WHITE PAPER
Intel® Atom™ Processor
INtime* RTOS from TenAsys
Embedded Computing

Delivering Deterministic, Real-Time Performance with a Low Cost Intel® Atom™ Processor-based Platform

An entry-level board uses virtualization to consolidate general purpose and real-time applications – reliably and securely – thereby reducing product cost, design complexity and time to market.

Introduction

Over the past decade, virtualization technology has inched its way down from Big Iron mainframes to highly embedded systems, like industrial controllers. Accelerating this trend, embedded developers can now consolidate real-time and general purpose applications onto a single board, a solution that costs about a hundred dollars to manufacture. This is possible with a low power, fanless platform based on the dual-core Intel® Atom™ processor D510 and INtime* from TenAsys, running real-time and Windows* operating systems at the same time. This low power platform – under 20 Watts - features the first embedded Intel® Atom™ processor with two cores, providing the computing performance needed to support virtualization for industrial and other embedded applications.

Why is this solution different? For applications with time-critical requirements, it delivers both deterministic, real-time performance and general-purpose application flexibility – at a low price. With this solution, equipment manufacturers can lower BOM cost through equipment consolidation, while reducing system size, energy consumption, design complexity and time to market. But foremost on the minds of system developers are real-time metrics, like interrupt latency and jitter. This paper presents real-time performance data and provides technical background on this virtualized Intel Atom processor-based platform running INtime for Windows.

Time-Critical Applications

On the factory floor, manufacturing equipment, like robots, motion controllers and programmable logic controllers (PLC), works together in synchronized fashion as products roll down the assembly line. For an industrial controller, trading determinacy for speed is not an option. Execution speed is sometimes important but determinacy is always important.

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. Measurements must be performed at precise, deterministic intervals. Consequently, the controller needs to achieve a level of determinism that ensures even its worst case response won't adversely affect the operation of the overall system.

"Hard" real-time systems are designed with specific response times, compared to "soft" real-time systems that deliver a best-effort response. The key differentiator is a hard real-time system ensures predictable behavior as defined by a bounded response to events. Critical machine control functions must follow a rigid schedule and meet deadlines predictably, in a definite order.

The level of determinism is a function of both software and hardware implementations. Software-wise, embedded system designers are well

aware that the vast majority of general-purpose operating systems (GPOS) are not deterministic. In a GPOS, if a real-time event occurs, like an external interrupt, there is a large variation in the time it takes for the operating system to context switch and initiate an interrupt service routine. Additional delay is incurred in hardware when the CPU switches tasks (e.g., saves registers, reloads the program counter), further increasing response times.

More GHz or More Cores?

Traditionally, chip manufacturers improved application performance by increasing CPU clock speed. For embedded applications, this approach presented some major cost and performance drawbacks: higher operating temperatures drove more expensive thermal solutions, and the use of processor power management features slowed down interrupt response. This is because reducing power consumption with power management effectively shuts down parts of the processor, changing its response times to external events, thereby increasing jitter (variation in response to an event).

Fortunately, multi-core processor technology evolved, providing much better performance per watt - lower power for the same or more processing capability. Another benefit for real-time systems is the ability to dedicate one of the processor cores to a real-time operating system (RTOS) and time-critical functions, significantly increasing determinism. This approach, which ensures dedicated computing resources for the real-time/deterministic control applications, will be covered in detail.

A Real-Time Dilemma

For many users and developers alike, Windows has become the standard for implementing human machine interfaces (HMIs) as well as other general-purpose operating system (GPOS) functions. The familiar GUI makes it easy for users to learn and use programs, while the behind-the-scenes protocols are readily interoperable with existing enterprise networks.

However, Windows alone cannot provide the level of determinism demanded by hard real-time systems, such as automated manufacturing systems that require closed-loop control. Such a time-constrained application can only be built if the timing of events – managed by the operating system – is reliable and predictable, and the developer is allowed extensive control over the relative priorities of all operations and events. Consequently, these systems need a RTOS, which is often implemented on a second board with its own computing system, like a DSP or microcontroller.

Two Boards or Dual-Core

Designing an industrial controller with two boards, real-time and general-purpose, is complex and expensive. But this can be avoided by using multi-core processors that provide an extra degree of freedom; they allow application software to optimize performance by partitioning tasks among multiple processor cores. Thus, a single board with a dual-core or many-core processor is the foundation for a real-time Windows environment that runs time-critical tasks on a dedicated execution core, resulting in a faster and more repeatable real-time response. This allows real-time applications to run without interference from non-deterministic tasks that would otherwise compete for CPU resources.

It is possible to run both real-time and Windows operating systems at the same time on an Intel Atom processor with INtime from TenAsys. With this solution, a two board PLC system comprising one board for control functions and another for human machine interface (HMI) can be consolidated onto a single board. The benefits from consolidation are far-reaching, impacting nearly every stage of a product's lifecycle. Consolidation reduces design complexity and BOM cost, and lowers power consumption and size, among other benefits, as listed in Table 1. However, these advantages are only applicable if the resulting system achieves hard real-time performance.

Benefits from Consolidation (e.g., two systems – PLC and HMI – on one board)	Results
Lower BOM cost	**Save product cost by...** ▪ eliminating redundant components, like storage devices, power supplies and chassis ▪ removing intersystem interconnect hardware, like connectors and cables ▪ manufacturing just one board
Better product	**Improve the product by...** ▪ lowering the overall power consumption because there are fewer components ▪ reducing the size of the system because there's just one board
Less design complexity	**Simplify the design by...** ▪ avoiding integration issues inherent in two platform systems ▪ using a single tools chain, which increases efficiency
Lower post release costs	**Reduce the operations complexity with...** ▪ fewer line items to order and inventory ▪ a single code base that is easier to maintain and support ▪ improved reliability by eliminating intersystem interconnect hardware
Improved performance scalability	**Cost-effectively create a family of products that...** ▪ leverage the wide breadth of 32-bit Intel® processors ▪ run the same code ▪ are based on the same design
Faster time to market	**Save time...** ▪ by reducing design complexity ▪ when adding new applications and features ▪ with a single code base, which saves validation and certification time
Better investment protection	**The code developed today...** ▪ will run on an Intel processor of tomorrow

Table 1: Benefits from Consolidation

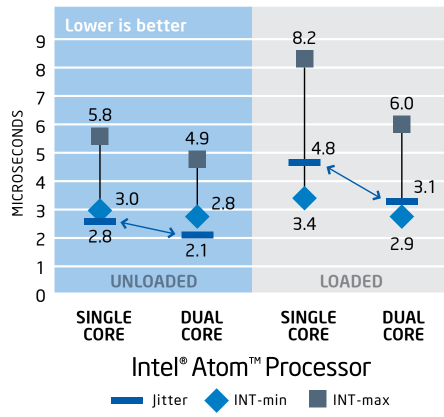


Figure 1. I/O Interrupt Resonse and Jitter

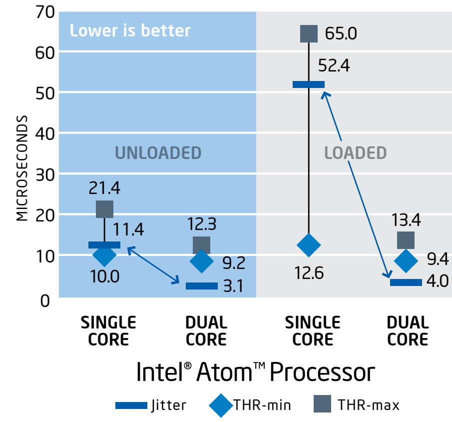


Figure 2. Thread Response Time and Jitter

Real-Time Performance Measurements

Industrial controllers operate with very high precision, executing control loops in the 100 microsecond (uS) to 20 millisecond (mS) range. When jitter becomes a significant percentage of the cycle time, it adversely affects the stability and quality of the control algorithm. Typically, jitter that's less than 10 percent of the loop time, less than 10 microseconds, is acceptable. Given the significance of jitter, the following compares the jitter measured on single core and dual-core Intel Atom processors when running real-time and general-purpose operating systems using INtime from TenAsys.

System Configuration

Both systems were loaded with Windows 7 and INtime for Windows, an RTOS that runs alongside Windows and schedules the control application. During the test, Windows and the INtime RTOS were loaded with applications that exercised their respective environments. The goal of this configuration was to measure how each platform responded to events in real-time while running general-purpose applications. For comparison, measurements were taken under two conditions:

Unloaded: Only the control application ran on INtime for Windows

Loaded: The INtime environment ran the Whetstone benchmark, and the Windows environment ran a Microsoft* graphics application[1].

Both Intel Atom processors incorporate Intel® Hyper-Threading Technology (Intel® HT Technology)[1], which enables each physical processor core to run two software threads simultaneously. Consequently, the single core Intel® Atom™ processor D410 ran two threads: one thread dedicated to Windows and the other to INtime. The dual-core Intel Atom processor D510 ran four threads: three threads dedicated to Windows and one thread to INtime (INtime is a single threaded OS).

The platform under test includes the Intel® NM10 Express Chipset, which has performance comparable to the embedded platform that uses the Intel® 82801HM I/O Controller Hub.

Performance Results:

The jitter performance of each platform (Table 2) was based on I/O interrupt and thread response time measurements: The I/O interrupt response time starts with an external event and ends when the first instruction of an interrupt service routine (ISR) is executed. The thread response time is the I/O interrupt response time plus the time for the ISR to start a thread; this includes the time needed

[1]Graphic application: The_Magic_of_Flight-1080 downloaded from www.microsoft.com/windows/windowsmedia/musicandvideo/hdvideo/contentshowcase.aspx.

for the operating system scheduler to setup the thread, allocate memory, etc. For more details on the method for measuring response time, see the side bar called Jitter Measurement Method.

In all, there were four test cases, each running for three minutes. An interrupt event commenced every millisecond, so 180,000 measurements were recorded for each case. In every case, the dual-core Intel Atom processor demonstrated better jitter performance[2] than the single core version, as seen in Figures 1 and 2.

Tested Configuration	
Processors	a) Intel® Atom™ Processor D510 b) Intel® Atom™ Processor D410
Chipset	Intel® NM10 Express Chipset
Memory	2 GB
Operating System	Microsoft* Windows* 7

Table 2. Test Platform Configuration

I/O interrupt jitter (Figure 1)
- Unloaded, the dual-core jitter was 0.7 uS better (25%)
- Loaded, the dual-core jitter was 1.7 uS better (35%)

I/O thread jitter (Figure 2)
- Unloaded, the dual-core jitter was 8.3 uS better (73%)
- Loaded, the dual-core jitter was 48.4 uS better (92%)

The worst case jitter corresponded to the I/O threaded measurement when both operating systems were loaded, and the jitter for the dual-core Intel Atom processor was 4.0 uS. This result is rather exceptional for a low-cost platform running both real-time and general-purpose applications.

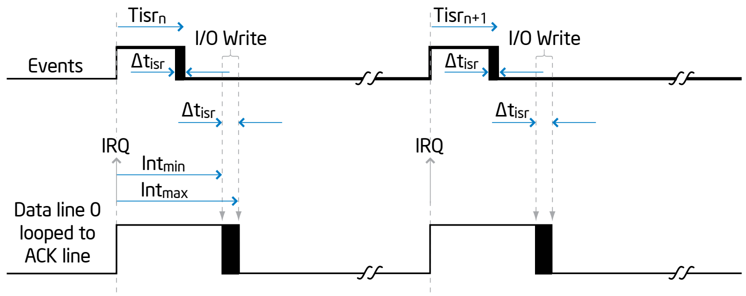


Figure 3. Interrupt Response Time Measurement (int)

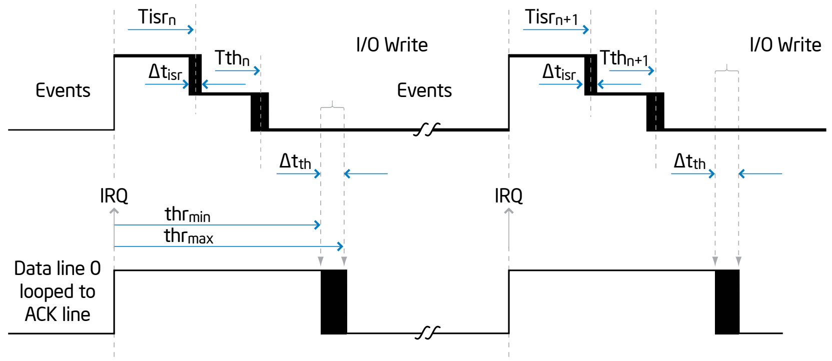


Figure 4. Thread Response Time Measurement (thr)

Jitter Measurement Method

The interrupt response time, measured with an oscilloscope, is the width of a pulse generated by an external pin, DATA 0 from the parallel interface, labeled intmax and intmin in Figure 3. The jitter is the difference between Int_{max} and Int_{min}, labeled "Δt_{isr}" in the figure.

The DATA 0 pin is connected to the ACK pin of the interface, and it is used to generate interrupts. First, the test program writes to the pin, which outputs a rising edge that in turn initiates an interrupt service routine (ISR). Second, a set of instructions in the ISR de-asserts DATA 0 and the pin outputs a falling edge; this completes the pulse used to measure the interrupt response time loop, called int.

The thread response time starts with a rising edge from the DATA 0 pin, exactly like the interrupt response just discussed. In this case, upon entry into the ISR, a thread call is made to start a thread. The time it takes for the scheduler in the operating system to setup the thread is labeled Tth_n in Figure 4, and this time is affected by the processor workload. Once the thread starts, it executes a set of instructions that writes to DATA 0, which generates a falling edge and completes the pulse. The thread response time is indicated by thr_{max} and thr_{min}, and their difference is the jitter, Δt_{th}.

INtime* Real-time OS for Windows*

INtime is an RTOS that provides real time services to the Windows platform. By dedicating one or more processor cores to real-time tasks, INtime 4.0 enables embedded systems to separate deterministic processes from Windows task management, maximizing system performance and enabling 100 percent reliability.

Real-time Windows* applications, side-by-side

A traditional industrial controller may employ one or more RTOSs for machine control, and a loosely coupled Windows OS for HMI applications, as depicted on the upper half of Figure 5. This approach may require two or more boards. Alternatively, a single board with a multi-core Intel® processor can support all of the system software.

For instance, INtime software can be installed on a system that already runs standard Windows (lower half of Figure 5). The INtime RTOS runs as a separate independent kernel, so the Windows system is not modified or compromised by process failures and faults on the INtime kernel, and vice versa. Real-time processes run on the INtime kernel, and non-real-time processes run on Windows: the developer decides whether a given process has a requirement for real-time responsiveness and what level of determinism is desired.

Save development, manufacturing and maintenance costs

By combining real-time and Windows applications on a single platform, redundant hardware is eliminated, saving manufacturing and component costs. Software development and maintenance are made easier because the same Visual Studio tools can be used for development and debugging of INtime and Windows applications.

Replace expensive special-purpose processors with a dedicated processor core

Rather than allocating time-critical tasks to an expensive DSP board, developers can run multiple instances of INtime 4.0 on dedicated cores and provide the same functions as multiple CPU boards. INtime makes it possible to implement DSP-like functionality quickly and achieve DSP-level performance by supporting the following Intel software technologies:

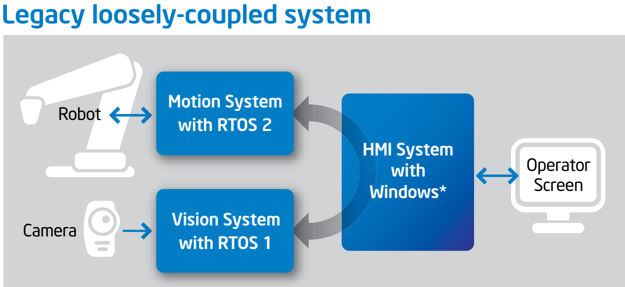
- Intel® Streaming SIMD Extensions (Intel® SSE)
- Intel® Integrated Performance Primitives (Intel® IPP) Library

Ultra-Low Power, Fanless Platform

Fitting on a board the size of a credit card (on a Mini-ITX carrier), a platform based on the dual-core Intel Atom Processor D510 is very high performance, yet low power. Besides delivering the impressive jitter performance mentioned previously, the platform supports several high-bandwidth interfaces such as PCI Express*, PCI, Serial ATA and Hi-Speed USB 2.0. Paired with the Intel® 82801HM I/O Controller, as shown in Figure 6, the Intel Atom processor is software compatible with previous 32-bit Intel® architecture products.

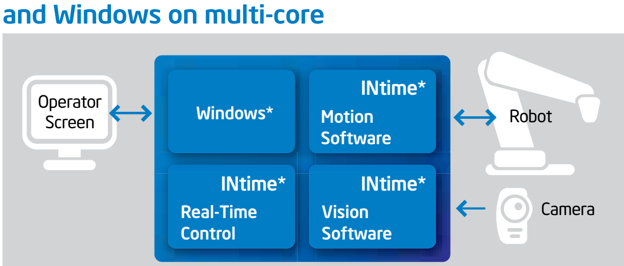
The processor integrates graphics and memory controllers that are built directly into the processor die, decreasing overall power consumption and footprint.
The integrated Intel® Graphics Media Accelerator 3150 supports high quality 3D graphics, video, and LVDS and VGA ports, eliminating the need for an add-in graphics card. The Intel Atom processor D510 and companion chipset feature 7-year extended lifecycle support.

Legacy loosely-coupled system



Traditional RTOS applications consisted of a PC running Windows*, loosely coupled to other computers with real-time OSes.

INtime consolidates real-time and Windows on multi-core



With INtime 4.0, real-time OS(es) and Windows* run side-by-side on the same multi-core processor.

Figure 5. INtime* Application Example

Consolidate With Dual-Core

Advances in multi-core technology and virtualization software are enabling industrial equipment manufacturers to significantly reduce design complexity and cost through consolidation. Further increasing the cost benefit, Intel Atom processors combined with an INtime environment achieve a very low price point, opening the door to more industrial applications. Suitable for hard real-time systems, the solution has an impressive level of jitter, about 4 microseconds, while running an RTOS and Windows simultaneously. This compact computing platform delivers dual-core performance, deterministic real-time response and a flexible set of standard interfaces. Now is the time to consolidate industrial control systems using dual-core Intel Atom processors.

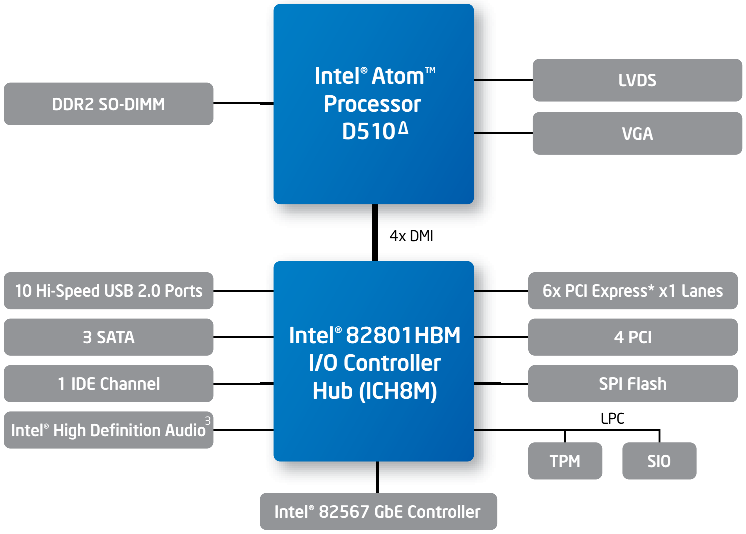


Figure 6. Platform Based on Intel® Atom™ Processor D510

For more information about INtime from TenAsys, visit **http://www.tenasys.com/products/intime.php**

For more information about Intel industrial embedded computing solutions, visit **http://www.intel.com/go/industrial**

Δ Intel® processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See www.intel.com/products/processor_number for details.

1 Intel® Hyper-Threading Technology (Intel® HT Technology) requires a computer system with a processor supporting Intel HT Technology and an Intel HT Technology enabled chipset, BIOS and operating system. Performance will vary depending on the specific hardware and software you use. See www.intel.com/info/hyperthreading/ for more information including details on which processors support Intel HT Technology.

2 Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit www.intel.com/performance/resources/limits.htm or call (U.S.) 800-628-8686 or 916-356-3104.

3 Intel® High Definition Audio (Intel® HD Audio) requires a system with an appropriate Intel® chipset and a motherboard with an appropriate codec and the necessary drivers installed. System sound quality will vary depending on actual implementation, controller, codec, drivers and speakers. For more information about Intel HD audio, refer to http://www.intel.com/.

Printed in USA 1010/RH/SD/PDF ♻ Please Recycle 324510-001US