside Windows, is the elimination of redundant computer and communication hardware, simpler and faster communication and coordination between embedded guests and Windows, improved reliability and robustness, re-use of proven application code, and simplified development and debugging. Systems that previously required multiple discrete computing platforms can be combined onto a single hardware platform at near native performance levels—saving significant design, manufacturing, and maintenance costs.
Virtualization for the Embedded IA Platform

Developers across the wide spectrum of embedded computing applications can benefit from the latest advances in multi-core processor technology. Intel® Architecture (IA) processors with multiple CPU cores offer a solution to the need for combining new features with legacy code, and for combining multiple operating environments on a single system. By exploiting Intel Virtualization Technology (Intel® VT), a highly-integrated system can be constructed from the combination of an embedded or real-time operating system (RTOS) and a general-purpose OS (GPOS), such as Windows, each running on separate cores.

Embedded virtualization manager enables the solution

Building such a system requires a unique class of virtualization management software. The TenAsys eVM™ platform is such a system and has been designed to host multiple operating systems without compromising performance or determinism. This is done by assigning critical I/O and CPU resources exclusively to a designated guest OS; in other words, by partitioning resources and virtualizing only those devices that are shared between guest operating systems or must be emulated, such as legacy I/O devices.

By giving each guest OS direct access to its time-critical hardware and dedicated CPU core(s), the TenAsys eVM solution is able to accommodate the hardware (access to I/O) and timing (interrupt latency and execution speed) needs of both an RTOS and a GPOS on a single hardware platform. This notion of assigning I/O and CPU resources exclusively to a virtual machine is essential to guaranteeing real-time responsiveness.

Hosting an RTOS in a virtual machine is quite different from executing a real-time thread scheduler, in the form of a device driver or subsystem, inside the Windows kernel. Device drivers and subsystem solutions lack the context to apply Intel VT for the isolation, protection, and independence of operation needed between your embedded OS and the Windows operating system.

Conventional VMM shortcomings

Historically, virtual machine management (VMM) software has emulated an entire machine, providing a general-purpose set of I/O devices that are adequate for many server and desktop applications, at the expense of special-purpose I/O devices that may be unique to your embedded application. The general-purpose VMM handles all I/O transactions on the guest’s behalf. Server VMM products are targeted at solving corporate IT network problems: they maximize the use of server resources and simplify the deployment and maintenance of hundreds of client desktops.

The eVM partitioning technique succeeds in a real-time embedded environment where the general-purpose server VMMs simply does not work. Like a general-purpose OS, such as Windows or Linux, a general-purpose VMM must share I/O resources among the virtual machines and, therefore, is not capable of supporting the deterministic scheduling and dedicated I/O requirements of real-time applications and their embedded operating systems.

Exclusivity for real-time

To maintain the determinism of a guest RTOS in a virtual machine environment, the TenAsys eVM platform distinguishes between resources that can be multiplexed and those that are exclusive to an OS. For example, devices like a console, disk, and an enterprise Ethernet interface might be shared among all hosted operating systems. However, hardware that is specific to a control application, such as a video capture card or fieldbus interface...
cannot be multiplexed between operating systems. Such specialized I/O needs to be
dedicated to its embedded virtual machine, so the embedded guest OS and application
using that I/O can maintain determinism and control.

Similar arguments can be made regarding access to RAM. In a conventional VMM some or
all of the memory in each virtual machine may be swapped to disk. An RTOS, and all of its
real-time applications, can never be “paged out.” The TenAsys eVM platform guarantees
that each embedded OS is locked into dedicated physical RAM, with a dedicated CPU core,
and is never swapped to disk. This is the only way to insure that every real-time event is
serviced consistently and deterministically.

**Exclusivity for performance**

Exclusive access to hardware resources doesn’t apply only to an embedded guest OS.
Graphics intensive Windows applications and video drivers, for example, need access to
real hardware to maximize functionality and performance. A virtual frame buffer is too slow
and inadequate in its feature set to render complex 3D moving images; such applications
require direct access to the frame buffer and its control I/O.

The TenAsys eVM virtualization platform provides Windows with direct access to
all resources allocated to its device drivers and applications—Windows runs
unmodified, at full speed, without loss of features or access to real hardware.

Dedicating the physical video frame buffer and respective control I/O to the Windows
virtual machine guarantees optimum video features and performance for Windows
applications and drivers. To accommodate the lack of a frame buffer for the embedded OS, a
virtual serial console is included as part of the embedded guest virtual machine where the
output is redirected to a standard Windows console, or “text box.”

**Intel® Virtualization Technology**

Until the introduction of Intel® Virtualization Technology (Intel® VT) as part of the
Intel® Core™ microarchitecture, a software-only VMM had to overcome some significant
challenges. Both the VMM and each guest OS expect to maintain supervisor-level control
over the hardware platform. Absent some form of cooperation between the VMM and each guest OS (usually referred to as para-virtualization), the VMM must resort to trickery. Supervisor-level control can be reliably maintained by only one entity, resulting in a conflict between the VMM and a guest OS. The tricks a VMM must use, without Intel VT, can include modifying the guest OS binary code and running each guest operating system at a ring levels for which they were not written.

The downside to VMM trickery is a decrease in performance and limited guest OS compatibility. For example, binary files of a guest OS might be modified to trap supervisor-level CPU instructions, requiring the VMM to emulate these instructions. Instruction emulation slows down the execution speed of the guest OS, and the need to “fix up” binary files limits your guest OS options to those that have been certified for use with the VMM. Likewise, problems exist in the area of address translation and interrupt masking where, again, pure software solutions invariably result in performance overhead that cannot be tolerated by many real-time and embedded applications.

Intel Virtualization Technology is designed to overcome the problems described above. In those processors that include Intel VT an overarching operating-mode has been added, called VMX root, where the TenAsys eVM software executes. eVM software intercepts key operations executed by a guest OS without requiring a priori knowledge of the guest OS binaries or internal operational techniques. Intel VT also provides eVM software with the means to implement virtual interrupts, essential for partitioning hardware resources.

Application Scenarios

Unlike the application of a conventional VMM to an IT server platform, the TenAsys eVM platform is best suited for control and data acquisition applications that require determinism, performance, and security and would also benefit from the use of a general-purpose operating system, such as Windows or Linux.

Legacy RTOS support

Many valuable and useful systems exist today that depend on embedded code written years ago. These legacy applications are in use because they work; they are proven, they are reliable, and they may even be certified for a specific application (e.g., medical, defense and aerospace). Unfortunately, these legacy applications may also be running on outdated expensive hardware, or be in need of a critical update, such as a modernized user interface or a requirement to be accessible over an enterprise network. Rewriting proven and/or certified applications is rarely desirable or economical.
The effort to migrate to new hardware or enhance existing embedded applications can be significantly reduced by hosting your legacy OS and its real-time application(s) on the TenAsys eVM platform. Legacy I/O can be virtualized, to ease the migration from outdated hardware to a modern more cost-effective platform. Inter-process communication, facilitated by the eVM system, can be used to augment system functionality with Microsoft Windows or another operating system.

Usage examples

Consider a manufacturer of machine tools that has spent over ten years developing real-time machine control software on QNX. Many of the original engineers that developed the software are no longer employed by the company. The machine tool manufacturer wants to upgrade the user interface by adding Windows, without risking changes to the real-time control software. The solution is to run the existing QNX software on one core of a multi-core processor and implement the new user interface functionality on the remaining core(s).

The eVM system, in this case, partitions I/O, memory, and other resources to insure that only the machine control software has direct access to the motion control hardware and that it operates on a dedicated CPU core. This insures a separation between the RTOS and Windows, thereby protecting the timing loops managed by the real-time software from being violated by operations performed by the new Windows user interface software.

A manufacturer of a robotic material handling system wants to lower the cost of their product built on multiple hardware computing platforms; an industrial PC running Windows and two custom PowerPC boards running vision and motion control software on the VxWorks RTOS.

The vision and motion software use a library that is also available for Intel Architecture (x86) processors, and are recompiled to run as VxWorks tasks on dedicated cores of an IA quad-core processor, while the Windows-based operator interface runs on the remaining cores of the same processor board. What was previously an expensive, loosely coupled system of multiple discrete hardware boards is now a single, cost-effective, and highly integrated hardware platform.

Conclusion

The net gains from the application of the TenAsys eVM virtualization platform on embedded Intel multi-core processor platforms are the elimination of redundant computer and communication hardware, faster communication and coordination between embedded OS and Windows subsystems, near native performance levels, improved reliability and robustness, re-use of proven code, and simplified development and debugging. Systems that previously required multiple discrete computing platforms can be combined onto a single hardware platform, saving costs in design, manufacturing, and maintenance.

Multi-core processors can easily support multiple operating systems and high-performance, low-latency, real-time applications by dedicating a CPU core to the RTOS. The CPU instruction cycles of the RTOS core are available exclusively for use by the real-time applications. Contention for key resources, such as CPU cycles, pipelines, and the FPU, are avoided. TenAsys eVM software uses Intel Virtualization Technology to eliminate resource contention by isolating and dedicating I/O devices for exclusive use by their respective guest OS. Hardware that can be shared, like keyboard, mouse, and enterprise network interfaces, are presented as virtual devices for use by all operating systems.