

## Webinar: Intel<sup>®</sup> System Studio 2015

Accelerate Development for Embedded, Mobile, and the Internet of Things

Robert Mueller-Albrecht Technical Consulting Engineer Intel SSG Developer Products Division

# Intel's Vision If it is smart and connected, it is best with Intel.



## Explosion Of Connected Smart Devices

... is driving an unprecedented need for efficient tools to meet shorter development cycles



50B Devices

2020 1

Copyright© 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners

## Intel<sup>®</sup> System Studio 2015 Overview

Deep system-wide insight into power, performance, and reliability that helps accelerate time to market of Intel Architecture-based mobile and embedded systems and embedded applications



Accelerate	Strengthen	Boost Power
Time To	System	Efficiency and
Market	Reliability	Performance
Speed-up development and	Enhance system stability using	Boost system power efficiency
testing with deep	in-depth system-wide	and performance using
hardware and	debuggers	system-wide analyzers,
software insight	and analyzers	compilers and libraries
Embedded or Mobile System	inside" inside" inside" inside"	intel nside DRE 17 KEON

<sup>1</sup> Linux\*, Embedded Linux, Wind River\* Linux\*, Yocto Project\*, Tizen\*

Learn more at: http://intel.ly/system-studio



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.



## Intel<sup>®</sup> System Studio 2015 Overview

Deep system-wide insight into power, performance, and reliability that helps accelerate time to market of Intel Architecture-based mobile and embedded systems and embedded applications



DEBUGGERS	& TRACING	ANALYZE	RS	COMPILER & LIBRARIES					
System	Application	Power & Performance CPU / Graphics	Memory & Threading	C/C++ Image, Signal, Media, Data & Math Compiler Processing					
JTAG Interface	JTAG Interface System & Application code running on Linux* <sup>1</sup> , Android* or Windows*								
Embedded or N	1obile System	(intel) Quark <sup>-</sup>	inside CORE'13 CORE'15	(intel) inside CORE 17 XEON					

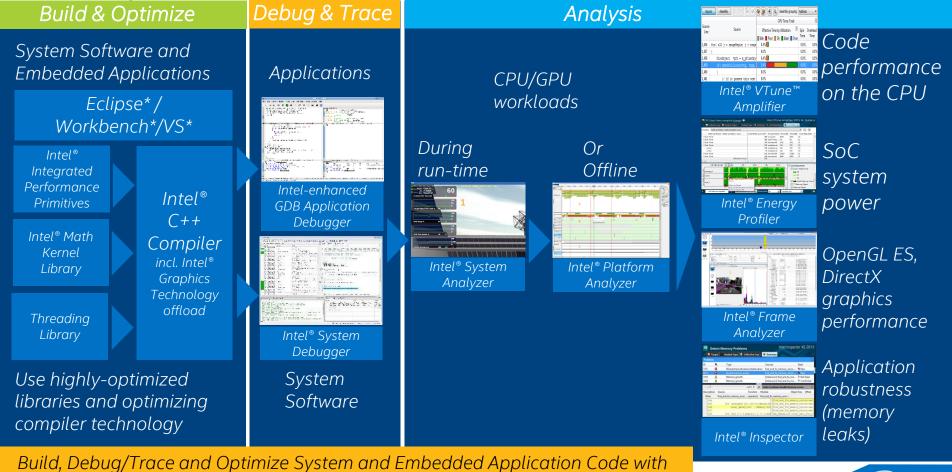
<sup>1</sup> Linux\*, Embedded Linux, Wind River\* Linux\*, Yocto Project\*, Tizen\*

Now also Android\* Lollipop\* and 64-bit ready



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

## Intel<sup>®</sup> System Studio Workflow



One Integrated Solution – Intel® System Studio

eir respective owners.

6

(intel

Intel <sup>®</sup> System Studio 2015 Components			Target OS Support									
inter System Studio 2015 Components			Linux* <sup>1, 5</sup>			Android* <sup>5</sup>			ows*	VxWorks*		
Category	Component	Composer Edition	Professional Edition	Ultimate Edition	Composer Edition	Professional Edition	Ultimate Edition	Composer Edition	Professional Edition	Composer Edition		
Host Operating S	ystems	Lini	ux*, Windo	WS*	Lini	ux*, Windo	WS*	Wind	ows*	Linux*, Windows*		
Integrated Develo	opment Environment		se*, Wind F Vorkbench		* Eclipse* Visual Studio*		Wind River* Workbench*					
	Intel <sup>®</sup> C++ Compiler	√	√	$\checkmark$	√	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	<b>√</b> <sup>2</sup>		
Compiler &	Intel <sup>®</sup> Integrated Performance Primitives		$\checkmark$	√	√	$\checkmark$	√	$\checkmark$	√	<b>√</b> <sup>2</sup>		
Libraries	Intel <sup>®</sup> Math Kernel Library		√	√				$\checkmark$	√			
	Intel® Threading Building Blocks	√	√	$\checkmark$	√	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$			
Application Debugger	Intel-enhanced GDB* Application Debugger	√	√	$\checkmark$	√	$\checkmark$	√					
	Intel <sup>®</sup> VTune™ Amplifier for Systems		√	$\checkmark$		$\checkmark$	√		$\checkmark$			
	Intel® Energy Profiler		√6	√6		$\checkmark$	√		√			
A	System Analyzer					$\checkmark$	√		√			
Analyzers	Platform Analyzer <sup>4</sup>					$\checkmark$	$\checkmark$		$\checkmark$			
	Frame Analyzer <sup>4</sup>					$\checkmark$	$\checkmark$		$\checkmark$			
	Intel® Inspector for Systems		√	$\checkmark$					$\checkmark$			
System Debugger	Intel <sup>®</sup> System Debugger (JTAG) <sup>3</sup>			~			√					
1	1 Linux*	Embeddeo	LLinux Winc	Pivor* Linu	v* Vocto Dr	oioct* Tizon	*					

<sup>1</sup> Linux\*, Embedded Linux, Wind River\* Linux\*, Yocto Project\*, Tizen\*

<sup>2</sup> Delivered with Wind River\* VxWorks\* platform\*

<sup>3</sup> Via Intel<sup>®</sup> ITP-XDP3 probe, OpenOCD<sup>\*</sup>, Macraigor\* usb2demon\* and EDKII\* for UEFI\*

<sup>4</sup> Available on Windows\* host only

<sup>5</sup> Linux\* and Android\* target support available in a single product
<sup>6</sup> For detailed processor support please visit: <u>https://software.intel.com/en-us/intel-energy-profiler</u>

intel



Copyright© 2015, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.



"With the new Intel® System Studio 2015, we improved the user experience of our recently launched Android\* based tablet tolino tab 8" (optimized for eReading) drastically by a factor of 3x (200ms vs. 500-700ms); which reduced the CPU workload and the resulting power consumption by at least the same factor." Dirk Hofmann, Chief Product Owner, Deutsche Telekom

"It is well apparent that if a new customer will develop Intel Architecture based products, then Intel<sup>®</sup> System Studio tools are essential for success..." Wayne Merrill, Manager, International Dept., Flatoak Co., Ltd./JAPAN "I am very satisfied with Intel System Studio 2015 product. The C++ compiler is very fast and complete. The development tools that are part of these suites are very useful and they help detecting performance issues quite easily." Eduardo Quintana, SFTWY CDI Ltda., Microsoft Partner

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

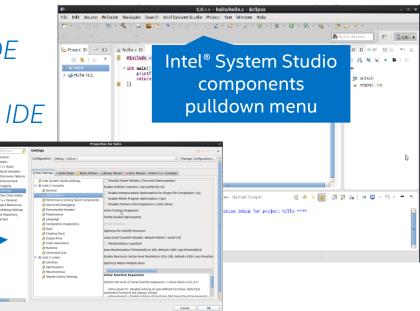
## IDE integration

9

## **Comprehensive Eclipse\* IDE Integration**

- Full integration of Intel<sup>®</sup> C++ Compiler into cross-build environments under Eclipse\* IDE
- Integrates Intel<sup>®</sup> System Debugger & Intel<sup>®</sup>
   VTune<sup>™</sup> Amplifier for Systems into Eclipse\* IDE
- Integration of Intel enhanced GDB\*

Easy configuration of Intel<sup>®</sup> System Studio components (build system, analyzers, debugger) within the IDE



#### Develop Fast Code Faster, Efficiently

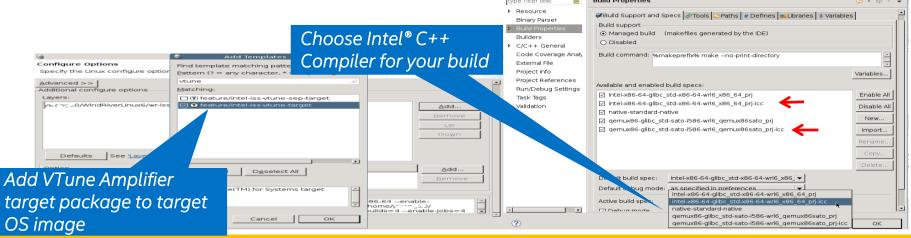


ntel | 10



# Tight Wind River\* Workbench Integration

- Seamlessly switch between Intel<sup>®</sup> C++ Compiler and GCC\*
- Full sysroot and secondary toolchain layer support
- Wind River\* build environment recipes
- Launch Intel<sup>®</sup> System Debugger & Intel<sup>®</sup> VTune<sup>™</sup> Amplifier for Systems from Workbench<sup>\*</sup>
- Deploy runtime libraries, VTune<sup>™</sup> Amplifier sampling collector automatically as part of target image generation



## Intel® Architecture Performance and Debug Solution for Wind River\* Linux\*



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

11

## Windows\* Target Support

Plug-in to Microsoft<sup>®</sup> Visual Studio\* the standard IDE for Windows\*-based development



- Intel<sup>®</sup> C++ Compiler and Libraries for improved performance
- Intel<sup>®</sup> Integrated Performance Primitives, Intel<sup>®</sup> Math Kernel Library, Intel<sup>®</sup> Threading Building Blocks performance libraries for improved development efficiency
- Intel® Inspector for Systems for advanced code correctness checking
- Intel<sup>®</sup> VTune<sup>™</sup> Amplifier for Systems for advanced performance profiling
- Intel<sup>®</sup> Energy Profiler to increase energy efficiency

#### Develop Fast, Energy-efficient Embedded Windows\* Applications



ompose

Professional Edition

#### Intel<sup>®</sup> System Studio 2015 for Windows\* NEW

File Edit View Project Build Debug Team Data Tools Test Window Help 🛅 • 🖼 • 💕 🚽 🔰 👗 🐴 🖄 🔊 • 🔍 • 🚚 • 🖳 🕨 Debug - 12 - 🗠 🚰 📷 🕺 📯 🛃 🖳 - 📮 Win32 [ 및 월 월 44 월] 史 영 [ 그 위 및 위 및 용 및 🗒 1 🕼 🗊 🧿 1 🚽 → 및 × New Amplifier Result × ng-tbb.cpp Intel VTune Amplifier 2015 for Systems 🚝 Choose Analysis Type File Edit View Project Build Debug Team Data Tools Test Window Help - 🔩 🕾 🎲 🕺 🏷 🛃 🖳 -🛅 = 🖼 = 💕 🛃 🦪 👗 🗈 🖳 🔊 = 🔍 - 💷 - 🖳 🕨 Debug Win32 - 🖄 Concurrency Analyze how your application is usin 🖪 😣 🗛 🌾 👔 🚖 💷 😫 🚍 😑 😑 🗛 🗛 👘 👔 A Advanced Hotspots ? > ng-tbb-intel Property Pages and identify potential candidates for for more details Solution Explorer 🔓 🚯 🖧 Configuration: Active(Debug) Platform: Active(Win32) V Configuration Manager... A Highly accurate CPU time of Microarchitecture Analys privileges. Solution 'NQueens' (7 projects) A General Exploration No Common Properties Unmask Floating Point Exceptions Re-openmp-intel CPU sampling interval, ms: Configuration Properties Floating Point Expression Evaluation Default 📓 ng-openmp-taskg - CPU Specific Analysis General Analyze user tasks Disable Function Splitting No 🞇 nq-serial intel Core 2 Proce Debugging Add Processor-Optimized Code Path None 💐 ng-stl Nehalem / Westr Analyze Intel runtimes and user Intel Debugging None 🐹 ng-tbb-intel 🛓 🗁 Sandy Bridge Ana Intel Performa Analyze GPU usage **External** Dependencies Haswell Analysis VC++ Directo Intel Netburst(R) Microarchitecture and Pentium(R) M processor family with Streaming SIMD Extensions 2 (SSE2) (/QxSSE2) Header Files A TSX Exploratic Analyze Processor Graphics hardware ⊿ C/C++ Resource Files Intel(R) Core(TM) and Intel Netburst(R) Microarchitectures with Streaming SIMD Extensions 3 (SSE3) (/QxSSE3) General Trace OpenCL kernels on Process Source Files Intel(R) Atom(TM) processor family with MOVBE instruction support (/OxSSSE3\_ATOM) & CDII/GDII Concurren General [Ir Intel(R) Core(TM)2 processor family with Supplemental Streaming SIMD Extensions 3 (SSSE3) (/QxSSSE3) end na-tbb.cpp Debug [Int Intel(R) Core(TM)2 processor family with SSE4 Vectorizing Compiler and Media Accelerators (/QxSSE4.1) 🐹 ng-tbb-lambda Optimizati Intel(R) Core(TM) processor family with SSE4 Efficient Accelerated String and Text Processing (/QxSSE4.2) Ima-win32api-intel Optimizati Intel(R) Core(TM) processor family with Intel(R) Advanced Vector Extensions (Intel(R) AVX) support (/QxAVX) 015 for Systems messages Preprocess Intel(R) AVX, including instructions in 3rd Generation Intel(R) Core(TM) processors (/QxCORE-AVX-I)

Code Gene Intel(R) Advanced Vector Extensions 2 (Intel(R) AV Code Gene Intel(R) Advanced Vector Extensions 512 (Intel(R) /

Intel Processor-Specific Optimiza

Generates optimized code for spec run on the targeted processor or a

Launch performance sampling and use Intel<sup>®</sup> VTune<sup>™</sup> Amplifier embedded into Microsoft\* Visual Studio

Switch easily between using Microsoft\* & Intel compiler. Convenient access to advanced optimizations and diagnostics

Fully Microsoft Visual Studio\* integrated solution to optimize Windows\* applications

Language Language miler

Precompiled Heade

Precompiled Heade Output Files

Browse Information Diagnostics [Intel C Advanced

Command Line



🖧 Analysis Type

& Basic Hotspots

A Concurrency

A Bandwidth

Platform Analysis

Show output from: Intel

A Locks and Waits

AAAA

Solution Explorer

🕒 👔 🖧

Solution 'NQueens' (7 projects)

External Dependencies

📓 ng-openmp-intel

🔯 ng-serial

▲ X ng-tbb-intel

🦹 ng-stl

🐹 ng-openmp-taskg

Header Files

🚞 Resource Files

nq-tbb.cpp

Source Files

🐹 ng-tbb-lambda

🗿 nq-win32api-intel

💐 S... 💐 Cl... 📠 Pr... 🏹

Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

## Compiler and libraries



## **Compiler & Libraries Overview**

C/C++Compiler

Image, Signal, Media, Data & Math Processing

	Intel® C++ Compiler	Intel® Integrated Performance Primitives	Intel® Math Kernel Library	Intel® Threading Building Blocks
What does it provide	Build highly optimized executables and libraries for Intel® Architecture	Performance optimized software building blocks	Advanced math and statistics functions	Scalable building blocks for advanced data and task parallelism
Key Purpose	Optimize performance of critical data and compute intensive workloads	Build highly optimized and easy to maintain signal and media processing codecs	Implement highly optimized solutions for data manipulation and data processing	Design and implement advanced task and data parallelism for highly parallel synchronized workloads
How	Rebuild critical workloads with architectural and parallelism optimizations	Integrates rich set of function primitives easily callable from workload	Advanced optimized libraries provide building blocks for complex math and data computation	Highly flexible C++ templates and synchronization primitives

Build and Performance Solutions for Embedded Cross-Development in a World of Connected Devices



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

15

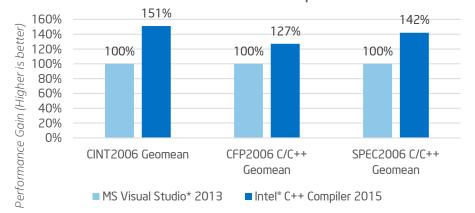
inte

## Intel<sup>®</sup> C++ Compiler Benchmarks on Windows\* Targets Estimated Performance Difference

#### **COMPILER & LIBRARIES**

C/C++ Image, Signal, Media, Data & Math Compiler Processing

#### CINT2006, CFP2006 C/C++, SPEC2006 C/C++ RATE Benchmarks - Best Option Set



#### Compilers

Intel® C++ Compiler for IA-32 applications, Version 15.0.007 Build 20140905 Intel® C++ Intel(R) 64 Compiler for Intel(R) 64 applications, Version 15.0.007 Build 20140905 Microsoft\* C/C++ Optimizing Compiler Version 18.00.21005.1 for x86 Microsoft\* C/C++ Optimizing Compiler Version 18.00.21005.1 for x64

Hardware nsticlew612.ins.intel.com Name Haswell i7-4770K Platform Microsoft\* Windows\* 7 SP1 Hardware Intel® Core™ i7-4770K CPU @ 3.50GHz RAM 16GB HDD 1TB

Benchmarks CINT2006 Geomean CFP2006 C/C++ SPEC2006 C/C++ Geomean

NOTE: 32-bit compilers for CINT2006 in RATE mode were used, as in SPEC publications

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance test, such as CINT2006\*, CFP2006 C/C++\*, SPEC2006 C/C++\*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. Benchmark Source: Intel Corporation. For more complete information about compiler optimizations, see our <u>Optimization Notice</u>.

## Intel<sup>®</sup> C++ Compiler for Windows\* Based Embedded Devices Provides A Performance Gain Up To **50%** Compared To Industry Leading C++ Compiler



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

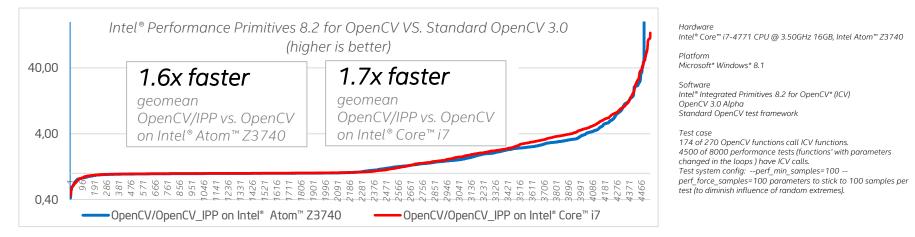
16

## Intel<sup>®</sup> Integrated Performance Primitives for OpenCV\*

COMPILER & LIBRARIES

C/C++ Image, Signal, Media, Data & Math Compiler Processing

**OpenCV** (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. An open source version is available which calls IPP functions to gain more performance. Download: <u>http://opencv.org/opencv-3-0-alpha.html</u>



Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance test, such as OpenCV test framework are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. Benchmark Source: Intel Corporation. For more complete information about compiler optimizations, see our <u>Optimization Notice</u>.

#### OpenCV\* 3.0 Performance Increases with Intel® Performance Primitives Optimizations Up To Factor **1.6x** (geomean)

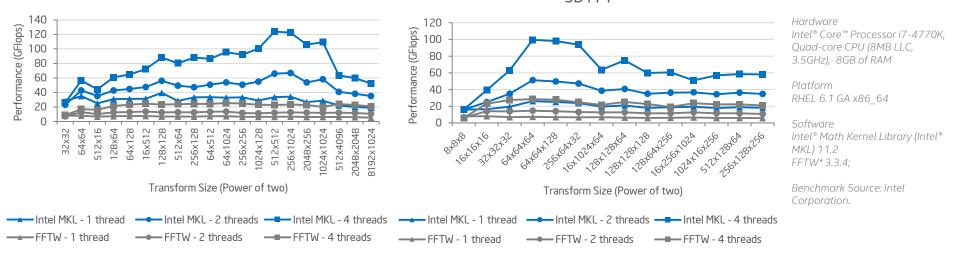


Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

17

Intel<sup>®</sup> System Studio 2015 - Performance and Power Efficiency

## FFT Performance Using Intel® MKL vs. FFTW\* Single Precision Complex 2D and 3D FFT on Intel® Core™ Processor i7-4770K



Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance test, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. Benchmark Source: Intel Corporation. For more complete information about compiler optimizations, see our <u>Optimization Notice</u>.

#### Intel® Math Kernel Library Helps to Boost the Performance of 2D FFT Image Processing Algorithms Significantly Over Open Source FFTW\* Library



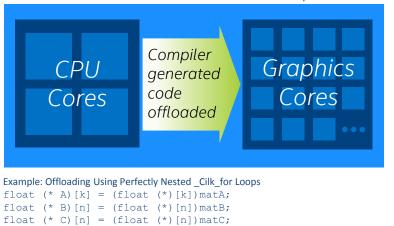
Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

Intel<sup>®</sup> System Studio 2015 - Performance And Power Efficiency

## Offload Compute-intensive Code To Integrated Graphics Cores

- Compiler-generated code executed across CPU and graphics cores with simple #pragma
- Employ Intel® Cilk™ parallel extensions for highly parallel execution across graphics cores

Intel<sup>®</sup> Core<sup>™</sup> Processors and Intel Xeon Processors with Intel HD or Intel Iris Pro Graphics



For Excellent Image And Signal Processing Performance In Embedded Applications



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

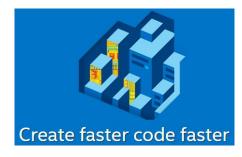
## Accelerate Build and Design Intel® System Studio

#### Intel<sup>®</sup> C++ Compiler

- Highly optimized build with Intel<sup>®</sup> C++ Compiler
- Seamless integration into your cross-build environment
- Intel<sup>®</sup> Threading Building Blocks
- Advanced parallel programming models for tuning concurrent programming flows
- Intel<sup>®</sup> Integrated Performance Primitives
- Function primitives building blocks for media and signal processing Intel<sup>®</sup> Math Kernel Library
- Highly optimized library for math and data analytics and manipulation

#### **COMPILER & LIBRARIES**

C/C++ Image, Signal, Media, Data & Math Compiler Processing



## Accelerate Performance and Time-to-Market



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

20

## Debuggers

Optimization Notice Copyright<sup>®</sup> 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective own



Debugger & <sup>-</sup>	Trace C	<b>)</b> verview

	System	Appli	cation	
	Intel <sup>®</sup> System Debugger	SVEN Technology SDK	Intel-enhance Debugger	ed GDB* Application
What does it provide	Interactive system software runtime issue debug with deep architectural, UEFI and OS insight	Static instrumentation based data event tracing across Linux* kernel and application space		ctive application cross-debug pecific branch trace and data pabilities
Key Purpose	Identify and fix deterministic system software defects	Identify and fix timing sensitive and non-deterministic data processing issues (especially across user space/kernel space boundary)	Identify and fix software defec	x deterministic application cts
How	Symbolic source level JTAG and debug agent run-control based stepping and execution flow analysis	Log and analyze data event traces in trace viewer and correlate to execution flow and time stamps	level stepping.	bug agent for symbolic source . Use process specific branch augment callstack.

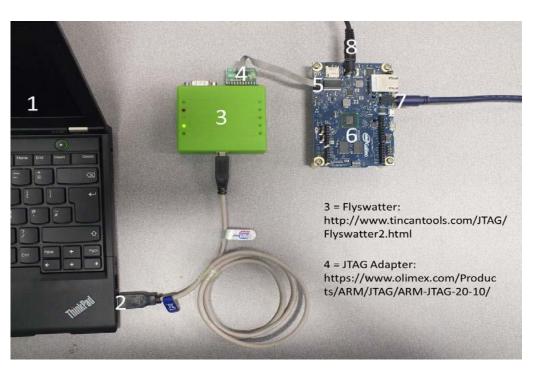
Deep insight into program flow and architecture for the entire software stack helps to resolve issues faster



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

22

## Intel<sup>®</sup> System Debugger and Intel<sup>®</sup> Quark SoC (2)



Recommended setup for debugging with OpenOCD\*

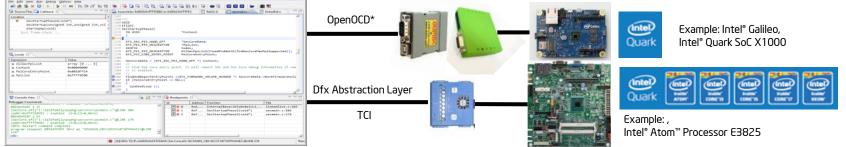
- Host System
- U
- JTASB 2.0 male-male A-B cableG Probe
- ARM-JTAG-20-10 Adapter
- JTAG Port
- Intel<sup>®</sup> Galileo Board
- Serial Cable to view boot process
- Power Supply



# Intel<sup>®</sup> System Debugger

#### **Key Features**

- Linux\* and Windows\* host
- JTAG debug for Intel<sup>®</sup> Atom<sup>™</sup>, Core<sup>™</sup>, Xeon<sup>®</sup> & Quark<sup>™</sup> SoC-based platforms
- EFI/UEFI Firmware, bootloader debug, Linux\* OS awareness
- In depth visualization of memory configuration, system state and register sets
- Dynamically loaded Linux kernel module debug
- LBR & Intel<sup>®</sup> Processor Trace On-Chip instruction trace support



Intel® ITP-XDP 3BR for all Intel platforms

# Complete system debug solution provides deep Insights into memory and system configuration



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

10

**DEBUGGERS & TRACING** 

Application

System

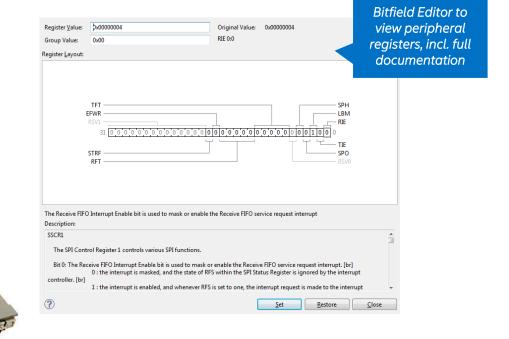
Intel<sup>®</sup> System Studio 2015 - Enhanced Developer Productivity

## System Debugging For Intel<sup>®</sup> Quark<sup>™</sup> Platforms

- Supports connection via low-cost OpenOCD\*-based JTAG devices
- Insight to Intel<sup>®</sup> Quark<sup>™</sup> SoCs



JTAG devices available for <\$100



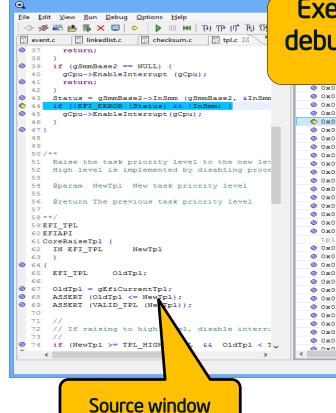
#### Full Featured, Low-cost System Debug For Intel® Quark™ Platforms



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.



## Execution Trace GUI Overview



Optimization Notice

# Execution Trace GUI is fully integrated with source debug, user can view trace data simultaneously with

## source, ASM , callstack...

	Ox0038:0x0000000A197 lea r8, ptr [rip+0xC96F]	SOURCE	for (Link = gHandleList.ForwardLink,
Smm (gSmmBase2, &InSmn	Ø 0x0038:0x0000000A197 lea rcx, ptr [rip+0xC940]	SOURCE	Handle = CR (Link, IHANDLE, AllHand
) && !InSmm) {	Ox0038:0x0000000A197 mov edx, 0x6A	SOURCE	for (Link = gHandleList.ForwardLink,
t (gCpu);	Ox0038:0x0000000A197 call 0xA197BEAC <copyguid(class guii<="" p=""></copyguid(class>	SOURCE	CoreReleaseProtocolLock ();
	© 0x0038:0x0000000A197 mov rcx, rbp	SOURCE	for (Index = 0; Index < Count; Index-
	Ox0038:0x0000000A197 mov eax, 0xFFFFFFF	SOURCE	CoreConnectController (HandleBuffe:
	• 0x0038:0x0000000A197 shl eax, cl     • 0x0038:0x0000000A197 shl eax, cl     • 0x0038:0x00000000A197 shl eax, cl     • 0x0038:0x0000000000000000000000000000000	SOURCE	
	Ox0038:0x0000000A197 movsxd r13, eax	SOURCE	CoreConnectController (HandleBuffe)
	Ox0038:0x0000000A197 jmp 0xA1977457 <corerestoretpl(unsic< p=""></corerestoretpl(unsic<>	SOURCE	CoreFreePool(HandleBuffer);
y level to the new lev	Ox0038:0x0000000A197 mov eax, ecx	SOURCE	if (FvDevice->Signature == FV2 D
ted by disabling proce	Ox0038:0x0000000A197 cmp rcx, rax	SOURCE	FvDevice->Fvb = Fvb:
	0x0038:0x00000000A197 jnz 0xA197731F <coreraisetpl(unsigne< p=""></coreraisetpl(unsigne<>	SOURCE	} else {
sk priority level	Ox0038:0x0000000A197 test ecx, ecx	SOURCE	if ((Type & EVT NOTIFY SIGNAL) != 0x(
	Ox0038:0x00000000A197 jnz 0xA197730D <coreraisetpl (unsigne)<="" p=""></coreraisetpl>	SOURCE	InsertHeadList (&gEventSignalQueue,
ask priority level	Ox0038:0x0000000A197 or rdi, 0xFFFFFFFFFFFFFFFFFFF	SOURCE	CoreReleaseEventLock ();
	Ox0038:0x0000000A197 jmp 0xA1977341 <coreraisetpl (unsigne)<="" p=""></coreraisetpl>	SOURCE	} else {
	Ox0038:0x0000000A197 mov edi, 0x1F	SOURCE	FvDevice = AllocateCopyPool (size
	Ox0038:0x00000000A197 jmp 0xA1977319 <coreraisetpl (unsigne)<="" p=""></coreraisetpl>	SOURCE	FvDevice->IsFfs3Fv = Compt
	tpl.c:68 ASSERT (OldTpl <= NewTpl);	SOURCE	FvDevice->Fv.ParentHandle = Fvb-:
	0x0038:0x0000000A197 dec rdi	SOURCE	if (Fvb->ParentHandle != NULL) {
	0x0038:0x0000000A197 add ecx, ecx	SOURCE	CoreFreePool (HandleBuffer);
	• 0x0038:0x0000000A197 test ecx, ecx	SOURCE	
	Ox0038:0x00000000A197 jnle 0xA1977314 <coreraisetpl (unsign)<="" p=""></coreraisetpl>	SOURCE	if (gCpu == NULL) {
	Ox0038:0x00000000A197 jmp 0xA1977341 <coreraisetpl (unsigned)<="" p=""></coreraisetpl>	SOURCE	if (gSmmBase2 == NULL) {
1;	♦ 0x0038:0x0000000A197 shr rcx, 0x20	SOURCE	if (!EFI ERROR (Status) && !InSmm) {
ol);	0x0038:0x0000000A197 test ecx, ecx	SUORCE	II (:EFI_ERROR (Status) && :InShin) (
[pl));	0x0038:0x0000000000000000000000000000000		
	♦ 0x0038:0x00000000A19 or rdi, 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF		
	0x0038:0x00000000A19 jmp 0xA197733D <coreraisetpl (unsigne)<="" p=""></coreraisetpl>		
el, disable interru	0x0038:0x0000000A1 nov edi, 0x1F		
	• 0x0038:0x00000000A1 mp 0xA1977339 <coreraisetpl (unsigne<="" td=""><td></td><td></td></coreraisetpl>		
C 66 OldTpl < 1↓	@ 0x0038-0x000000021 c_rdi		
>	< >	<	>
	IP=0x0038:0x0000000A19772EA [DxeCore.efi] TPL C		
		B) (CORESETINTER	NOP ISTATE (
			Execution Trace
indow	ASM window		
			history
			instory
C			
Copyright" 2014, Intel C	orporation. All rights reserved. *Other brands and names are the property	i ot their respe	ective owners. (intel)

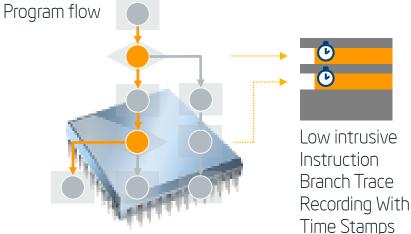
# Intel® System Debugger 2015 For Program flow

Low-overhead Processor Trace In New Intel<sup>®</sup> Core<sup>™</sup> M Processor

Intel<sup>®</sup> System Studio low-overhead hardware-assisted tracing capabilities help developers isolate

Program Execution History & Debugging

- Intel<sup>®</sup> Processor Trace low-overhead hardware based tracing capability on chip
- Now exposed with new Intel<sup>®</sup> Core<sup>™</sup> M processors



Intel Confidential

#### Isolate and resolve defects quickly

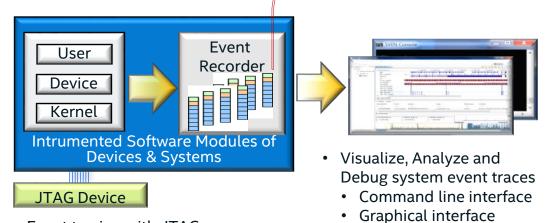


non-deterministic errors

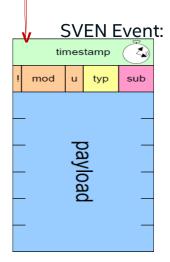
Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

(intel)

# SVEN SoC Trace – low overhead technology for static instrumentation of key SoC components



- Event tracing with JTAG
- Smart breakpoints
  - Interrupt execution on key events





## SVEN - A Stethoscope for your System System & SoC trace through JTAG

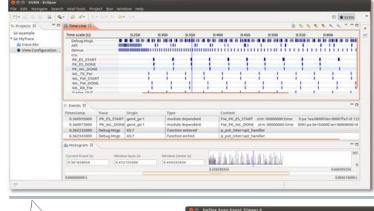
#### Trace Visualization

- Advanced navigation, search & filter
- Graphical and textual event display
- User controlled trace line grouping

#### Smart Event Triggers

- Live JTAG system debug with event tracing
- Smart breakpoints that interrupt execution on trace
- Set smart breakpoints for in-depth analysis For example:
  - Break on any event from the USB driver
  - Break on any Debug String that starts with "A
  - Break if register X is accessed
  - Break if register X bits [7-9] have value 0b101

#### Enhance system stability through powerful JTAG & event tracing



e e nt calls	Generic (Aug First Tag Fi Type: Def	pater	vent Tripper 0	Checkpole			sanys Defffffd	• Ust	- anyx	×.
<ul><li>Timeline view</li><li>Search &amp; Filter</li></ul>		itype in [] ini	0.0.0.0.0.1	Auto Trac Checkpoli Fatat Erro Function Function Generic Invalid Pa	si r Intered Inited	1	0101010	ielelelelele	Unit Do Medule	
Smart Event Trigger	Peyload Car DWORD(0)	Value: Mask	eanys (eanys (0x00000000	Taning Warning Kanyo		DWORD(1	Value: Maski	-any= 0+00000000	(*) (*)	
definition	bwcwo(2)	Value: Mask:	*#*9** 0x00000000			DWORD[3]	Value: Mask:			
	DWORD(4)	Value: Mask	*4/19* 0x00000000		+	DWORD(S)	Value: Mask	«#19/» 0x00000000		-
	Besult SVENTAG an	0049999	0000-0x01820000	(Lagtype-C	whogst	ring solitype	Oedep	eint)		
	æ							Cano		QK.

**Optimization Notice** 

## Software Instrumentation Example

```
static void ipc_message_received(
   struct HostIPC_Handler
                                   *hipc.
                                                                    WHO?
   struct Host IPC ReceiveOueue
                                   *rcv_q,
   const struct _IPC_MessageHeader
                                   *mh,
                                                                    DID WHAT?
   const char
                                    *message,
   unsigned int
                                    message_size )
                                                                    AND WHEN
{
   struct Host_IPC_QueuedMessage
                                   *msg;
   DEVH_FUNC_ENTER(hipc->devh):
    /** Queue the message for reading with HostIPC_GetNextInboundMessage() */
   DEVH_ASSERT( hipc->devh, (message_size > 0) );
    DEVH_ASSERT( hipc->devh, (message_size <= CONFIG_IPC_MESSAGE_MAX_SIZE) );
   DEVH_ASSERT(hipc->devh, (mh->ipc_mh_dst_gnum < CONFIG_IPC_HOST_MAX_RX_QUEUES));
   devh_ReadReg32( hipc->devh, CONFIG_IPC_ROFF_DOORBELL_STATUS );
   if ( NULL != (msq = HostIPC_GetFreeMessage(hipc)) )
       if ( message_size > CONFIG_IPC_MESSAGE_MAX_SIZE )
           message size = CONFIG IPC MESSAGE MAX SIZE:
       msg->mh = *mh; /* copy the header */
       memcpv( msg->msg, message, message_size );
       /* Add to inbound messages */
       OS_LIST_ADD_TAIL( &msq->node, &rcv_q->inbound_msqs );
   élse
       DEVH_WARN( hipc->devh, "HIPC_RX_OVF" );
   DEVH_FUNC_EXIT(hipc->devh):
```



## Intel<sup>®</sup> System Studio Application Debug and Trace

- Intel-enhanced GNU\* GDB
- Plug into existing Eclipse\* IDE for increased productivity
- Debug issues where symptoms are not visible immediately
- Remote debug with branch trace and data race detection
- Intel<sup>®</sup> enhanced GDB\* Debugger with pre-build binaries for Yocto Project\* and WR Linux\* targets

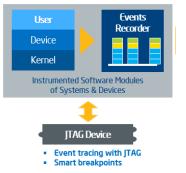


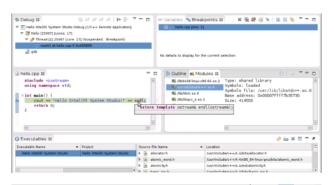
#### Detailed SoC & CPU System View

- Multiple cores (IA, DSP, other)
- User and kernel code

#### **Ultra-low Overhead Sampling**

- Can remain in production builds
- Around 1/10 of a microsecond
- Well defined event structure





**DEBUGGERS & TRACING** 

Application

System



## Intel<sup>®</sup> enhanced debug and trace solutions help to resolve issues fast



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

∕intel

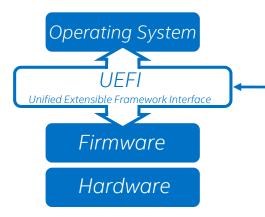
#### **DEBUGGERS & TRACING**

System

Application

## Intel<sup>®</sup> System Debugger Advanced UEFI Debugging

- Full access and visibility to UEFI through JTAG or USB connections
- Helps quickly isolate nondeterministic bugs



Dital(F) Debugger - Edit View Run Debug	Options Help									= [ LU
* # = = = = (	ित रह स्म । भ 🗉 🖣 🕴		a 🖬 M	6	🖳 😼 🖬 🖬 🖉	1 🐱 📾 🕸	<b>X</b> X			
Assembler: 50110:0x79F98232 t	c 0x0010.0x79F9833F 🖾			E pein		🗄 permentypoint.c	2			
rail Address	Optodes	Source	*	22 23	The entry point of	PE/COFF Image fo	r a PEIA	8		
Aub812:2x79F96262	55	push stp		24		Constants statistics for				
@ exc0012:2x79F90261	9B EC	moy cbp, esp		20	This function is t	ne entry paint fo	n a Pella	L this function must	call ProcessLibraryCons	tructorLis
pelment repair ( .c.:42		if (_gPermHeeision 1= 2] {		26	and ProcessinduleF	stryPointList().	The net	unn value from Process	HodulaEntryPointList() ms than gPoinRevison.	Es returne
@ exected : 2x79F98263	93 3D FC C7 F9 79 28	cmp dword ptr [2x79F9C7FC], 6x0		27	It _gPeinRevision	is not zero and P	eiservie	ics offer. Revision is le	ess than _gPeinRevison,	then ASSER
Antibilit: 8x79F96264	74 34	jr 0x79F902A0 < ModuleEntryPoint(void)	',class	28	Sparan MileHandle	Handle of the f	ile beir	g invoked.		
peimentrypoint.d:46		ASSERT (("PeiServices)->Hdr.Revis	ion >=		Boaram PeiService	Oescribes the 1	ist of t	ussible PHI Services.		
💩 excerta : 8x79190280	E6 CF 05 80 00	call 8x79F90048 <debugassertinabled(ex< td=""><td>&lt;(bio</td><td>35</td><td></td><td></td><td></td><td></td><td></td><td></td></debugassertinabled(ex<>	<(bio	35						
@ 0x00010:2x79F98271	0F 86 C0	movie can, al		32	@netval EFI_SUCCE	is The FEIM Exe	cuted no	ermally.		
execute: 8x79F90274	95 CØ	test eax, eax		22	@netval IEFI_SUCC	ISS The FEIN Tai	let to r	secure normally.		Î
0x8812:2x79F96276	74 24	12 Bx79F9819C < ModuleEntrePoint(void)	.class		EFT_STATUS					
@ 0x0010; 8x79F90279	90 4D QC	mov ecx, dword ptr [ebc+3x0]	1000000		EFIAPI					
@ 2008318:2x79F9E278	8B 11	mov ads, dword str [ecs]	- 12		NoduleEntryPoint [					
@ 0x0010:8x79F90270	90 42 09	mov max, dword ptr [edx+2x9]			IN EFT_PET_FILE_NA	VOLE FileHa				
Augente : Bit 79E 9E 288	38 #5 EC C7 E9 79			74 42	IN CONST EFT_PET_S	RVICES ***PeiS	ervices.			
@ 0x0012:2x79F96285	73 14	UEFI debugging		0 42	1					
Anion 8: 8:79596288	68 (8 (7 59 79	OLITUCDUYYIIIY		0 42	if ( gPeinRevision	1 . 27 4				
@ 0x0012:2x79F96280	66 2E	with full trace		43	11	and the second second				
One of the second se	CR 80 C7 F9 79	with full trace		.44	77 Bake sure that	t the PEI spet re	vitian a	of the platform is an A	El spec revision of the	e driker
ex001212x79F9E234	E8 37 08 20 00			45	11					
Orden 1 Chryst 90288     Orden 2: 2x79090288     Orden 2: 2x7909028     Orden 2: 2x7909028     Orden 2: 2x7909028     Orden 2: 2x7909028     Orden 2: 2x7909     Orden 2: 2x7909     Orden 2: 2x790     Orden     Orden 2: 2x790     Orden 2: 2x79	83 C4 0C	support	- P.U.	45 47	ASSERT ((*Peiser	vices)->Hdr.Revis	ion >= _	gPeimRevision);		
0x0001010x79F98295	33 (9	Support		47	3.					
• • • • • • • • • • • • • • • • •	75 CC	fing 2x79F9026C < ModuleEntryPoint(voi)	44	49	11					
perment rypoint, c:52	au	ProcessLibraryConstructorList (File		5.8	// Call constructo	- for all librari	05			
0x0001212x79F96242	90 55 92	mov edg, dword ptr [ebp+2xC]	nanulw,	51	11					
Otobela 1 847 97 96246	52	push eds		Ø 🖬 52	ProcessLibraryCons	tructorList (File	Handle,	PeiServices);		
exi0012:2x79F98:245	98 45 09	mov max, dword ptr [ebc+2x9]		63 54						
<ul> <li>0 0x001212x79F96244</li> <li>0 0x001212x79F96247</li> </ul>	98 45 89 50	nov sax, overa per [esp+axe]		55	IT Call the driver	enter noist				
© 0x0012:0x79F96245	E8 23 02 20 00	call 2x79F962D2 <processlibraryconstru< td=""><td></td><td>56</td><td>11- Carton Cartonia</td><td>a costa costa -</td><td></td><td></td><td></td><td></td></processlibraryconstru<>		56	11- Carton Cartonia	a costa costa -				
600001212679F96248	83 (4 (8	add esp. 008	uctorLi	C 57	return FrozessRody	eintryPointList	(File far	dle, Feisenvices);		
	83 C4 05	return ProcessHoduloEntrePoIntList		0 58	}					
peimontrypoint.c:57			(11093	50 60						
© 6x001.2:8x79F90202	66 40 80 51	now ecx, dword ptr [ebp+2x0]		61	1008					
example: 0:79F98283	51	push acir	6		-	(mm)		n functionality to 3	<pre>ioduleCotrePoint().</pre>	
	31	and the second of the second second	🕼 Sourc	e Files 23	Instruction Trace	[PT]		10		
Console View 🕄 🐻 Breaks	sorts 🔂 Locals					E	• 🕀 🐣			~ .
ugger Commands	proved suffrage of the		File Nan	e				Module	File	Address
race: PT configuration:	Trace enabled		18.	puIoPei	ofi					
<pre>b&gt; @COTO_PEIN(CpuIoPei)</pre>		The second second second second second		eiCore.				_PET_SI CouloPet.eft	peimentrypoint.c:57	8x79F98
		: enabled (S=0,CS=0,HK=0) CADINAGELCADINAGE\ELINE 485"						WND_OFF Peicore.efi	dispatcher.c:690	Ex79FAB
EAKFOINT & AT WADDRESS of		CHDINAGELCHDINAGEUGLINE 485-	See 2	mAcces	s.eti			ff*,cla:PeiCore.efi	peimain.c:346	Øx79€90
n until blockend at addr										<b>CHFFFED</b>
EARFOINT & AT (addressed	000000079990200) : enable	d (5=0,CS=0,NW=0)								Barrres.
ogram stopped: SREAKPOIN										
EAKPOINT & AT BPEIN_ENTR				IFFI	modules	in				
		0 00000022200079698262 for PE/CCFF header (s	_ C		modules					
ign=0x222001000 range=0x0		Skeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee		+ 1-	a. a					
76 bytes]		and control in secon carrypoints and control of		-tn	e system					
	dening control - found at of	fset 8x8200000079=98000:								

#### Improve System Boot Time And Reliability



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

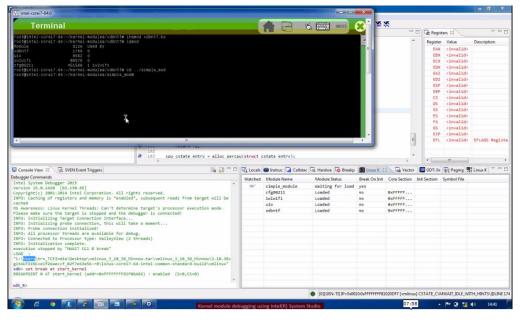
3

inte

## Linux\* Kernel Module Device Debug on Intel® Atom™ E38xx - MinnowBoard MAX

Video Demo





#### Dynamically loaded device driver debug without instrumentation



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.



**DEBUGGERS & TRACING** 

Application

System

## Accelerate Debug and Trace with Intel<sup>®</sup> System Studio

- Intel-enhanced GNU\* GDB
- Process specific instruction trace and interactive data race debug
- Accelerates finding and resolving system software reliability issues

Intel<sup>®</sup> System Debugger

- Deep insight into application and system software
- Understanding memory and platform configuration
- Full-features low-cost JTAG debug solutions for makers and IoT edge devices

SVEN Technology SDK

• Ultra-low overhead data event tracing for hard to reproduce problems

## Accelerate Reliability and Time-to-Market







NEW

JTAG devices available for

<\$100



34

DEBUGGERS & TRACING
System Application

## Analyzers



## Analyzer Overview

Power & Performance CPU / Graphics Memory & Threading

	Intel® VTune™ Amplifier for Systems	Intel® Energy Profiler	Intel® Inspector for Systems	System Analyzer	Platform Analyzer	Frame Analyzer
What does it provide	In-depth analysis of CPU performance	System-wide energy analysis	Memory and thread analyzer	Overview of system performance with real-time view	Overview of system performance with offline view	Single-frame graphics analysis for DirectX and OpenGL ES workloads
Key Purpose	Optimize compute workloads for CPU	Boost power efficiency and battery life	Enhance system and application robustness	Optimize graphical workloads across CPU/GPU	Optimize graphical workloads across CPU/GPU	Optimize graphical workloads at the detailed draw call level
How	Uses system counters and sampling to identify performance hot spots	Uses processor sleep states to identify power sapping execution	Identifies memory and threading conflicts	Make real-time experiments with graphics	Make offline analysis of CPU and GPU metrics and workloads	Experiment with graphics workloads to improve performance and visual quality

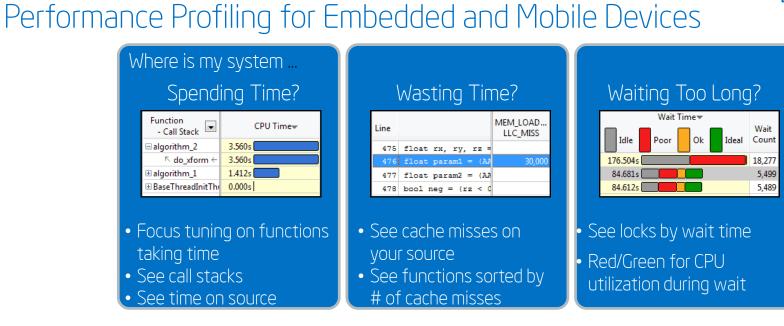
Comprehensive Platform Analysis Capabilities To Boost Power Efficiency, Performance And Robustness



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

36

înte



- Linux & Windows hosted cross sampling for Embedded & Mobile target devices
- Low overhead & easy to use

Intel<sup>®</sup> VTune<sup>™</sup> Amplifier for Systems

No special recompiles

### Advanced system profiling for scalable performance

ANALYZERS

Memory & Threading

Power & Performance

CPU / Graphics

#### Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

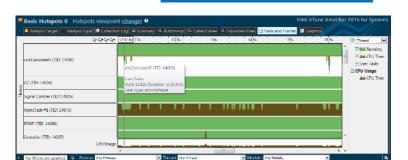
Enhanced Intel<sup>®</sup> VTune<sup>™</sup> Amplifier For Systems

- Easy remote target collection through graphical interface
- Android\* 64-bit Lollipop\* and Android\* Run-Time (ART) support
- Correlate Android Systrace and Linux\* Perf or F-Trace information with other performance data
- Advanced Android graphics and life cycle events analysis
- Drivers for Java performance analysis now preinstalled into Android Lollipop simplifying usage

Target type: Laur Android device (ADB) remote Linux (SSH) Graphical interface to collect data on a remote Linux\* system via SSH or Android\* via ADB

Binary/Symbol Search Source Search

remote Linux (SSH)



Target

Connection type:

Correlate Android Framework information with other VTune™ performance data

Improve Performance and End-User Experience of Intel® Architecture-based Devices





Power & Performance Memory & CPU / Graphics Threading

## Intel<sup>®</sup> VTune<sup>™</sup> Amplifier for Systems Remote Performance Data Sampling

### Video Demo



A DE LOCAL	🔋 🕨 🕼 🎬 🕐 Welcome r001ah 🗙				International	al VTuna Amplifiar 201	E for Sunt		
Advanced Hotspots Hotspots viewpoint ( <u>change</u> ) 0 Intel VTune Amplifier 2015 fo									
😌 Ar	alysis Target 💫 Analysis Type 🔜 Collection Log 🖬 Summary 🐝	Bottom-up Caller/Callee Top-dov	vn Tree 📴 1	Tasks a	nd Frames 🛛 🗞 grid.cpp 🛛 🔯 grid.cpp 🕺		211		
Source	Assembly 📰 📰 💿 🐵 🕸 😰 🔍 Assembly g	ouping: Function Range / Basic Block / Address							
ur	Source	CPU Time: Total by Utilization	Func Rang*	Sour.	Assembly		CPU Tir		
ine	300/02	🗐 Idle 📕 Poor 📋 Ok 🛢 Ideal 📑 Over 🗏	Rang_	Line	Assembly		Idle Poor		
1	if (ry->mbox[cpg->ob]->id] != ry->serial) {	11.564s	0x409c02	581	lea (%rax,%rdx,4), %rax		5.516		
3	cur->obj->methods->intersect(cur->obj, ry);	1.406s	0x409c06	581	mov1 0x10(%r15), %edx		0.257s		
	else if (tmax.z < tmax.y) (	1.046s	0x409c0a	581	cmpl %edx, (%rex)		0.162s		
2	if (ry->mbox[cur->ob]->id] != ry->serial) (	0.6995	0x409c0c	581	jz 0x409c19 <block s7=""></block>		1.4705		
0	while (cur != NULL) (	0.671;	0x409c0e	0	Block 56		1.802s		
0	if (ry->mbox[cur->ob]->id] != ry->serial) {	0.618:	0x409c0e	582	movl %edx, (%rax)		0.396s		
z	ry->mbox[cur->obj->id] = ry->serial;	0.396s	0x409c10	583	movq 0x10(%rdi), %rax		0.467s		
5	cur = cur->next;	0.377s	0x409c14	583	mov %r15, %rsi		0.824s		
8	if (ry->maxdist < tmax.z    curvox.z out.z)	0.1855	0x409c17	583	callg (%rax)		0.1165		
	curpos = nZp;	0.150s	0x409c19	0	Block 57		0.377s		
	if (Imax.x < Imax.y 44 Imax.x < Imax.z) (	0.135s	0x409c19	585	movg (%r14), %r14		0.3775		
6 (		0.131s	0x409c1c	0	Block 58		0.671s		
1 3		0.1265	0x409c1c	580	test %r14, %r14		0.524s		
	while (cur != NULL) {	0.1115	0x409c1f	580	jnz 0x409bf8 <block 55=""></block>		0.1475		
7	curvox.z += step.z;	0.101s	0x409c21	0	Block 59		0.227s		
	<pre>tmax.z += tdelta.z;</pre>	0.089s	0x409c21	587	mov1 0xb8(%rsp), %edx		0.033s		
	cur->obj->methods->intersect(cur->obj, ry);	0.083s	0x409c28	588	movadg 0x10(%rap), %xmm2		0.093s		
2	<pre>cur = g-&gt;cells[voxindex];</pre>	0.077s	0x409c2e	587	add %r12d, %edx		0.023s		
2	tmax.z = offset + ((voxel2z(g, curvox.z + 1) - curpos.	0.065s	0x409c31	588	ucomisdg 0x50(%r15), %xmm2		0.020s		
2	cur->obj->methods->intersect(cur->ob), ry);	0.065s	0x409c37	587	movl %edx, 0xb8(%rsp)		0.045s		
0	<pre>voxindex += step.z*g-&gt;xsize*g-&gt;ysize;</pre>	0.063s	0x409c3e	588	jnbe 0x40963a <block 2=""></block>		0.014s		
)	tdelta.z = g->voxsize.z / ry->d.z:	0.062s	0x409c44	0	Block 60		0.059s		
1	tmax.x = offset + ((voxel2x(g, curvox.x + 1) - curpos.	0.060s	0x409c44	588	cmpl %edx, 0x7c(%rsp)		0.027s		
3	nZp.x += pdeltaZ.x;	0.056s	0x409c48	588	jz 0x40963a <block 2=""></block>		0.032s		
)	tmax.y = offset + ((voxel2y(g, curvox.y + 1) - curpos.	0.054s	0x409c4e	0	Block 61		0.423s		
5	tmax.y = offset + ((voxel2y(g, curvox.y) - curpos.y) /	0.048s	0x409c4e	590	mov1 0x20(%rbx), %edx		0.024s		
6	tdelta.y = g->voxsize.y / - ry->d.y;	0.048s	0x409c51	592	movadg 0x30(%rsp), %xmm5		0.078s		
	tmax.z = offset + ((voxel2z(g, curvox.z) - curpos.z) /	0.048s	0x409c57	592	movadg 0x28(%rsp), %xmm4		0.008s		
	while (cur != NULL) {	0.047s	0x409c5d	592	movadg %xmm5, 0xc0(%rap)		0.0145		
	<pre>tmax.x = offset + ((voxel2x(g, curvox.x) - curpos.x) /</pre>	0.045s	0x409c66	592	movadg 0x20(%rsp), %xmm2		0.009s		
7	tdelta.x = g->voxsize.x / - ry->d.x;	0.039s	0x409c6c	590	imul %r12d, %edx		0.012s		
2	tdelta.x = g->voxsize.x / ry->d.x:	0.038s	0x409c70	591	movadg 0x10(%rsp), %xmm3		0.008s		
4	nZp.y += pdeltaZ.y;	0.035s	0x409c76	592	movadq %xmm4, 0xc8(%rap)		0.024s		
5	n2p.z += pdelta2.z;	0.032sl	0x409c7f		movedg %xmm2, 0xd0(%rsp)		0.018=		

#### Boost performance with remote system-level and application level analysis

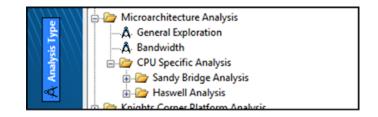


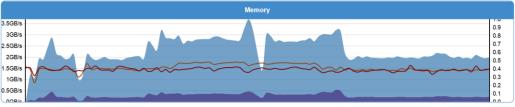


ANALYZERS

Power & Performance Memory & CPU / Graphics Threading Intel® System Studio 2015 - update 2 - Enhanced Developer Productivity Intel® VTune™ Amplifier for Systems added bandwidth analysis

- Preset profiles provide an easy "point and shoot" set-up. No memorizing complex event names. Advanced profiles like bandwidth analysis, cache analysis and branch mis-predictions find tuning opportunities
- Now with SoC, DDR memory and GPU bandwidth analysis





#### More performance insights into Intel<sup>®</sup> Architecture SoC components and peripherals

Intel<sup>®</sup> Confidential – CNDA Required



# Intel<sup>®</sup> System Studio Graphics Analysis Tools

- Real-time system-level performance analysis for Intel-based Android and Windows devices
- Immediate experiments and overwrites enable developers to isolate CPU and GPU performance problems
- Metrics for CPU, GPU, API, memory, network, power, etc.
- Frame-level graphics tuning for DirectX and OpenGL ES workloads



ANALYZERS



Optimize CPU and Intel<sup>®</sup> HD Graphics performance with real-time profilers



### Intel<sup>®</sup> Energy Profiler



### Enhanced Intel<sup>®</sup> Energy Profiler



- Available now for Windows\* targets
- Supports new Intel<sup>®</sup> Core<sup>™</sup> M processor
- Correlates system activity to source code to identify power sapping implementations
- New power data for graphics processors and DRAM self-refresh

Grouping: Wake-up Reason / Wa	ke-up Object / Core					▼ L→■ Q	
Wake-up Reason / Wake	-up Object / Core	Total Wake-up Count	Process Name	ProcessID	ThreadID	Core Sleep State	
🗄 User Timer		568	socwatch	6719	6719	C6	
⊞ User Timer		548	irq/67-intel	78	78	C1	
⊞ User Timer		209	kworker/0:8	1111	1111	C1	
🗆 User Timer		195	mediaserver	151	270	C6	
core_2		160	mediaserver	151	270	C6	
core_3		35	mediaserver	151	270	C6	
🗉 User Timer		193	kworker/0:4	23310	23310	C1	
User Timer			kworker/0:3	6560	6560	C1	
	Selected 1 row(s):	195					
Q∞Q+Q-Q+         Zi           package_0         III           iffg         core_0         III           core_1         III         IIII	3695 55 225				العلية ( العلية ( العلية ( العلية 1 العلية (	C-States Time C6 C1	nf

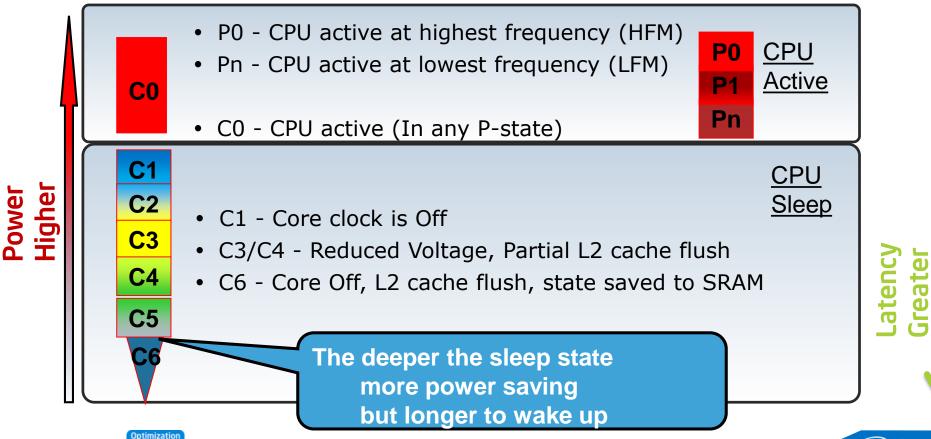
#### Improve Power Efficiency for Intel<sup>®</sup> Architecture-based Devices



43

# CPU C-States / P-States

Notice



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

intel

## Components of power analysis

Idle vs. Active

- Do nothing efficiently
- Hurry up and get idle.
  - e.g. Multi-threading (distributing work evenly across cores)

#### **Optimize Sleep Behavior**

- Minimize sporadic wakeups.
- Schedule all periodic activities from the app into same wakeup period.
- What is waking h/w from low power states? Why?

#### **Optimize Utilization**

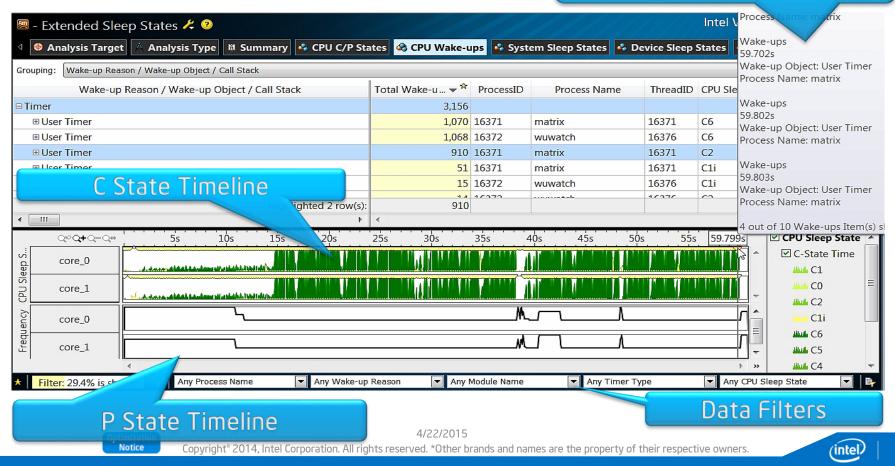
- What is active? Why is it active?
- Minimize Polling Loops. Use event driven framework when possible.
- Turn devices off. Open devices can prevent the system from entering power saving state.





### C/P States Tab

#### Timeline tooltips show wakeup cause



What C-State residency numbers tell you

Residency in the deepest C-state should be >95% @ idle

- If you see high residency in CO state
- CPU is executing instruction
- Next Step: Active Analysis
- If you see high residency in the intermediate C-states
- Frequent active/idle transition is keeping CPU from entering deepest Cstate. Possible causes are:
  - Application scheduling periodic timer with short period activity
  - Application waits for interrupts (from device, IPI) very frequently



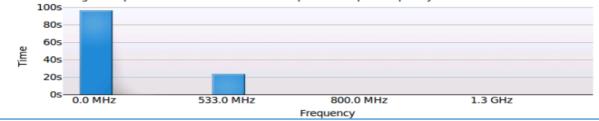


# Comparisons

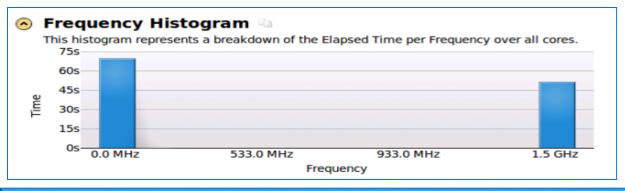
#### Native Video Playback vs 3rd Party Application

#### 📀 Frequency Histogram 🗈

This histogram represents a breakdown of the Elapsed Time per Frequency over all cores.



#### Native Video Playback



### 3<sup>rd</sup> Party Application

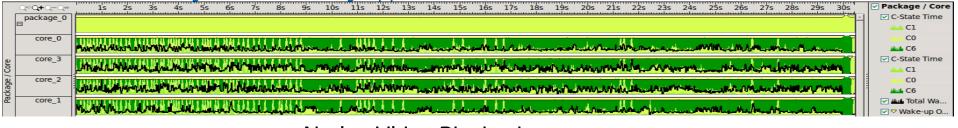
### Which is "Better"?



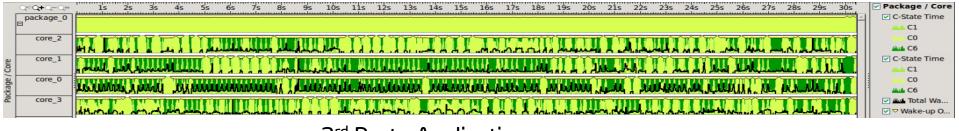


# Comparisons

### Native Video Playback vs 3rd Party Application



#### Native Video Playback



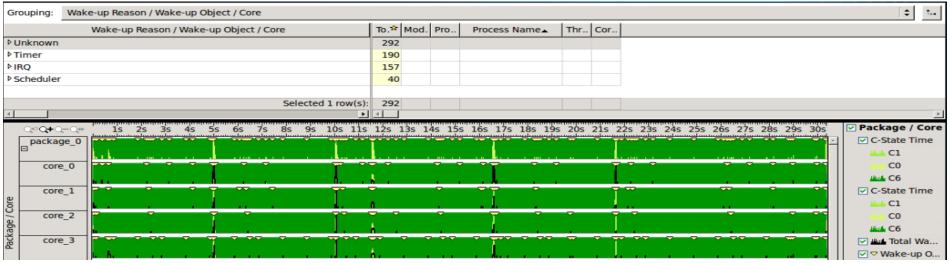
3<sup>rd</sup> Party Application

### Much more time spent in CO for the 3<sup>rd</sup> part application





### Comparisons Adding a new Kernel Driver



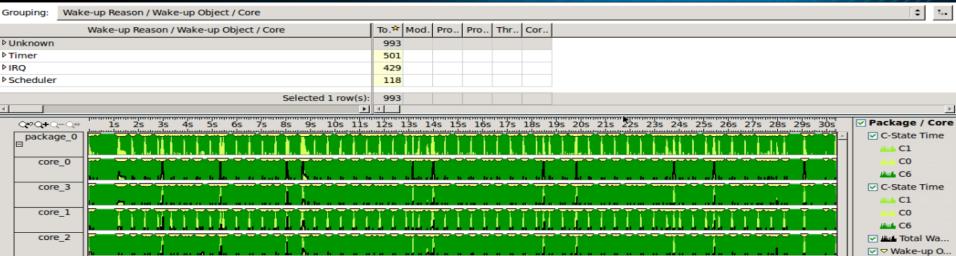
Idle Workload

### Note the Wake-up Counts





### Comparisons Adding a new Kernel Driver



#### New Driver

### Note the Wake-up Counts





# Comparisons

### Adding a new Kernel Driver

### Reduce the kernel logging (printk) in driver

```
void my timer callback ( unsigned long data )
   int i, j, ret;
   for (i=0;i<5000000;i++) {</pre>
     for (j=0;j<5000000;j++) {
        sum=sum+i:
   printk( "my timer callback called %d\n", count );
  } }
     setup the timer again to fire 500ms
  if (count <50) {
     setup timer( &my timer, my timer callback, 0 );
     ret = mod timer( &my timer, jiffies +
msecs to jiffies(500) );
     if (ret) printk("Error in
my timer callback\n");
     count++;
```

```
void my timer callback( unsigned long data )
   int i, j, ret;
   for (i=0;i<5000000;i++) {</pre>
     for (j=0; j<5000000; j++) {</pre>
        sum=sum+i:
   //printk( "my timer callback called %d\n", count );
  } }
  // setup the timer again to fire 500ms
  if (count <50) {
     setup timer( &my timer, my timer callback, 0 );
     ret = mod timer( &my timer, jiffies +
msecs to jiffies(500) );
     if (ret) printk("Error in my timer callback\n");
     count++;
  else {
   printk( "callback called %d times\n",count);
```

### VTune Amplifier Comparison Feature









### Intel® System Studio 2015 Value Proposition

Accelerate Time To Market	Strengthen System Reliability	Boost Power Efficiency and Performance	Category	Component						
√		$\checkmark$		Intel <sup>®</sup> C++ Compiler						
$\checkmark$		$\checkmark$	Compilar 9	ntel® Integrated Performance Primitives						
$\checkmark$		$\checkmark$	Compiler & Libraries	Intel <sup>®</sup> Math Kernel Library						
$\checkmark$		$\checkmark$	Libraries	Intel® Threading Building Blocks						
$\checkmark$	$\checkmark$		Application Debugger	Intel-enhanced GDB* Application Debugger						
$\checkmark$		$\checkmark$		Intel <sup>®</sup> VTune™ Amplifier for Systems						
$\checkmark$		$\checkmark$		Intel <sup>®</sup> Energy Profiler						
$\checkmark$		$\checkmark$	Angluzors	System Analyzer						
$\checkmark$		$\checkmark$	, Analyzers	Frame Analyzer						
$\checkmark$		$\checkmark$		Platform Analyzer						
$\checkmark$	$\checkmark$			Intel <sup>®</sup> Inspector for Systems						
$\checkmark$	$\checkmark$		System Debugger	Intel <sup>®</sup> System Debugger (JTAG)						



(inte

# Next Steps – Getting Started

### Download and evaluate

<sup>Intel\*</sup> System Studio (intel)



### Get support

User Forum:

- <u>http://software.intel.com/en-us/forums/intel-system-studio</u> Intel Premier Support:
- <u>https://premier.intel.com</u>, Product: Intel<sup>®</sup> System Studio Articles, User Guides and Getting Started:
- <u>https://software.intel.com/en-us/articles/intel-system-studio-articles</u>
- <u>https://software.intel.com/en-us/articles/intel-system-studio-release-notes</u>

#### Contact us for next generation Intel® Processor support:

IntelSystemStudio@intel.com





Intel <sup>®</sup> System Studio 2015 Components			Target OS Support									
			Linux* <sup>1, 5</sup>			Android* <sup>5</sup>			ows*	VxWorks*		
Category	Component	Composer Edition	Professional Edition	Ultimate Edition	Composer Edition	Professional Edition	Ultimate Edition	Composer Edition	Professional Edition	Composer Edition		
Host Operating Systems		Linux*, Windows*			Linux*, Windows*			Windows*		Linux*, Windows*		
Integrated Development Environment			Eclipse*, Wind River* Workbench*			Eclipse*			tudio*	Wind River* Workbench*		
	Intel <sup>®</sup> C++ Compiler		$\checkmark$	$\checkmark$	√	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	<b>√</b> <sup>2</sup>		
Compiler &	Intel® Integrated Performance Primitives		$\checkmark$	√	√	$\checkmark$	√	$\checkmark$	√	<b>√</b> <sup>2</sup>		
Libraries	Intel® Math Kernel Library		√	√				$\checkmark$	√			
	Intel® Threading Building Blocks	√	√	$\checkmark$	√	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$			
Application Debugger			√	$\checkmark$	√	$\checkmark$	√					
	Intel <sup>®</sup> VTune™ Amplifier for Systems		√	$\checkmark$		$\checkmark$	√		$\checkmark$			
	Intel® Energy Profiler		√6	√6		$\checkmark$	√		√			
A	System Analyzer Platform Analyzer <sup>4</sup>					$\checkmark$	√		√			
Analyzers						$\checkmark$	$\checkmark$		$\checkmark$			
	Frame Analyzer <sup>4</sup>					$\checkmark$	$\checkmark$		$\checkmark$			
	Intel <sup>®</sup> Inspector for Systems		√	$\checkmark$					$\checkmark$			
System Debugger	System Intel <sup>®</sup> System Debugger (ITAG) <sup>3</sup>			~			√					
1	1 Linux*	Embeddeo	LLinux Winc	Pivor* Linu	v* Vocto Dr	oioct* Tizon	*					

<sup>1</sup> Linux\*, Embedded Linux, Wind River\* Linux\*, Yocto Project\*, Tizen\*

<sup>2</sup> Delivered with Wind River\* VxWorks\* platform\*

<sup>3</sup> Via Intel<sup>®</sup> ITP-XDP3 probe, OpenOCD<sup>\*</sup>, Macraigor\* usb2demon\* and EDKII\* for UEFI\*

<sup>4</sup> Available on Windows\* host only

<sup>5</sup> Linux\* and Android\* target support available in a single product <sup>6</sup> For detailed processor support please visit: <u>https://software.intel.com/en-us/intel-energy-profiler</u>



Copyright° 2014, Intel Corporation. All rights reserved. \*Other brands and names are the property of their respective owners.

(intel

# What Are You Developing For?

Systems and IoT

IA powered hardware, Embedded Systems, Mobile Systems, IoT



•

•

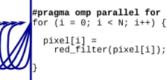
•

### Responsiveness

Ultrabooks, OS X devices, Tablets, Phones

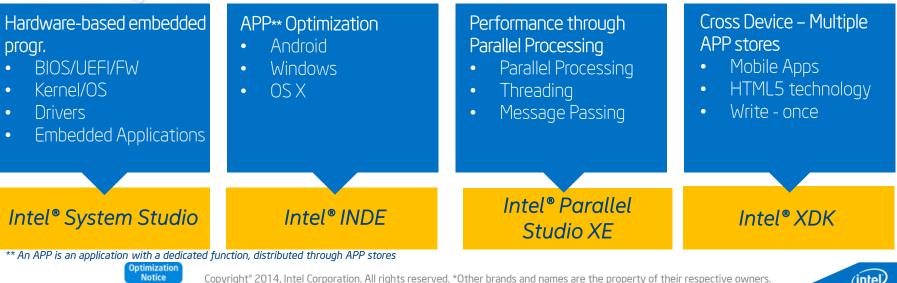


#### Scientific & Performance HPCs, servers, clusters



**Multiple Devices** Ultrabooks, Tablets, Phones





### Intel<sup>®</sup> IoT Developer Kit – Product Brief

Intel<sup>®</sup> IoT Developer Kits are versatile, performance-optimized and fully integrated end-to-end IoT solutions supporting a variety of programming environments, tools, security, cloud connectivity and hardware such as Intel<sup>®</sup> Edison, Intel<sup>®</sup> Galileo and Intel<sup>®</sup> Gateway Solutions for IoT. The Intel IoT Developer Kit lowers the barriers to entry by combining a small, powerful and production ready hardware & software platform together with a comprehensive IoT Developer program and a highly engaged global developer community.

#### Software included in the IoT Dev Kit:

- The Yocto\* Linux system
  - Provides resources for creating applications in various programming languages: C/C++, Python, Node.js and visual programming
- Integrated Development Environments (IDEs) and Tools
  - Allows developers to create, run and debug applications directly on the Galileo and Edison boards: Eclipse\* (C/C++), Intel® XDK IoT Edition (Node.js), Wyliodrin\* (Visual)
  - Intel<sup>®</sup> System Studio for IoT: development suite that provides deep hardware and software insights to speed development, testing and optimization
- Intel<sup>®</sup> IoT Analytics
  - Provides cloud APIs for data collection, data visualization, reports, rules engine and analytics
- Middleware libraries
  - Provides developers high level API access to Galileo and Edison boards and middleware libraries to easily control the various sensors and actuators













### Intel Compiler Benchmark Configuration Information

C++ benchmark configuration Info - SW Versions: Intel<sup>®</sup> C/C++ 14.0, Microsoft Visual C++ 2012 (Windows), GCC 4.8.1 (Linux); Hardware: HP ProLiant DL360 G8\*, 2 x Intel<sup>®</sup> Xeon<sup>®</sup> processor E5-2670 (2.60GHz, 20480KB LLC, TurboBoost is on, HyperThreading is on), 128GB RAM, SAS; Windows Operating System: Windows 7 Enterprise, Service pack 1; Linux Operating System: Red Hat Enterprise Linux Server release 6.2 (Santiago), Kernel 2.6.32-220.el6.x86\_64. ||| Compiler Options: INT Speed (Linux): Intel compiler 14.0- C: -xAVX -ipo -O3 -no-prec-div -static -parallel -opt-prefetch -auto-p32. C++: xAVX -ipo -O3 -no-prec-div -opt-prefetch -auto-p32. GCC 4.8.1- C: -m64 -Ofast -ffast-math -flto -march=native -mfpmath=sse -funroll-all-loops -static -ftree-parallelize-loops=16. C++: -m64 -Ofast -ffast-math -flto march=native -mfpmath=sse -funroll-all-loops. ||| FP Speed (Linux): Intel Compiler 14.0- C: -xAVX -ipo -O3 -no-prec-div -static -opt-prefetch -ansi-alias. C++ xAVX -ipo -O3 -no-prec-div -static -opt-prefetch -ansi-alias. GC 4.8.1 - C:-m64 -Ofast -ffast-math -flto -march=native -mfpmath=sse -funroll-all-loops -static -ftreeparallelize-loops=16. C++: -m64 -Ofast -ffast-math -flto -march=native -mfpmath=sse -funroll-all-loops. ||| INT Speed (Windows): Intel compiler 14.0- C: -Qvc11 -Qstd=c99 -QxAVX -Qipo -O3 -Qprec-div - Qopt-prefetch -Qparallel -Qauto-ilp32. C++: -Qvc11 -QxAVX -Qipo -O3 -Qprec-div - Qopt-prefetch -Qcxx\_features -Qautoilp32. Visual C++\* 2012- C: /O2 /Ob2 /GL /arch:AVX /favor:EM64T /fp:fast /Qpar. C++: /O2 /Ob2 /GL /arch:AVX /favor:EM64T /fp:fast /Qpar -EHsc -GR. ||| FP Speed (Windows): Intel compiler 14.0- C: -Qvc11 -Qstd=c99 -QxAVX -Qipo -O3 -Qprec-div - Qopt-prefetch -Qansi-alias -Qparallel. C++: -Qvc11 -QxAVX -Qipo -O3 -Qprec-div -Qopt-prefetch -Qcxx\_features -Qansi-alias. Fortran: -QxAVX -Qipo -O3 -Qprec-div - Qopt-prefetch -Qansi-alias -Qparallel. C++: -Qvc11 -QxAVX -Qipo -O3 -Qprec-div -Qopt-prefetch -Qcxx\_features -Qansi-alias. Fortran: -QxAVX -Qipo -O3 -Qprec-div - Qopt-prefetch -Qparallel.

Fortran benchmark configuration Info - Compiler Versions: Intel® Fortran 14.0, PGI 13.6;, Absoft 13.0.3, gFortran 4.8.1; Hardware: Blue Hills ATX Media IVB Desktop DZ77BH-55K-IDD; Intel® Core™ i7-3770K CPU @ 3.50GHz, TurboBoost is on, HyperThreading is off, 16GB RAM; Windows Operating System: Windows 7 Enterprise, Service pack 1; Linux Operating System: Red Hat Enterprise Linux Server release 6.3 (Santiago); Kernel 2.6.32-279.el6.x86\_64 ; Compiler Options (Windows and Linux): Intel Fortran compiler 14.0: ifort -03 -fast -parallel -ipo -no-prec-div , PGI 13.6: pgf95 -fastsse -Munroll=n:4 -Mipa=fast,inline -Mconcur=bind . Polyhedron benchmark (www.polyhedron.com) performed by Intel Corp on August 20, 2013.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. \* Other brands and names are the property of their respective owners. Benchmark Source: Intel

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804



60

### Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Copyright<sup>®</sup> 2014, Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Xeon Phi, Core, VTune, and Cilk are trademarks of Intel Corporation in the U.S. and other countries.

The cost reduction scenarios described in this document are intended to enable you to get a better understanding of how the purchase of a given Intel product, combined with a number of situation-specific variables, might affect your future cost and savings. Nothing in this document should be interpreted as either a promise of or contract for a given level of costs.

#### **Optimization Notice**

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

