

Graphics Performance Analyzer for Android



What you will learn from this slide deck

- Detailed optimization workflow of Graphics Performance Analyzer

Android* System Analysis Only

- Please see subsequent slide decks for in-depth technical training

Game Analysis & Optimization Tools

Intel® Graphics Performance Analyzers



Optimize Performance

- Optimize performance on Intel® Core™ and Intel® Atom™ processors
- Intuitive Intel® GPA Tools improve efficiency
- Real time CPU and GPU power metrics increase power efficiency

System Analyzer:
Real-time
game analysis



Hosts: Windows*, Ubuntu Linux*,
OS X*¹

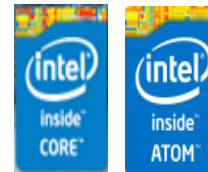
Platform Analyzer:
CPU and Intel
Graphics metrics



Hosts: Windows

Targets: Windows, Android

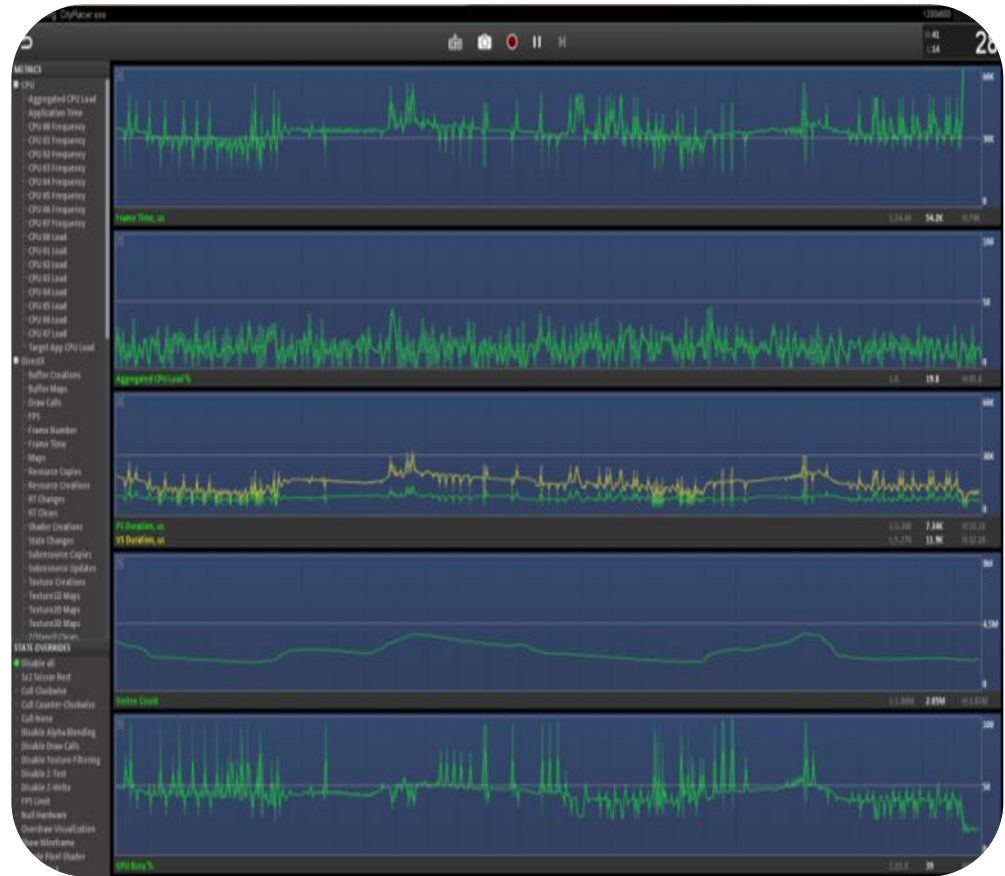
Supported Processors & Graphics





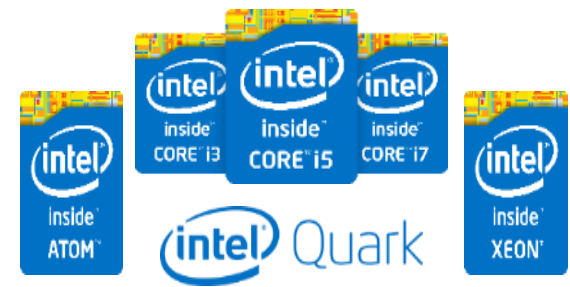
Intel® Graphics Performance Analyzers System Analyzer

- Real-time system-level performance analysis for Intel-based Android devices
- Immediate experiments and overwrites enable developers to isolate CPU and GPU performance problems
- Metrics for CPU, GPU, API, memory, network, power, etc.



Drag 'n Drop Metrics
Real-time Experiments

Support for Latest Intel Processors & SoCs



	Intel® Quark SoC X1000	Intel® Atom™ Processors	Haswell microarchitecture
Intel® JTAG Debugger [†] – System Debug ¹	✓	✓	--
Enhanced GDB* Debugger – Application Debug	✓	✓	✓
Intel® Inspector – Memory & Threading Analysis	--	✓ Memory & Thread Analysis	✓ Memory & Thread Analysis
Intel® VTune™ Amplifier ^{††} – Power & Performance	--	✓ Hardware Events	✓ Hardware Events
Intel® Graphics Performance Analyzers	--	✓	✓
Intel® C++ Compiler	✓	✓ SSSE3	✓ SSE, AVX, AVX2, FMA3
Intel® MKL library	--	--	✓ SSE, AVX, AVX2, FMA3
Intel® IPP library	✓	✓	✓

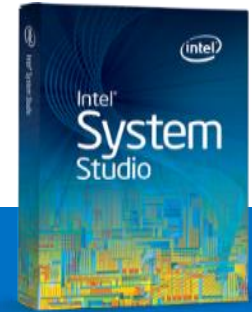
[†] Hardware platform debug for new processors added as new processors ship

^{††} Hardware events for new processors added as new processors ship

¹ optional



Integrated & Comprehensive Development Suite



Accelerate Time To Market	Strengthen System Reliability	Boost Power Efficiency and Performance		
✓	✓		Debuggers & Trace	Intel® JTAG Debugger 2014 ¹
				GDB* Debugger 7.5
✓	✓			SVEN Technology 1.0
✓		✓	Analyzers	Intel® VTune™ Amplifier 2014 for Systems
				Intel® Graphics Performance Analyzers (Android*)
✓	✓			Intel® Inspector 2014 for Systems
✓		✓	Compiler & Libraries	Intel® C++ Compiler 14.0
✓		✓		Intel® Integrated Performance Primitives 8.1

Deep system-level insights into power, reliability and performance which help accelerate time to market of Intel Architecture-based embedded and mobile systems

.1

Intel® System Studio 2014 – What's New?

Broader host and target OS coverage

- New Tizen™ IVI, Embedded Android*
- New Yocto Project* 1.5; Wind River* Linux 6
- New Windows* host development for Linux*-based targets

New Intel® Architecture

- New Intel® Quark SoC X1000-based platforms
- New support for Intel® Atom™ processor E3xxx & C2xxx series
- New support for 4th generation Intel® Core™ processor

Enhanced Eclipse* integration & cross-build

- Automated Eclipse* IDE Integration on Linux* and Windows* hosts
- Enhanced cross-build sysroot support and Wind River* Workbench integration
- OpenEmbedded* 3rd party toolchain layer recipes

New features across all key components

- Intel® C++ Compiler and libraries generated code compatible with Wind River Simics*
- Intel® JTAG Debugger 2014 - New support for Intel® Core™, Intel® Xeon™ & Intel® Quark processors; Agent based UEFI debug
- GNU* GDB - Branch Trace Store (btrace) for Intel® Atom™ or 4th generation Intel® Core™ Processors
- Intel® VTune™ Amplifier 2014 for Systems - Adds system-wide event-based sampling of uncore and SoC memory bandwidth
- Intel® C++ Compiler 2014 - Optimizations for the latest Intel® processor generation
- Intel® Integrated Performance Primitives 8.1 - Adds new signal processing features for LTE, and optimizations for the latest Intel® processor generation
- Intel® Math Kernel Library 11.1 - Optimizations for the latest Intel® processor generation
- Intel® Graphics Performance Analyzers - Supports GPU optimization for Android*

Intel® System Studio 2014

Phase	Component	Feature	Benefit
Verify, Debug & Flash	Intel® JTAG Debugger 2014 ¹	In-depth system and application debug <ul style="list-style-type: none"> Intel® Core & Intel® Xeon & Intel® Quark support Source level debug of OS kernel software, drivers, firmware, BIOS, UEFI SVEN 1.0 - Ultra-fast software SoC trace infrastructure for debug 	<ul style="list-style-type: none"> In-depth debug insight into CPU, SoC and chipset for fast issue resolution Leave trace instrumentation in production code for fast system-wide issue resolution
	GDB* Debugger	<ul style="list-style-type: none"> Enhanced GDB* application debugger 	<ul style="list-style-type: none"> Detailed application debug and trace for fast issue resolution Data race detection in parallel software
	Intel® Inspector 2014 for Systems	<ul style="list-style-type: none"> Memory & threading analysis for improved code quality 	<ul style="list-style-type: none"> Increased productivity and code quality, and lowers cost, finds memory, threading, and security defects before they happen
Tune	Intel® VTune™ Amplifier 2014 for Systems	In-depth software analysis and tuning <ul style="list-style-type: none"> SoC wide analysis Power and performance profiling 	<ul style="list-style-type: none"> Fast in-depth analysis of SoC behavior Remove guesswork, saves time, makes it easier to optimize for power efficiency and find performance optimization opportunities In-depth analysis on resource limited targets
	Intel® Graphics Performance Analyzers	<ul style="list-style-type: none"> In-depth GPU analysis for Android* 	<ul style="list-style-type: none"> Optimize user experience of IA-based Android* devices
Build	Intel® C++ Compiler 14	Compiler and performance libraries <ul style="list-style-type: none"> High-performance C/C++ cross compiler; Intel® Cilk™ Plus threading runtime Eclipse and sysroot support 	<ul style="list-style-type: none"> Boost system performance for IA-based embedded designs and achieve scalability benefits of multicore and forward scale to many-core Cross platform development and integration into Eclipse for ease-of-use
	Intel® Integrated Performance Primitives 8.1	<ul style="list-style-type: none"> A set of high-performance software building blocks for signal, media, and data processing 	<ul style="list-style-type: none"> Speed up development & performance with key software building blocks for signal, data, and media processing GCC* compatibility
		<ul style="list-style-type: none"> Leading provider of high-performance math functions for 	

Enhance power efficiency, system reliability, and boost performance with Intel® System Studio 2014

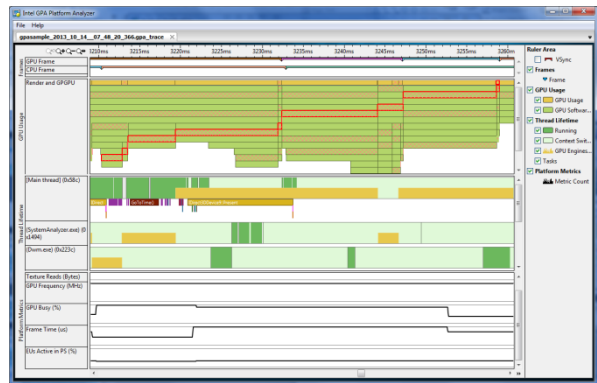
Optimization Workflow

Start Here:

System Analyzer:
Real-Time Problem
Overview



Platform Analyzer:
CPU/GPU
Interaction



VTune Amplifier:
CPU/GPU
Compute Analysis



System Analyzer

- High Level performance analysis tool
- Displays a variety of metrics in real-time
 - Similar to perfmon on windows
 - A number of metrics supported
- Allows one to do what-if experiments
 - One can determine whether the graphics app is CPU or GPU bound
 - Set the cpu frequency and see if FPS changes
 - Tells whether the app is cpu bound or not



System Analyzer Metrics

- CPU metrics
- GPU metrics
- OpenGL API metrics
- Power Metrics
- Memory Metrics
- Device I/O Metrics



Profiling Over wifi

- Set up the Android* Debug Bridge USB connection.
- Plug in the Android* OS device to your machine's USB port using the micro USB connector.
- Type "adb devices" to check that the Android* OS device is listed and no other devices are connected.
- Restart the Android* Debug Bridge (ADB) tool in TCP mode with the command "adb tcpip 5555".
- Disconnect the Android* OS device from your machine.
- Connect your machine and the Android* OS device to Wi-Fi network.
- On your Android* OS device open Wi-Fi connection settings and find the IP address assigned to this device.
- Execute the command "adb connect [android-device-ip]" to establish the ADB connection wirelessly via the TCP protocol.
- Open the Intel GPA System Analyzer on your machine.
- Find the Android* OS device with the name "[android-device-ip]:5555" and select it.
- Choose the application for profiling.



Preparing your application to be analyzable by GPA

- The Intel® GPA System Analyzer **Application List** screen displays two groups of applications:
 - Analyzable applications
 - Non-analyzable applications
- **Analyzable applications** have the proper permissions that allow Intel GPA to analyze these applications.
- For **Non-analyzable applications** list modify the application's permissions
 - In AndroidManifest.xml
 - Enable the Internet connection permission: `<uses-permission android:name="android.permission.INTERNET">`
 - Enable the debuggable permission: `<application android:debuggable="true">`

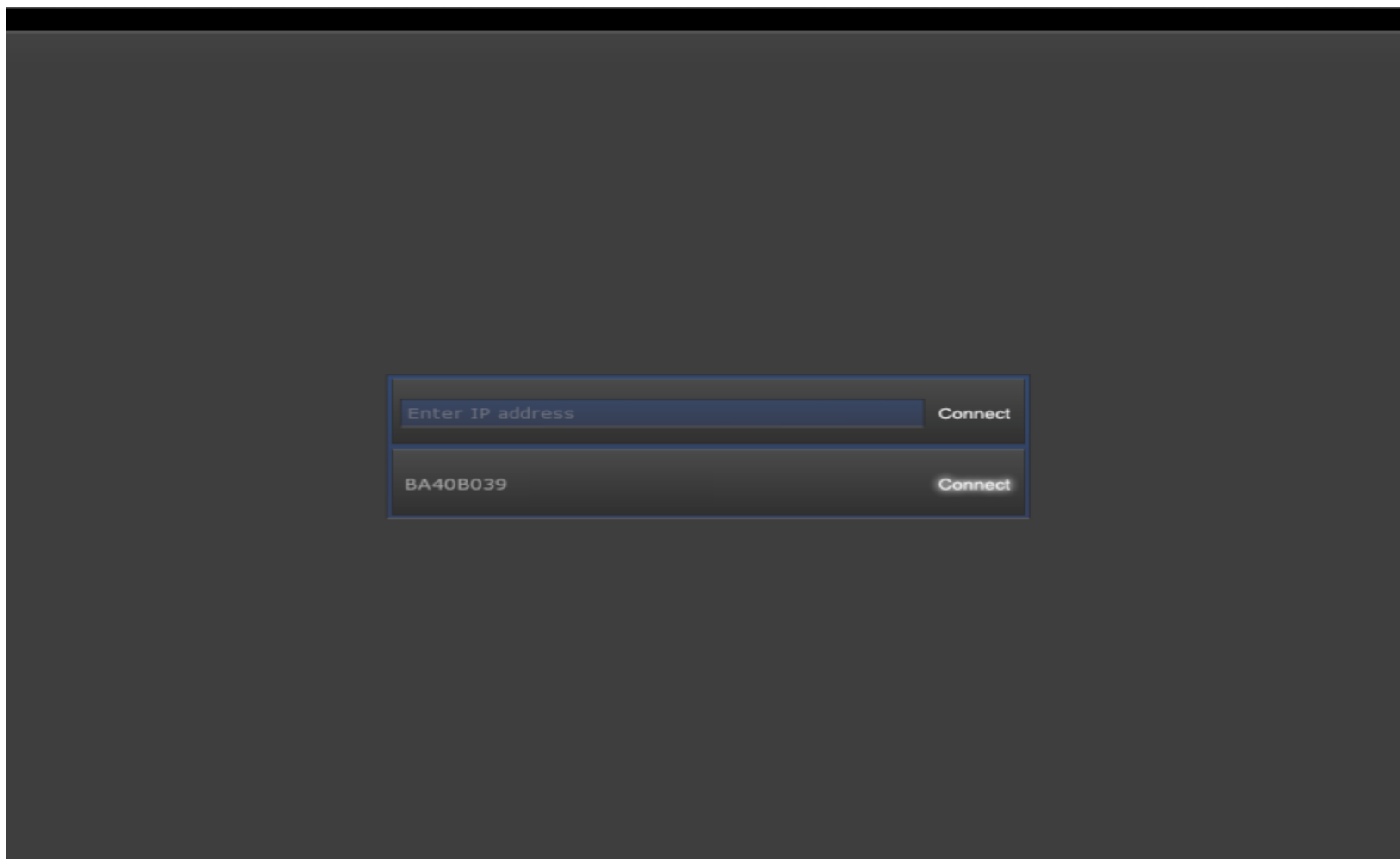


Launching GPA System Analyzer

- Connect the Intel® GPA System on the analysis system to the target system:
 - Launch the Intel GPA System Analyzer on the analysis system.
 - Connect the Android* OS device(s) to the analysis system using a USB cable.
 - Wait for up to 10 seconds while your client system is detecting the target device(s). Found devices appear in the dialog window. The list of the target devices refreshes every 5-6 seconds.
- Find the device you want to connect to and click **Connect**.
 - Required components are copied to the target device
 - A list of installed applications is shown on screen.

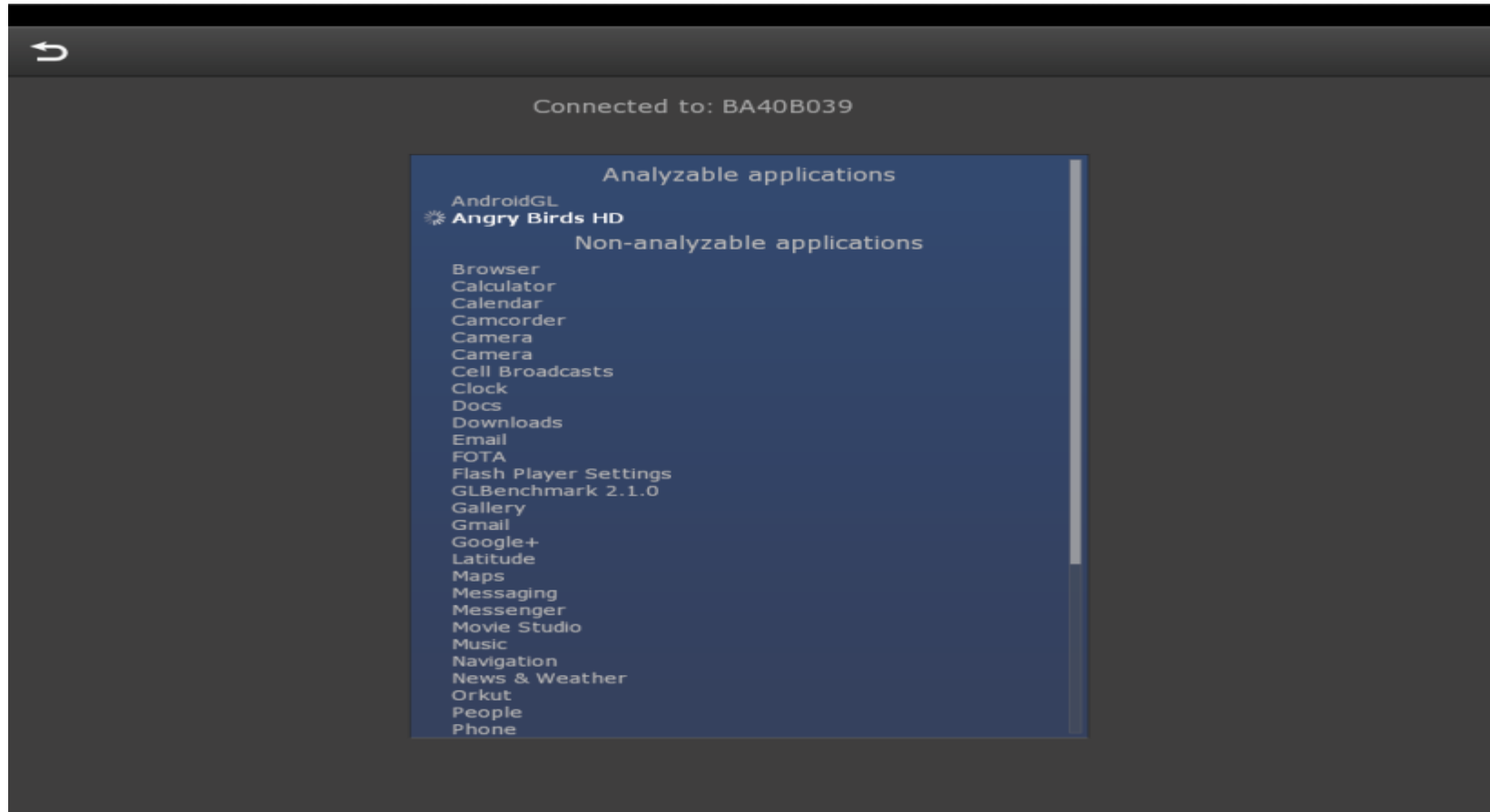


Launching System Analyzer



Launch System Analyzer

Pick the application you want to launch



Launching System Analyzer

- The application is launched and displays a message box "Waiting for Debugging".
 - wait for Intel GPA to attach to this application and start displaying metrics.
- To switch to a different application, click **Back**.
 - running application will be forced to close.
- To switch to a different target device, click **Back**.



System Analyzer



Custom Metrics

- Custom metrics can be defined in the user applications source code and updated by the application via OpenGL* ES extension API provided by Intel GPA.
- They can provide application specific metrics meaningful to a given application
 - Number of lights in a scene
 - Duration of post-processing pipeline stage
 - Application Markers
- Supported for Android* OS graphics applications using OpenGL* ES 1.0 or OpenGL* ES 2.0 API.
 - More info at Intel GPA Online Help



What-if experiments with System Analyzer

- Override modes provide a method for high-level performance analysis and visual debugging.
 - modify one or more of the render states of the graphics pipeline to show the effect of that phase on the rendering process,
- If using a certain override mode improves performance significantly, then that overridden mode might be a performance bottleneck
 - example, overrides can help isolate where in the rendering pipeline your bottlenecks are: try out **Texture 2x2** to see whether your textures are causing memory bandwidth issues (“thrashing”).



Override Modes





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 © 2013, Intel Corporation. All rights reserved. Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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