Technology (Intel® VT) provide device manufacturers opportunities and options to ensure that the varying priorities of device traffic are not compromised when moving to a virtualized platform. Intel® VT hardware features such as Intel VT-x, Intel VT-d, Intel VT-c, and SR-IOV are crucial steps forward in ensuring that the priorities of data traffic received from external devices are retained in products hosting multiple different workloads.

Embedded virtualization and Intel® VT hardware-assist for virtualization allow vendors and developers to migrate existing designs to hardware offering these features. Migrating existing designs to new hardware platforms that support Intel® Virtualization Technology is a compelling objective, but detailed investigation is required to find the optimal configuration necessary to maintain overall product responsiveness.

This paper offers insights on how to migrate existing designs to new hardware and add new and differentiating product functionality while resolving the challenges of enforcing device responsiveness by leveraging embedded virtualization, multi-OS separation and device partitioning.

Multiple Operating Systems on a Single Multi-core Device:

Many embedded systems today use multiple operating systems on different processor boards, each offering services uniquely hosted by that operating system. Embedded virtualization offers developers the opportunity to consolidate hardware by running multiple OSes in partitions on a single CPU.

Running multiple OSes on a single embedded multi-core device allows developers to take full advantage of the unique benefits of more than one individual operating system. For example, they are no longer constrained by limited graphics libraries in a real-time OS when developing a human-machine interface (HMI). Embedded virtualization and multiple operating systems allow developers to maintain the real-time behavior of the device while adding other components such as standards-based...
communication protocols or enhanced graphics, well supported in general purpose OSes.

There are challenges when considering porting existing operating systems and applications to new hardware atop of embedded virtualization. As shown in Figure 1, supporting multiple operating systems on a single device implies that there are multiple aspects of functionality being handled by the various operating systems; functionality that is likely orthogonal to one another. This implies that the interactions with the outside world are vastly different for each operating system.

An important result of vastly different functionality serviced by different operating systems is that the streams of data directed to the embedded device have different priorities. This means that the streams of data generated by the equipment being serviced by the real-time operating system needs to be processed with different priority and latencies than the stream of data being directed to the general-purpose operating system.

Leveraging multiple operating systems to extend the functionality of an embedded device is a compelling objective with measurable benefits, at both the technical and business levels. Doing so successfully, requires careful planning and consideration to ensure that access to the various devices, the differing data streams, is handled in a manner which supports and maintains the differing priorities of traffic.

Virtualization and Partitioning

There is a strict distinction between virtualization and partitioning:

Virtualization is a computing concept in which an OS runs on a software implementation of a computer, or “machine”—that is, a virtual machine (VM). Multiple VMs can run on a single hardware device, allowing multiple operating systems to coexist. The VMs are managed by a virtual machine manager, also called a hypervisor, which provides abstraction between the underlying physical hardware and the VMs. It can also provide communication between VMs if required as well as security and reliability (for example, one VM could crash but without affecting the rest of the system).

Partitioning refers to the need and ability to isolate certain devices and aspects of the physical hardware and present these devices or components to a specific virtual machine. When architecting an embedded product, certain devices need to be specifically presented to a particular VM, while at the same time, not detectable by other VMs. As operating systems go through their boot phase detecting hardware components, it is necessary that certain devices only be accessible by operating systems in VMs in which the devices are being handled and supported. For example, it would be less than desirable for a Windows 7* instance to try to manage and control sensors and actuators of an industrial robot rather than networking or graphics on-board devices.

This distinction between virtualization and partitioning is a key aspect when crafting an embedded virtualized device.

Hardware assist with Intel® VT features:

Until recent virtualization hardware-assist features emerged in CPUs, there needed to be a sizeable amount of work involved to get multiple OSes to collaborate with one another and with the hypervisor; there was more paravirtualization effort required to achieve both the virtualization and partitioning of devices on the physical board. This
sizeable amount of effort can lead to undesirable results. With increased instrumentation (paravirtualization) of the operating systems and integration with the hypervisor, the behavior of the virtualized real-time partitions is less deterministic. Also, more instrumentation increases the overall system complexities when debugging the virtualized system. Tracing system calls or board bring-up issues with a sizable hypervisor can be a significant challenge.

Intel has enhanced the capabilities of virtualization technology with a hardware-assist technology called Intel® Virtualization Technology (Intel® VT). Intel® VT performs various virtualization tasks in hardware, like memory address translation, which reduces the software footprint of the hypervisor and improves its performance and overall determinism.

Wind River® Hypervisor takes advantage of Intel® VT features to provide optimal virtualization performance and increased reliability. Without this new technology, the hypervisor would be responsible for handing off more of the platform control to the operating system, requiring complex modifications to the operating system.

Intel® VT technology includes hardware enhancements that shift much of the burden of software-based virtualization into the hardware. This simplifies virtualization implementations by reducing software overhead, making it easier for device developers and architects to create designs that maintain real-time performance in a virtualized embedded device. It also helps make virtualization more efficient and secure in general, and significantly improves performance — to near native levels, depending on the virtualization model.

**Benefits of hardware-assist features**

Migration of operating systems and software across multigenerational hardware upgrades has always been a challenge for embedded customers. Very often the existing software infrastructure grew inseparable ties with the original hardware platform by means of industry audits and certifications and the resulting cumulative software changes. Severing these hardware and software ties sometimes becomes an impossible task. Using Intel® Virtualization Technology – Intel VT-x processor virtualization features, migration of operating system and software to next generation hardware platform becomes much simpler. Because Intel VT-x technology enables the processor hardware to manage per guest Extended Page Tables (EPT) so the hypervisor does not have to maintain resource intensive software shadow page tables and interrupt maps, it allows fully unmodified operating systems to reside as guests with near native performance.

Moreover, because the hypervisor layer arbitrates most legacy transactions with hardware, it is possible for the hypervisor to recognize and manage OS or software specific nuances to ensure full OS – next generation hardware compatibility. Because fully unmodified OSes can now reside as guests atop the hypervisor, the need for revalidation and recertification of the original software/OS code can be virtually eliminated.

Another clear advantage of using Intel® VT technology on the newest generation Intel® Architecture system is the performance boost the new hardware can provide over the legacy infrastructure. In some embedded scenarios, where upgrades span multiple generations, there can be a significant boost in performance even with the additional hypervisor layer required for virtualization.

Another key feature of Intel® VT-based hardware virtualization is the ability to isolate hardware (cores, interrupts, I/O) that provides guests with direct access to the resources needed. Using Intel® VT, it is possible to assign individual cores, interrupts, or even individual I/O devices directly to a suitable guest kernel, with hardware acceleration at near native performance levels. Because the system resources can be isolated in hardware to distinctive OS/kernels, it enables enhanced routing of external information (via I/O) and highly robust security features. Intel® VT hardware assist and acceleration feature allows for the compartmentalization of a system into critical and non-critical buckets as shown in Figure 2.
The user has the freedom to fine tune the necessary system bandwidth so that the system resources are distributed fairly, depending on the priority and importance of the tasks being performed in each bucket. Since system consolidation is the key value proposition behind virtualization, with Intel® VT hardware assist features, such consolidation can be easily achieved without affecting system performance or hypervisor complexity. Moreover, Intel® VT allows for compelling OS and software scalability across multiple generations of Intel hardware.

Intel VT-x:

Intel VT-x technology enables multiple guest operating systems to efficiently reside on the same system while providing hardware assists for guest level memory, interrupt and I/O device management. Traditionally, in software only virtualization implementations (without Intel VT-x), the hypervisor is responsible for trapping and emulating almost all guest access to memory, interrupts and I/O transactions. With increasing guest loads, these trap and emulation tasks would require an increasing number of hypervisor cycles adding more latencies, further reducing efficiencies and benefits of virtualization. Using Intel VT-x hardware assist technology, the hypervisor can now rely on hardware instructions on the CPU to help mitigate most of these trap-and-emulate latencies by taking advantage of available hardware resources. Additionally, in newer generations of Intel® Architecture platforms, Intel has introduced features such as Extended Page Tables (EPT) and VPID (Virtual Processor IDs), which enable per-guest tracking and tagging of page tables in memory using hardware resources, eliminating the need for the hypervisor to maintain latency intensive shadow page tables.

Intel® VT-x hardware accelerators trap and execute sensitive instructions relieving the hypervisor of these duties. By performing these instructions directly, Intel VT-x:

- Greatly reduces the overhead of virtualization
- Improves system responsiveness
- Makes virtualization operations more reliable and secure

Intel VT-d:

Intel VT-d technology allows I/O devices to directly transfer bulk data (DMA) into a guest’s memory space. Intel VT-d also enables direct assignment of I/O devices into guests’ memory ranges, bypassing the hypervisor therefore allowing the native drivers of the guest OS to communicate directly with the hardware. All guest memory references and translations are tracked and maintained in hardware logic thereby unburdening the hypervisor from these tasks. By enabling hardware-arbitrated partitioning in I/O to guest communications, Intel VT-d provides isolation between guests and I/O devices. This also reduces the possibility of rogue or malicious transactions to unsecure guests or partitions.

Intel VT-d decreases the load on the processor and accelerates data movement throughout the system. Intel VT-d also provides significant performance boosts for high-bandwidth applications. In addition, Intel VT-d enhances I/O security and availability because data intended for a specific device or VM cannot be accessed by any other hardware or VM.

Intel VT-c and SR-IOV:

Intel VT-c technology enables Intel network devices with dedicated hardware resources (such as data queues or VMDqs) for parsing and routing of guest...
specific bulk or streaming data to be channelled directly into a specific guest’s memory space without the requirement of hypervisor oversight. In addition, Intel VT-c facilitates the sharing of a single hardware network device by converting it into multiple virtual network resources in compliance with the PCI-SIG Single-Root I/O Virtualization (SR-IOV) standard. In this scenario, the physical network device presents multiple virtual ports that can then be directly assigned to guest OSes, providing faster network data throughput and further reductions in hypervisor latencies and cycles.

To take advantage of Intel® VT, a virtualization solution requires the following:

- Processor with Intel VT-x (available with specific Intel® Xeon®, Intel® Core™, and Intel® Atom™ processors)
- Chipset with Intel VT-d (available with specific Intel® Xeon™ and Intel® Core™ processors)
- Intel® Ethernet Network Adapter with Intel VT-c (optional)
- VMM/hypervisor software (available from Alliance Associate member Wind River and others)
- Intel® VT-enabled BIOS (available from Alliance Affiliate members American Megatrends, Insyde Software, Phoenix Technologies, and others)

How to migrate existing designs to Intel® VT using Embedded Virtualization

Intel® VT speeds up the transfer of platform control and the movement of data between different operating systems running on a virtualized platform. Intel® VT increases system performance and reliability by lowering the workload on the virtualization software (i.e., VMM). With virtualization, embedded developers can more easily migrate legacy software, consolidate functions and applications, and increase real-time performance of embedded products.

Intel® VT hardware-assist features for embedded virtualization enable legacy, existing applications to be migrated to new CPUs with minimal, or no porting efforts; it is not necessary to instrument operating systems in order to properly execute on a hypervisor layer. The hardware provides support for virtualization allowing the existing operating system to run, unmodified, on the new hardware.

When migrating existing applications to new compute platforms, it is necessary to pay specific attention to the devices on the hardware, to ensure that the existing application and operating system have the expected access to the device(s) to ensure that the application behaves as required. When adding secondary or multiple operating systems to an embedded device, further attention must be paid to ensure that there is no contention in device access.

Use Case: Adding Windows 7* to existing designs for increased functionality while maintaining data stream priorities

Virtualization in the IT industry focuses on hardware abstraction to virtualize access to all devices on the host server to maximize guest OS consolidation and provide homogeneous host environments. This results in compute platforms that appear identical to all guest OSes regardless of the physical host and its hardware. Virtualization in the embedded industry focuses on a different set of benefits. The operating systems in an embedded product need to collaborate to deliver the complete functionality of the product. Each operating system uses its own subset of hardware devices, memory, and processing cores and needs to communicate with the other operating systems in the device. All of this usually has to operate within tight memory limitations and adhere to strict timing requirements and sometimes needs to be certified to certain safety standards.

Embedded virtualization can be adopted to utilize a mixture of different operating systems to build an embedded device. As already noted, it may make more sense to manage and control robotic sensors and actuators with a real-time operating system while graphics and networking aspects of the product would be better supported by a general purpose operating system offering improved graphics support and connectivity. This is shown in Figure 3.

[325933-001]
When architecting multi-OS embedded products, the product’s architect or designer is faced with several challenges:

- Configuring and directing device access among multiple operating systems to ensure that there is no contention
- Partitioning or virtualizing a subset of devices among multiple operating systems
- Maintaining deterministic timing behavior when moving to a virtualized environment
- Migrating existing applications to new multi-core CPUs without making widespread software changes

Wind River Hypervisor combined with Intel® Virtualization Technology resolves these challenges and provides the architects the ability to seamlessly migrate existing applications to multi-OS designs on multi-core hardware.

It is imperative that the real-time VM and application continue to maintain bare-metal access to the devices that the real-time application is managing. Providing virtualized device access to a real-time critical application may introduce unacceptable latencies.

**Hardware Selection**

The starting point for building a virtualized device that hosts both real-time and general-purpose operating systems is to select a hardware platform that contains both the devices necessary for all operating systems, and an Intel® Architecture platform that provides hardware assist features for virtualization such as Intel® vPro™ Technology compliant systems which includes Intel® VT support.

Figure 4 shows a generic hardware platform with Intel® Virtualization Technology and some standard devices for this example.

Hosting both Wind River VxWorks® and Microsoft Windows 7* on the same hardware platform requires decisions to be made with regard to which devices are presented to which operating system and which devices are (or need) to be shared between OSes.

![Figure 4 - Selection of Hardware Platform](image)

Hypervisors provide virtualized access to memory, CPU cores, and board-level devices. This means that the CPU cores and system memory are virtualized and presented to each guest OS. Memory is partitioned and virtualized in such a manner that each guest OS has full read/write access to a subset of the physical memory range but sees that subset as the complete range of memory available on the system. That is, the memory that the hypervisor presents to the guest is seen by the guest as the full range of memory on the system, as the guest sees the system. If needed, it is possible to provide guests with access to shared memory, if the application requires memory to be shared among the guest OSes. This latter case represents memory virtualization whereas the former represents partitioning.

**Partitioning and Virtualizing**

As further shown in Figure 5, devices that may be partitioned directly to the Windows 7* Virtual Board include the Ethernet device(s) and the IDE port for access to a local hard drive. This example is also

1 Intel® VT support may also be available Intel® Xeon® and select Intel® Atom™ processor systems.
2 Intel® VT may also be available in other select Intel® Core™ i7/i5/i3* processor based systems; *Intel® Core™ i3 processor and Intel® Atom™ processor systems do not support Intel VT-d.

---

* Intel® VT: Intel® Virtualization Technology
* Intel VT-x: Intel® Virtualization Technology for IA-32, Intel® 64 and Intel® Architecture
* Intel VT-d: Intel® Virtualization Technology for Directed I/O
* Intel VT-c: Intel® Virtualization Technology for Connectivity

325933-001
specifically allocating a single core to the VxWorks partition and two CPU cores to the Windows 7* partition.

result of leveraging Intel® Virtualization Technology (using Intel VT-d) – the guest OSes are provided direct access to the devices that they require in order to operate as designed. The guests are also able to continue to use the native device drivers that are offered by default in the operating system. This can be crucial when migrating an existing application to new multi-core CPUs on a hypervisor, as continued use of the same device drivers on new hardware significantly reduces retesting efforts.

**Figure 5 - Virtualizing and Partitioning Devices**

---

**Single Ethernet Device Shared Among Guests**

SR-IOV technology can resolve the apparent limitation of situations where both guest OSes require individual dedicated Ethernet ports, but are hosted on a board on which there is only a single physical Ethernet device.

The SR-IOV specification can be used in conjunction to make a single PCIe device appear as multiple separate PCIe devices which can then be mapped to the individual guest operating systems, allowing each OS to have direct access to a specific “lightweight” PCIe Ethernet device. Support for SR-IOV is required in the Ethernet card, the board’s BIOS, and the hypervisor.

The resulting hardware view as seen by both of the guest operating systems is shown in Figure 6. Each virtual board has CPU core(s), memory, and virtualized access to the timer and serial port.

The board-level devices presented through the hypervisor into the VxWorks virtual board are not detectable by the Windows 7 virtual board – the latter guest does not detect the presence of those devices. This means that it is ensured that there will be no device contention between virtual boards and the Windows 7 guest OS will not try to interact with the real-time devices supported by the VxWorks application.

When partitioning devices in this manner, the resulting impact of virtualization is minimized as a

---

SR-IOV technology can resolve the apparent limitation of situations where both guest OSes require individual dedicated Ethernet ports, but are hosted on a board on which there is only a single physical Ethernet device.

The SR-IOV specification can be used in conjunction to make a single PCIe device appear as multiple separate PCIe devices which can then be mapped to the individual guest operating systems, allowing each OS to have direct access to a specific “lightweight” PCIe Ethernet device. Support for SR-IOV is required in the Ethernet card, the board’s BIOS, and the hypervisor.

The resulting hardware view as seen by both of the guest operating systems is shown in Figure 6. Each virtual board has CPU core(s), memory, and virtualized access to the timer and serial port.

The board-level devices presented through the hypervisor into the VxWorks virtual board are not detectable by the Windows 7 virtual board – the latter guest does not detect the presence of those devices. This means that it is ensured that there will be no device contention between virtual boards and the Windows 7 guest OS will not try to interact with the real-time devices supported by the VxWorks application.

When partitioning devices in this manner, the resulting impact of virtualization is minimized as a

---

**Figure 6 - Logical Devices as Seen by Guest OSes**

This process of partitioning and virtualizing devices, and presenting subsets of board-level devices to guest OSes is performed and controlled through hypervisor configuration.

**Figure 7 - Embedded Virtualization Hosting Two OSes**

---

Hypervisor Configuration
An embedded hypervisor that can be configured to present system-level definitions and hardware device mappings to the virtual boards and guest OSes provides the developer with a mechanism to describe all device configuration in one location, as shown in Figure 8. Such device assignment includes partitioning physical memory ranges and local physical hardware devices, assigning interrupts, and allocating CPU cores to guest OSes. Contrast this with virtualization in the IT industry where most hardware devices are virtualized and visible to all guest OSes for maximal virtualization. Embedded

Intel® VT: Intel® Virtualization Technology
Intel VT-x: Intel® Virtualization Technology for IA-32, Intel® 64 and Intel® Architecture
Intel VT-d: Intel® Virtualization Technology for Directed I/O
Intel-VT-c: Intel® Virtualization Technology for Connectivity
virtualization puts the system developer in control; only he or she can make sure that the partitioning of the system is done in such a way that the final system behaves in the desired fashion.

The system-level configuration that an embedded hypervisor uses partitions the system into multiple virtual boards, each executing a real-time or general purpose guest OS. Based on the configuration presented to the hypervisor at boot time, the virtual board is managed by the hypervisor, and the hypervisor controls several components:

- the cores on which the virtual board executes
- the memory range
- devices the guest OS can access

The memory, PCI attributes, and interrupts can be directly mapped into individual guests. The hypervisor is not involved in the data path to or from the devices. This delivers performance that approaches native non-virtualized performance.

Using a highly configurable hypervisor that has minimal-footprint and real-time responsiveness makes this possible

When architecting mission-critical embedded systems, it is best practice to ensure that code bloat is avoided and software is architected with streamlined function calls for maximum efficiency in both runtime and debugging. With the introduction of a hypervisor in the software layers, it is imperative to select a hypervisor that was developed with similar design principles and requirements.

The code and memory footprint of Wind River Hypervisor is strategically kept to a minimum. This ensures that unnecessary code is not executed as part of a hypervisor-based system runtime, ensuring that the hypervisor has minimal impact on system operations. With minimal code size, applications’ determinism and safety are retained without impact to expected behavior. The virtualization layer is scaled down to the level where it contains only the minimal functionality required to provide its services. Any additional code, such as device drivers, is not contained inside the hypervisor but resides directly inside the guests that require it.

When consolidating multiple operating systems on a multi-core device, and architecting mission-critical virtualized embedded systems, having complete flexibility to describe how applications can access CPU cores, memory, and devices is a critical requirement in order for embedded products to function within the necessary operating parameters.

Summary

Leveraging multiple operating systems to extend the functionality of an embedded device is a compelling objective with measurable and tangible benefits. Having the ability to augment a real-time product with a general-purpose operating system that provides enhanced graphics or standards-based protocols provides device manufacturers with opportunities for rapid product differentiation.
The challenges of such product augmentation can be mitigated when leveraging an embedded hypervisor and an Intel® Architecture platform that are designed to work together and support built-in hardware extensions for virtualization, such as an Intel® VT-compliant system and Wind River Hypervisor. Choosing an embedded virtualization platform that allows the designer the ability to specifically dictate which guest OS has access to which board-level devices will ensure that the embedded product continues to interact with real-time external devices as designed, with no processing impact from additional general-purpose operating systems.

An embedded virtualization development platform that consists of an embedded hypervisor that works in tandem with Intel® VT-compliant platforms is a key component that offers designers the ability to augment existing product designs with additional functionality in a deterministic manner, with the fastest time-to-market capabilities available.

Increase the time-to-market for your products, simplify your designs, differentiate your products, and embrace embedded virtualization with Intel® Virtualization Technology–enabled systems and Wind River Hypervisor.
Legal Disclaimers

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Copyright © 2011 Intel Corporation. All rights reserved.

Intel, Intel Atom, Intel Core, Intel vPro, and Xeon are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel® Virtualization Technology may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel, Intel Virtualization Technology, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2011, Intel Corporation. All rights reserved.