



Achieving Real-Time Performance on a Virtualized Industrial Control Platform

Good for many applications down to the 100 microsecond cycle time range

Introduction

A mainstay in data centers and other server environments, virtualization technology has been instrumental in increasing equipment utilization and lowering capital expenditures. Now, the technology is making its way into smaller systems, like industrial controllers. Accelerating this trend, equipment developers can now consolidate real-time and non-time-critical applications onto a single board, a credible path to streamlined operations, improved productivity, and reduced cost and complexity. This is possible because advances in real-time operating systems (RTOSs), lightweight hypervisors, and hardware-assisted virtualization have significantly boosted the responsiveness of virtualized systems.

Virtualization enables a computing platform to share its resources across multiple workloads, which all run as if they had their own dedicated system. But due to the overhead typically associated with virtualization processes, there is a valid concern that these platforms will be unable to deliver the real-time and deterministic response required by today's advanced automation processes. This paper explores this question by measuring the interrupt latency of the Intel® Industrial Solutions system consolidation series—a virtualized platform for the consolidation of multiple industrial workloads. The test methodology and system configuration are detailed to help equipment manufacturers assess whether this virtualized industrial control platform is responsive enough for their applications.

TABLE OF CONTENTS

Introduction 1
Time-Critical Applications 2
Interrupt Latency and Its Sources..... 3
Minimizing Interrupt Latency..... 3
System Consolidation Platform Overview 3
Performance Considerations 5
Interrupt Latency Testing 5
Conclusions..... 7

Time-Critical Applications

On the factory floor, robots, motion controllers, programmable logic controllers (PLCs), and other manufacturing equipment work together in synchronized fashion as products roll down the assembly line. This necessitates getting everything to work in lockstep, which is generally dependent on real-time and deterministic performance of the equipment:

Real time indicates a system is fast enough to service all critical events, like external and fixed-clock interrupts, within an allotted time. Real time is a relative metric because it depends on frequency of events. When events arrive quickly, a real-time system must be faster than if events come in more slowly. Similarly, a slow system can still be real time if the amount of time between events is relatively large.

Determinism is a measure of the variation in a system's response time to a particular event. One gauge is jitter, as shown in Figure 1, which is the worst case execution time minus best case execution time. An industrial control platform must be able to perform measurements at precise intervals, so its maximum response time should be predictable and bounded. For example, a chemical plant controller reading a sensor before a chemical reaction has taken place—faster than expected—will get the wrong measurement. Faster is not better. Sampling a measurement at non-deterministic or random times will not provide correct results.

The average cycle times and jitter requirements for several industrial applications are shown in Table 1. The worst case execution time (e.g., best case execution time + jitter) must not exceed the maximum cycle time. Motor drive and motion control applications typically have more stringent response requirements than PLCs.

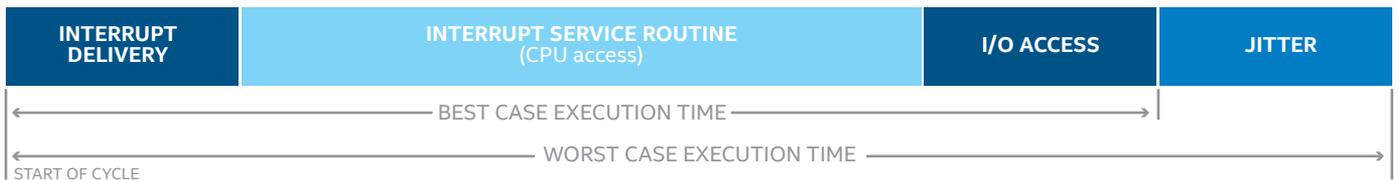


Figure 1. Cycle Time Components.

| Application | Average Cycle Time | Jitter |
|---|--|--------------------------|
| Automation Control PLC (High-speed sensors) | Tens of microseconds to tens of milliseconds | 10 to 1,000 microseconds |
| Motion Control | Tens of microseconds to hundreds of milliseconds | 1 to 5 microseconds |
| Motor Drive Control | Tens of microseconds | 1 to 2 microseconds |

Table 1. Typical Cycle Time and Jitter Requirements by Application.

Interrupt Latency and Its Sources

When an event needs to be processed, a signal in the form of an interrupt is sent to the industrial control system. Intel® processor-based systems employ message-signaled interrupts (MSIs), which write to a special memory-mapped I/O address, thereby delivering interrupts directly to the processor in contrast to traditional methods that assert a pin. MSIs greatly reduce the interrupt latency and processor overhead associated with servicing interrupts, boosting general system performance as well as I/O responsiveness, both critical to industrial applications.¹

Interrupt latency has several main components: the hardware circuits receiving the interrupt signal, the operating system scheduling the interrupt, and the interrupt service routine (ISR) processing the interrupt request. In a virtualized environment, the hypervisor is also a source of considerable latency when it must perform a context switch because a dormant OS received an interrupt.

Minimizing Interrupt Latency

Interrupt latency can be reduced through the use of various products and technologies:

- **Multi-core Intel® processors** with “core affinity” allow application developers to dedicate a processor core to a time-sensitive application so it runs unencumbered by the other software on the system.
- **Real-time operating systems (RTOSs)** improve system response and determinism by allowing software developers to easily prioritize the execution of particular software threads, compared to general-purpose OSs that attempt to treat all the applications running on the system fairly.
- **Type 1 hypervisors**, also referred to as bare-metal hypervisors, have a very small code size and memory footprint, which allows them to run very quickly.
- **Intel® Virtualization Technology (Intel® VT)²** performs various virtualization tasks in hardware, which reduces the overhead and footprint of virtualization software, thus boosting performance and reducing latency.

These capabilities are integrated in the Intel Industrial Solutions system consolidation series: development and production kits designed to greatly simplify the use of virtualization technology in industrial systems. The kits pre-integrate key software and hardware components, thus reducing development time to consolidate multiple factory functions onto a single platform. Original equipment manufacturers (OEMs) can use this design to develop equipment that delivers many benefits to factories, including reduced operating expense, factory footprint, energy consumption, and support effort.³

System Consolidation Platform Overview

The Intel Industrial Solutions system consolidation series integrates a production-ready virtualization software stack with three preconfigured virtual machines (VMs) running a combination of real-time and embedded operating systems, as shown in Figure 2. Developers can utilize the two instances of Wind River VxWorks* RTOS to run applications with real-time performance requirements while simultaneously

Interrupt latency has several main components: hardware circuits receiving the interrupt signal, the OS scheduling the interrupt, and the interrupt service routine processing the interrupt request.

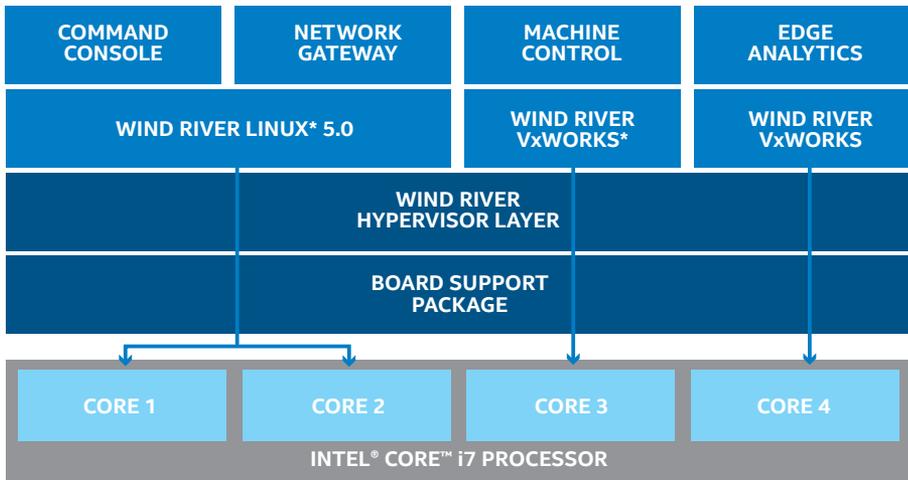


Figure 2. Sample Solution Using the Intel® Industrial Solutions System Consolidation Series.

running embedded and general-purpose applications on Wind River Linux* 5.0. To help developers port their applications to the multi-OS environment, the development kit provides the Wind River Workbench* development environment.

Multi-core Intel® Core™ processors with built-in virtualization technology allow systems to simultaneously run multiple RTOSs and embedded OSs, each on dedicated processor cores using a feature called core affinity. This configuration enables the deterministic behavior of time-critical applications, allowing them to run without being interrupted by non-real-time tasks that would otherwise compete for CPU resources in a non-virtualized system. Developers can build systems with applications running on all the OSs simultaneously or choose to port applications to only a subset of the OSs, depending on their needs.

The solution was tested on a power-efficient, quad-core Intel® Core™ i7 processor with Intel VT and supports common I/O, such as RS-232/422/485, USB 2.0 and 3.0, Ethernet, dual-displays, and mini-PCI Express* sockets. The platform is fanless and ruggedized for use in harsh environments and includes a 128 GB solid-state drive and 16 GB DDR3 memory. The following describes three key platform components designed to decrease latency:

Wind River VxWorks RTOS provides scalability, reliability, and real-time performance to ensure high performance along with deterministic behavior. Key features include a tunable memory footprint, hard real-time performance, state-of-the-art memory protection, advanced multi-core processor support, and extensive connectivity options. Together, these features deliver the capabilities and support demanded by mission-critical, connected, industrial systems.

Wind River Hypervisor* is an embedded hypervisor that consolidates multiple applications and operating systems onto a single multi-core platform to decrease hardware size, weight, power, and cost. It provides low latency, determinism, and real-time performance—all in a small footprint.

Intel VT integrated in Intel Core processors can improve the performance and security of virtualization. This hardware-assisted virtualization technology enhances the capabilities of software-based virtualization technology by performing various virtualization tasks in hardware, like memory address translation. As a result, the overhead from virtualization software is reduced, resulting in higher performance. For instance, memory access time is significantly faster when virtual-to-physical memory address translation is performed in hardware, rather than software.

In addition, Intel VT increases the robustness of virtualized environments by using hardware to prevent the software running in one VM from interfering with the software running in another VM. Along these lines, virtualization helps avoid unintended interactions between applications by preventing one from accessing another's memory space.

Performance Considerations

Developers can improve the performance of virtualized systems by taking the following suggestions into account:

1. Use individual drivers at the VM layer—not shared drivers—to avoid having a point of contention between applications that needlessly results in additional latency.
2. Check for code that causes unnecessary VM exits because they incur a relatively large delay. For instance, consider a polling method instead of hard interrupts for non-real-time processes.
3. Implement MSI interrupts because they are quickly directed to specific cores, and their interrupt vectors are provided to the processor, eliminating the need for an ISR table lookup.
4. Write kernel-level ISRs instead of task/user-level ISRs that typically take more time to execute.
5. Use the latest hardware. Intel continuously enhances Intel VT to improve virtualization performance.

Interrupt Latency Testing

Intel conducted interrupt latency testing on the Intel Industrial Solutions system consolidation series to provide data on its real-time and deterministic behavior.

Test Methodology

The interrupt latency of the virtualized industrial control platform was measured using an FPGA-based card that sent MSI interrupts to a 1x PCI Express* lane. The card calculated the time from interrupt invocation until the Intel Core i7 processor wrote to memory mapped I/O (MMIO) on the card. For each round of testing, 105,000 interrupts were sent at one millisecond intervals and the minimum, average, and maximum latency was determined.

Platform Configuration

The platform configuration is shown in Table 2.

Test Scenarios

Interrupt latency data was collected for four scenarios, which are described in the following and depicted in Figure 3.

1. **VxWorks runs native on unvirtualized platform** – VxWorks runs alone on the platform.
2. **VxWorks runs alone on virtualized platform** – The Wind River Hypervisor runs VxWorks in a VM.
3. **VxWorks runs with other VMs on virtualized platform** – Two VMs are added: a second VxWorks instance and Wind River Linux. This case was tested two ways:
 - a. Without application multiplexed I/O (AMIO)
 - b. With AMIO

Scenario 3 was run with and without AMIO, which is a serial communications channel that allows the user (test engineer) to interact with the OSs. AMIO is a development tool that enabled Intel engineers to monitor the OSs during testing, but it normally would not be implemented in a production system. AMIO generated interrupts on a regular basis, so that if the FPGA card generated an interrupt while an AMIO interrupt was executing, there was additional delay created by the hypervisor performing a context switch to service the interrupt to VxWorks.

| Platform Configuration | |
|------------------------|--|
| Hardware | |
| Processor | Intel® Core™ Processor i7-2710QE |
| Chipset | Intel® Q67 Express Chipset (Intel® BD82Q67 PCH) |
| Memory | 16 GB 1066 SODIMM |
| SATA | 128 GB Solid-State Drive |
| Ethernet | Gigabit x4: 2 x Realtek* RTL8111C, 1 x Intel® 82574IT Gigabit Ethernet Controller, 1 x Intel® 82579LM Gigabit Ethernet PHY |
| Software | |
| Target OS (non-RTOS) | Wind River Linux* 5 |
| Target RTOS | VxWorks* 6.9.2.2 kernel 2.13 BSP 2.9/2 |
| Hypervisor | Wind River Hypervisor* version 2.0.0.1 |

Table 2. Platform Configuration.

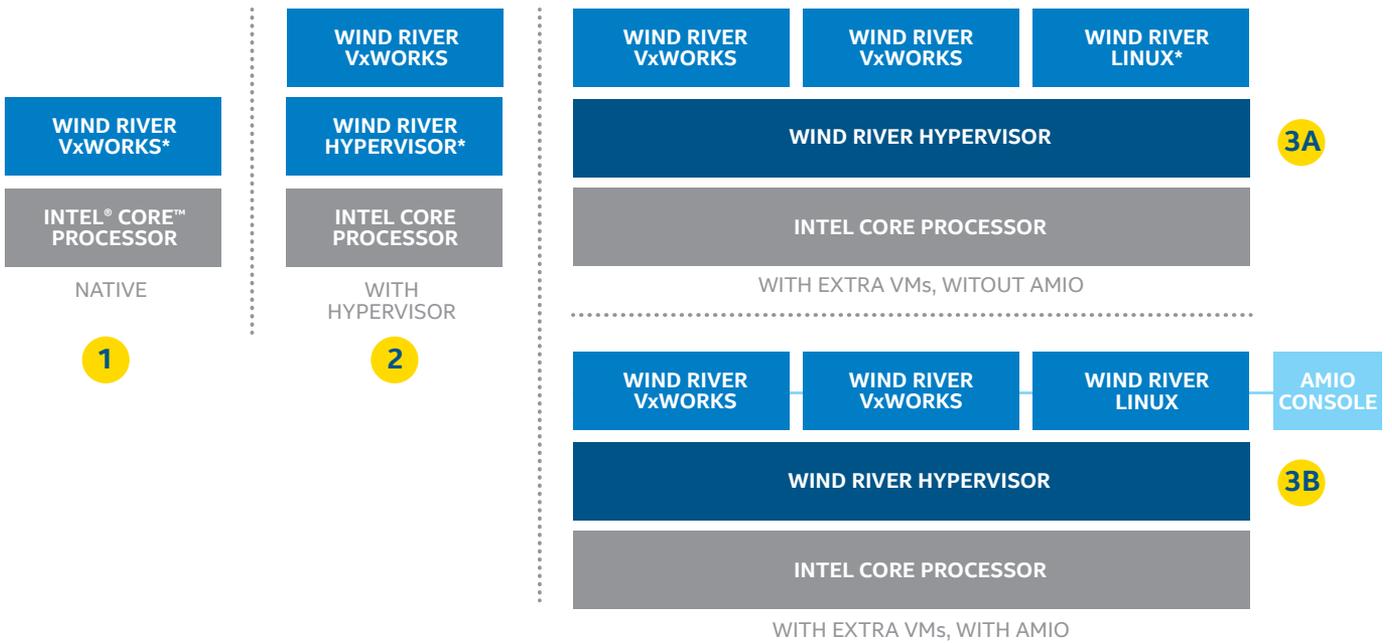


Figure 3. Test Scenarios.

Each scenario was tested three times for a total of 315,000 interrupts.

Test Results

The test results are presented in Table 3. The maximum interrupt latency for VxWorks running native on the platform was just over 3 microseconds, which was predictably the fastest scenario tested. The second scenario gives an indication of the latency impact created by the Wind River hypervisor; however, using virtualization to run

a single OS would not be practical in an actual application. After adding the hypervisor, the maximum latency was 11.78 microseconds.

Next, another VxWorks VM and a Linux VM were added to the platform. Since industrial applications vary a great deal, these OSs sat in a loop and did not run an application, thus providing best case results for the platform with three VMs running. The interrupt latency increases a small amount (to 12.38 microseconds) when the two VMs are added and AMIO

is not used. The impact of using AMIO was greatest on the maximum latency, and this likely occurred when the hypervisor had to perform a context switch from an AMIO interrupt to the FPGA card interrupt. This case may be most useful to developers because it indicates the latency when the hypervisor must first suspend a lower priority interrupt before starting up a higher priority interrupt, which presumably would be time-critical.

| | AMIO Console | VxWork* Interrupt Latency (microseconds) | | |
|---|----------------|--|---------|---------|
| | | Minimum | Average | Maximum |
| VxWorks* runs native on unvirtualized platform | Not applicable | 2.28 | 2.50 | 3.03 |
| VxWorks runs alone on virtualized platform | Not applicable | 4.67 | 5.53 | 11.78 |
| VxWorks runs with other VMs on virtualized platform | Without | 4.90 | 5.89 | 12.38 |
| | With | 4.95 | 5.84 | 17.70 |

Table 3. Test Results.³

Conclusions

The test scenarios conducted show that the Intel Industrial Solutions system consolidation series can perform with a sub 20 microsecond maximum interrupt latency response. At this level of performance, the platform can potentially execute real-time application control loops in the 100-300 microsecond range with very high precision, which is sufficient for some PLCs, and perhaps for certain motion and motor drive control applications.

This testing provides insight into best case interrupt latency performance. When applications and interrupt service routines (ISRs) are added, the interrupt latency will increase. Therefore, developers need to test their own applications on the platform to see if its real-time performance is sufficient.

Learn more about the Intel Industrial Solutions system consolidation series:
www.intel.com/industrialconsolidation

¹ Source: Intel Whitepaper, "Reducing Interrupt Latency Through the Use of Message Signaled Interrupts," January 2009. <http://www.intel.com/content/www/us/en/intelligent-systems/intel-technology/msg-signaled-interrupts-paper.html>.

² Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, and virtual machine monitor (VMM). Functionality, performance or other benefits will vary depending on hardware and software configurations. Software applications may not be compatible with all operating systems. Consult your PC manufacturer. For more information, visit <http://www.intel.com/go/virtualization>.

³ Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

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. UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

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. The information here is subject to change without notice. Do not finalize a design with this information. 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 by visiting Intel's Web site at www.intel.com.

Copyright © 2014 Intel Corporation. All rights reserved. Intel, the Intel Logo, and Intel Core are trademarks of Intel Corporation in the United States and/or other countries.

*Other names and brands may be claimed as the property of others. Printed in USA 0714/RH/HBD/PDF ♻️ Please Recycle 330740-001US

