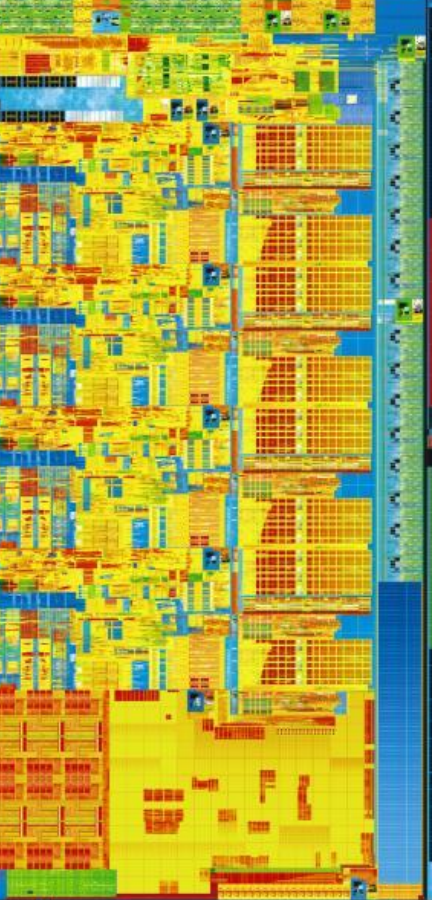


Software Development Tools targeting the entire software stack on Intel® Atom™ Processor Based Devices

An Overview of the
Intel® Embedded Software Development Tool Suite



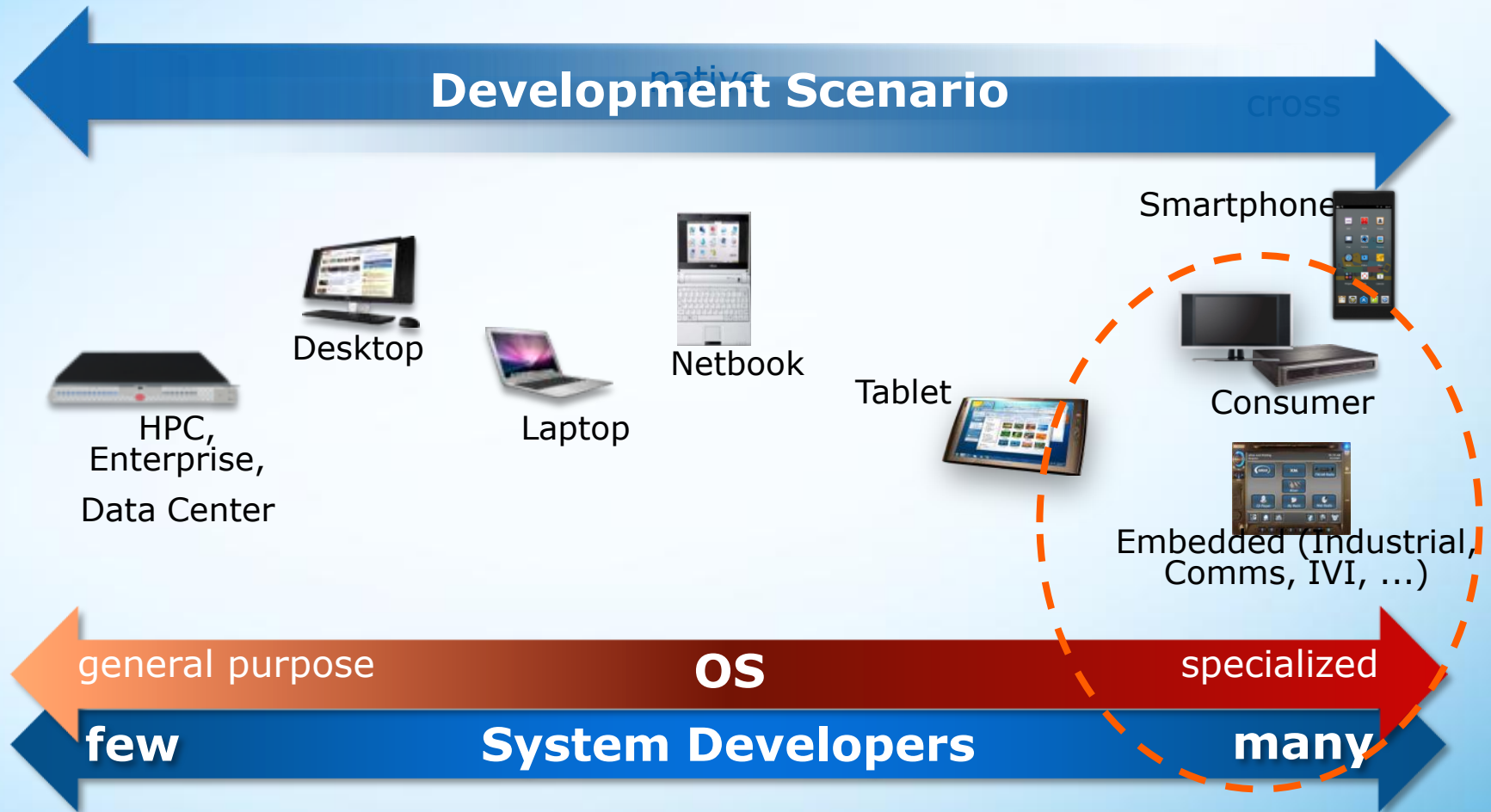
Software Development Tools targeting the entire software stack on Intel® Atom™ Processor Based Devices

An Overview of the
Intel® Embedded Software Development Tool Suite

Agenda

- **The Embedded Software Compute Continuum**
- **Embedded Software Development, Analysis & Debugging**
 - **Intel® C++ Compiler**
 - **Intel® Integrated Performance Primitives**
 - **Intel® VTune™ Amplifier XE**
 - **Intel® Debugger**
- **Summary & Conclusion**

Compute Continuum in Adjacent Markets



**A market that requires
specialized software development tools**

Embedded Development is Different

- Cross Development Support (host \neq target)
- Customizable build environment
- Custom target OS and RT scheduler support
- Custom device layer support
- Multi-platform support
- System level and Apps level debug
- Insight into platform architecture and devices
- Scriptability for automation and validation

Added complexity compared to non-embedded development

Addressing Diverse Needs

- Variety of Linux* base platforms
 - Open source / commercial / roll your own
- Variety of cross-development models
 - sysroot, jailroot/chroot, native
- Variety of host/target communication methods
 - tcp/ip, usb, JTAG

Flexibility in Development Model

=

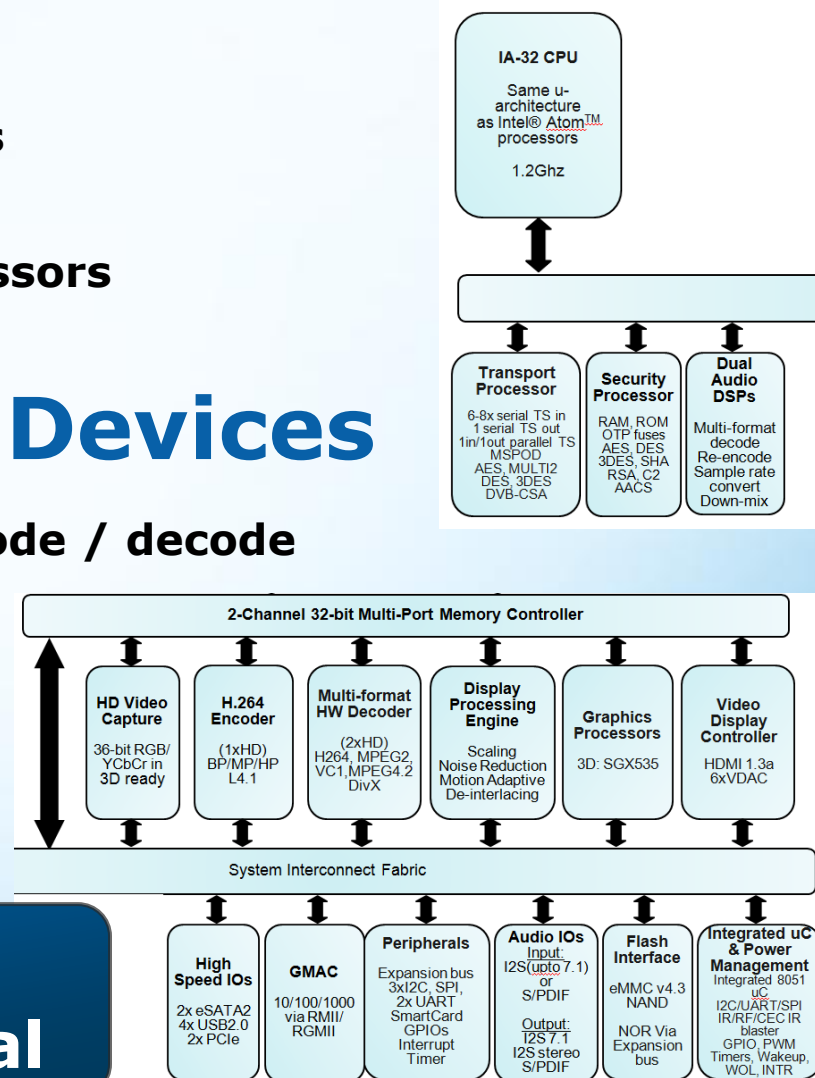
Flexibility in Targeted Platforms

SoC = Heterogeneous Multicore

- Several different architectures
- Several different specialized processors
- Common memory bus
- Data exchange between multiple processors

Orchestra of Devices

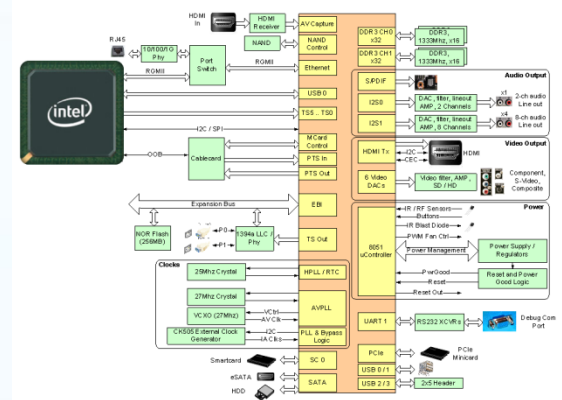
- Specialized micro-engines for data encode / decode
- Multitude of I/O devices
- Advanced power management
- Common memory & data bus system



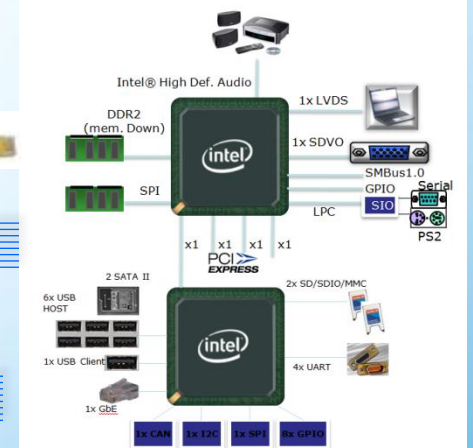
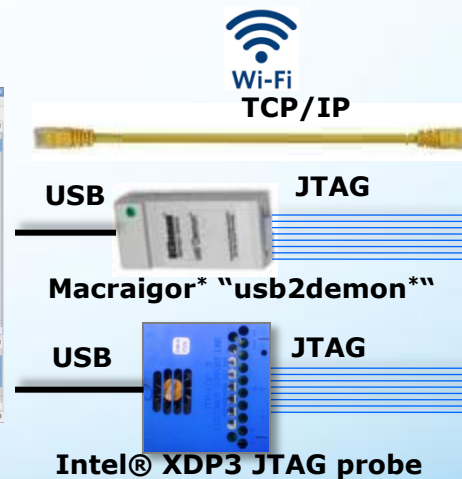
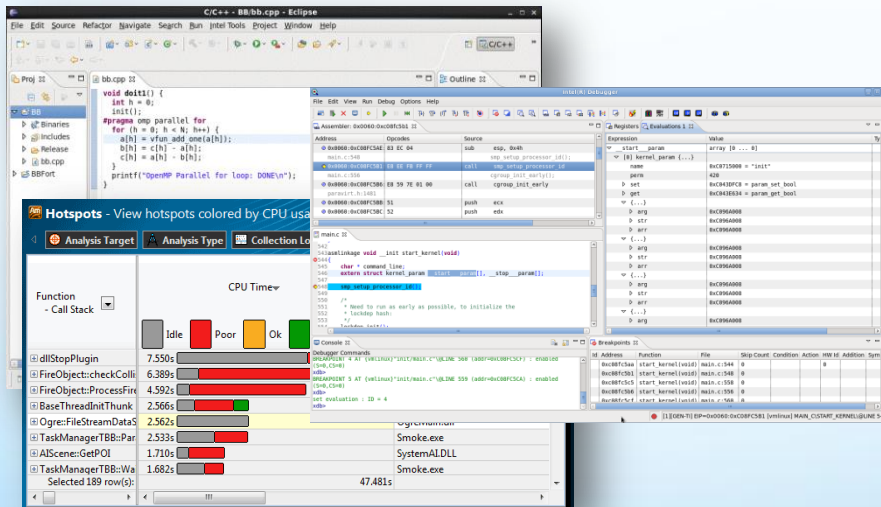
Access and visibility to all SoC components is essential

Development Across Devices

- Common tool set for Intel® Atom™ processor based platform designs
- Early availability of support and optimization
- Cross-development solutions for complex setups
- Full Awareness of all SoC devices



Intel Atom Processor CE4200



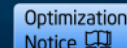
Intel Atom Processor E600

**A single flexible tool solution
for embedded development and validation**

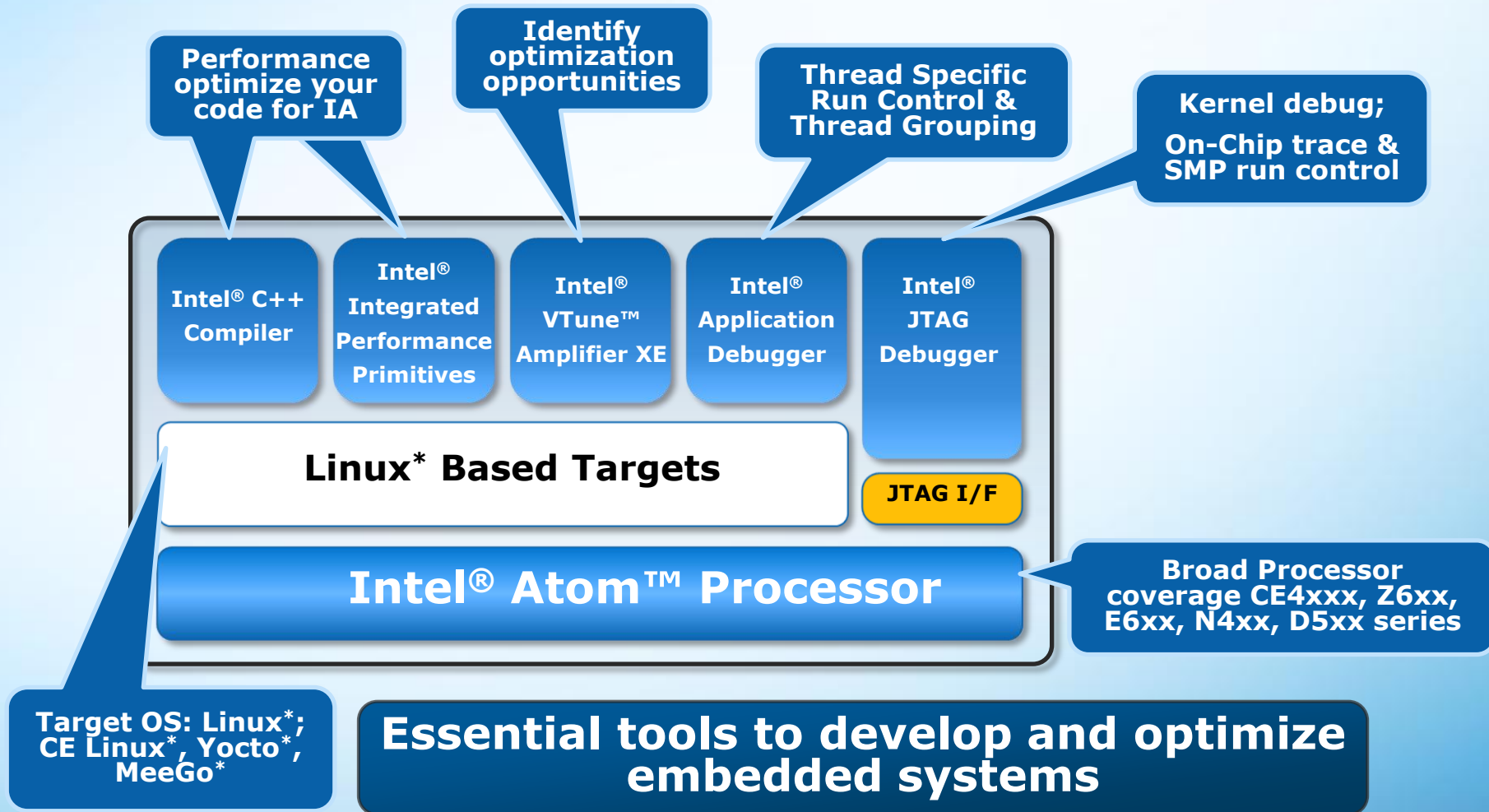


Software &
Developer Products Division

Copyright© 2011, Intel Corporation. All rights reserved.
*Other brands and names are the property of their respective owners.



Intel® Embedded Software Development Tool Suite 2.3 for Intel® Atom™ Processor



The Development Cycle

Intel® C++ Compiler

- SSSE3 Vectorization
- In-order scheduler
- Memory access optimization

Intel® Integrated Performance Primitives



- **Intel® Application Debugger**
- **Intel® JTAG Debugger**
- **Intel® Flash Memory Tool**

- **Intel® VTune™ Amplifier XE**
- **Sampling Collector for Intel® VTune™ Amplifier XE (SEP)**
- **Intel® Debuggers**

**Tool Suite for all Phases of Development
Design through Validation**

Agenda

- **The Embedded Software Compute Continuum**
- **Embedded Software Development, Analysis & Debugging**
 - **Intel® C++ Compiler**
 - **Intel® Integrated Performance Primitives**
 - **Intel® VTune™ Amplifier XE**
 - **Intel® Debugger**
- **Summary & Conclusion**

Intel® C++ Compiler and Intel® Atom™ Processor



- Optimization Switch `-xSSE3_ATOM`
 - In order scheduler
 - IDIV → DIVB expansion
 - Arithmetic operations feeding addresses turned into LEAs
 - All stack adjusts done using LEAs
 - Support for movbe instruction
 - Streaming SIMD Extensions 3 (SSE3) instruction support
- Compiler Based Vectorization and Automatic Processor Dispatch – `ax[?]`
 - Single executable optimized for Intel® Atom™ processors and generic code that runs on all Intel® Architecture-32 processors
 - For each target processor it uses: Processor-specific instructions, vectorization, low overhead, some increase in code size

Dedicated performance optimizations for Intel® Atom™ Processor

Build Support for Cross-Build Environments

Embedded cross-build environments for Linux* tend to have varying install locations for

- Preprocessor defines
- GNU tools paths and names
- GNU startup files, C++ includes/runtime
- Location of target system headers and libraries
- The list of default libraries

Intel® C++ Compiler supports

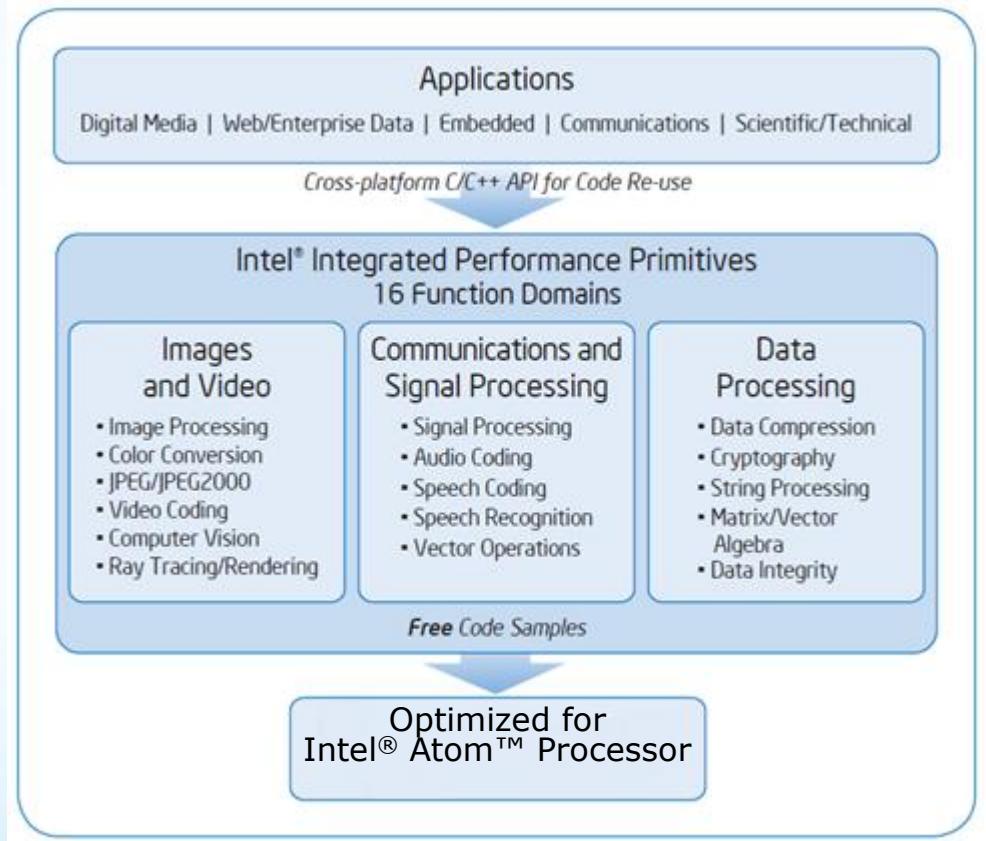
- --sysroot
- Chroot/jailroot installs
- Detailed build environment definition via `-platform=<name>` (where name is the name of a user editable environment file)
- Tested against Poky Linux*, MADDE*, CE Linux* SDK
- Yocto Linux* Application Development Toolkit Support

**Compiler is flexible in meeting
embedded build environment needs**

Intel® Integrated Performance Primitives




- Extensive library of highly optimized software functions for digital media and data-processing applications
 - Image & Video
 - Communication & Signal Processing
 - Data Processing
- Fully utilizing
 - AVX optimizations
 - SSE2, SSE3, SSE4.2 optimizations
 - Multi-core / Intel® Hyper-Threading technology
- Rapid application development
- Cross-platform compatibility & code re-use
- Outstanding performance



Highly optimized functions for digital media and data-processing applications

Event Count	Hardware Event Type	
L11 MISSES BY Package	RS_UOPS_DISPATCHED BY Package	UOPS_RETIRED_ANY BY Package
1,000	4,686,000,000	4,484,000,000
1,000	4,654,000,000	4,654,000,000
1,000	BR_INST_RETIRED_MISPRED	BR_INST_RETIRED_MISPRED
1,000	CPU_CLK_UNHALTED.CORE	CPU_CLK_UNHALTED.CORE
1,000	RS_UOPS_DISPATCHED	RS_UOPS_DISPATCHED
1,000	UOPS_RETIRED_ANY	UOPS_RETIRED_ANY
1,000	UOPS_RETIRED_MISPRED	UOPS_RETIRED_MISPRED
1,000	CPU_CLK_UNHALTED.CORE	CPU_CLK_UNHALTED.CORE
1,000	L11 MISSES	L11 MISSES
1,000	RS_UOPS_DISPATCHED	RS_UOPS_DISPATCHED
1,000	UOPS_RETIRED_ANY	UOPS_RETIRED_ANY
1,000	UOPS_RETIRED_MISPRED	UOPS_RETIRED_MISPRED
1,000	CPU_CLK_UNHALTED.CORE	CPU_CLK_UNHALTED.CORE



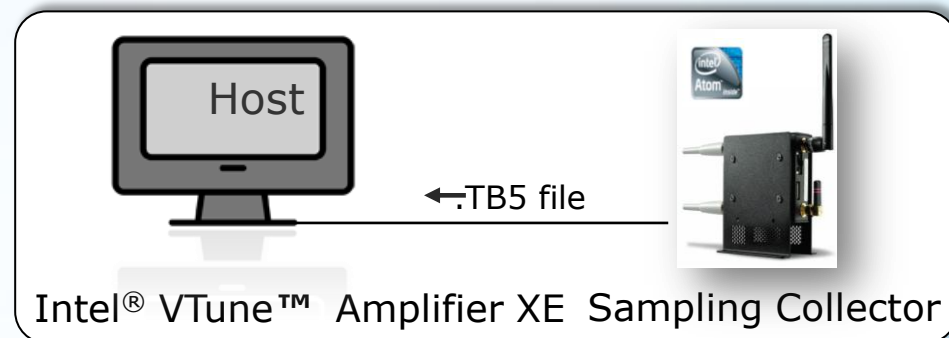
Optimization
Notice 

Intel® VTune™ Amplifier XE in Embedded



Usage Model

- Two components
 - Intel® VTune™ Amplifier XE on host
 - Sampling Collector on the target
- Collect data on target and analyze it on the host



Features

- Statistic Analysis
- Low overhead sampling
- No instrumentation required
- Monitor processor events like cache misses, etc.
- View results in source or assembly

Current Limitations

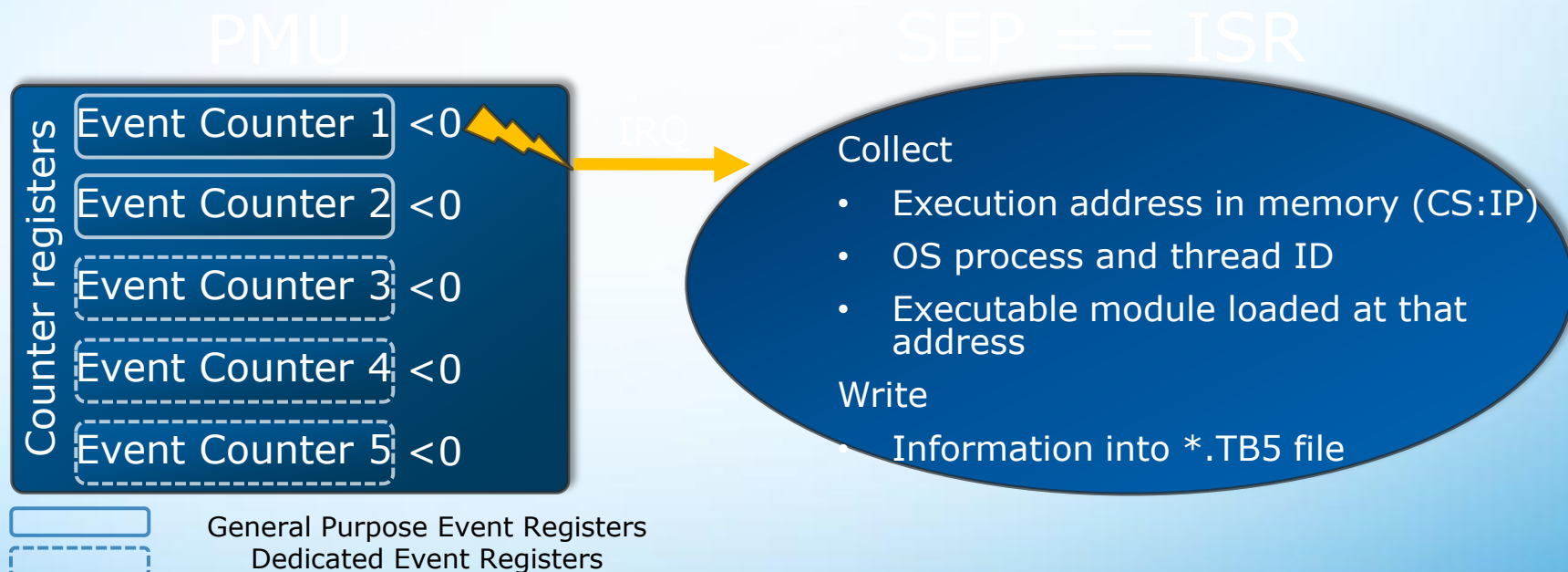
- Frame Analysis requires engineering patch
- No call trace or call stack available
- No Semaphore, Lock & Wait Analysis

Identify optimization opportunities in Modules, Functions or Routines

Sampling - How To Find Hotspots



- Pick an event to sample and configure PMU
 - Cache misses, branch mis-predictions, Dependency/pipeline stalls
- Start SEP sampling routine and application
- Performance Monitoring Unit (PMU) periodically interrupts the processor
 - Time based sampling
 - Event based sampling



- Numbers in counters define sampling rate

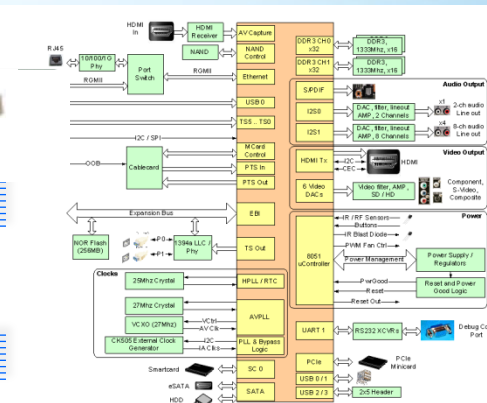
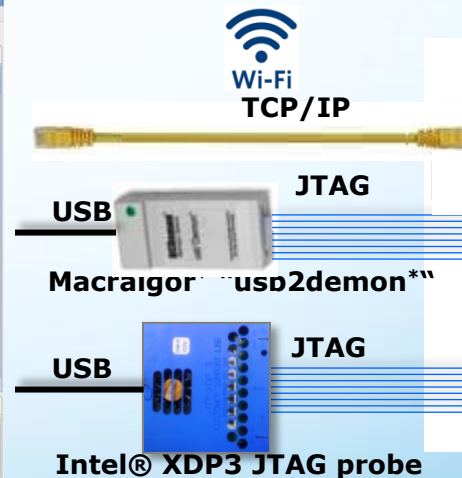
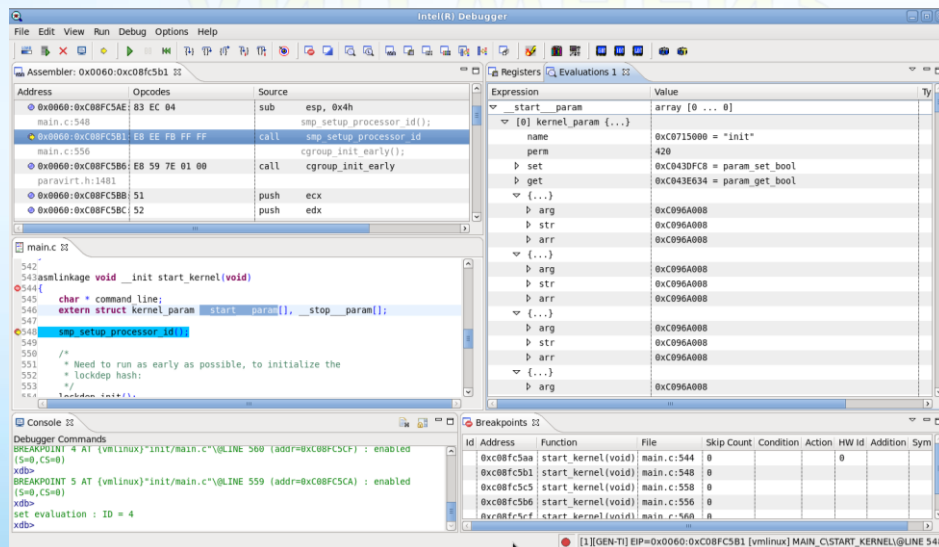
Debug and Validation



**WHICH CODE,
DID WHAT,
AND WHEN?**

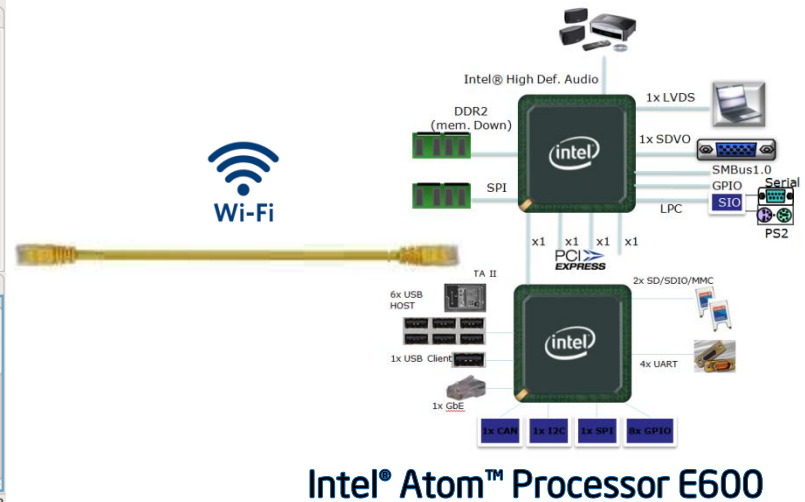
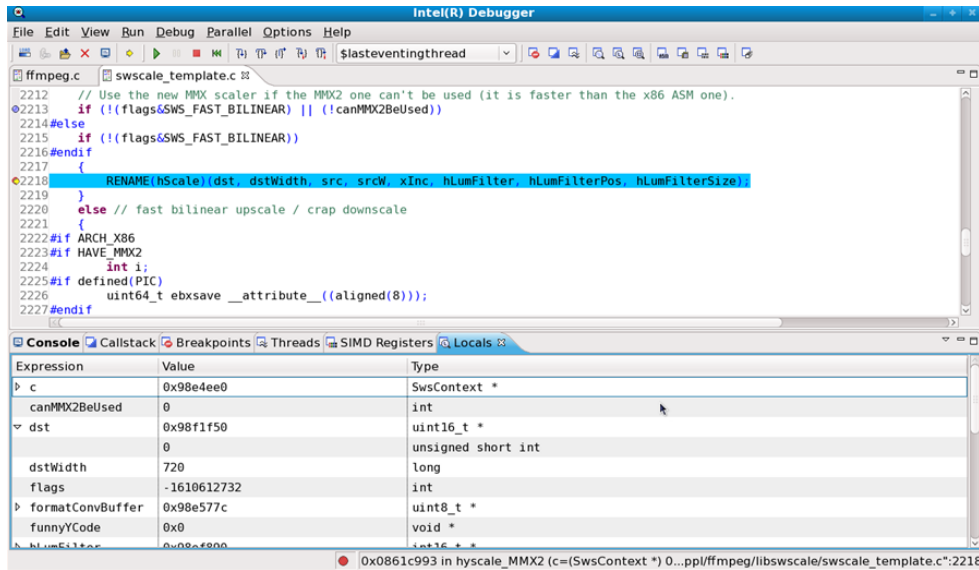
Kernel Mode and User Mode

- Platform stability and start-up
 - Firmware, boot loader
- Operating system, driver
 - Application



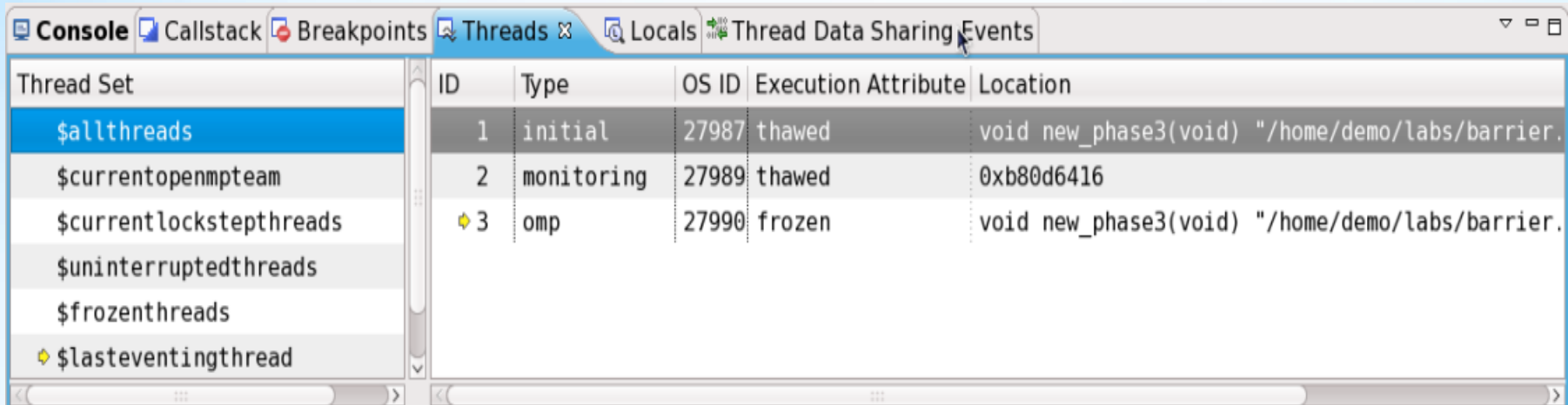
Intel® Debugger covers embedded debug needs

Intel® Debugger for Linux*



Cross-Debug Solution with advanced thread awareness

Control which thread to debug



In the Threads window you can select which thread you would like to debug. The context menu allow you to 'freeze' and 'thaw' individual threads.

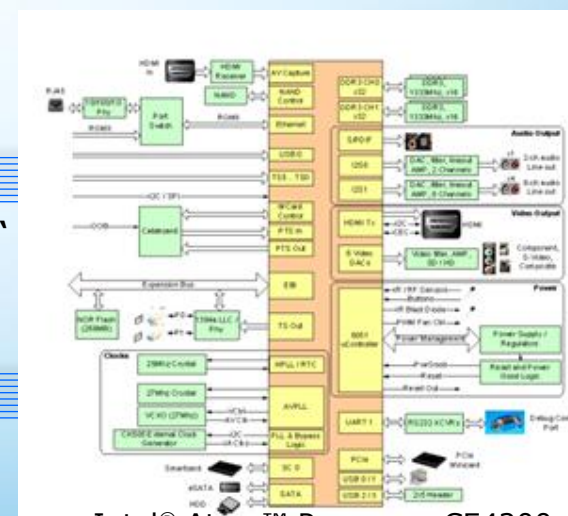
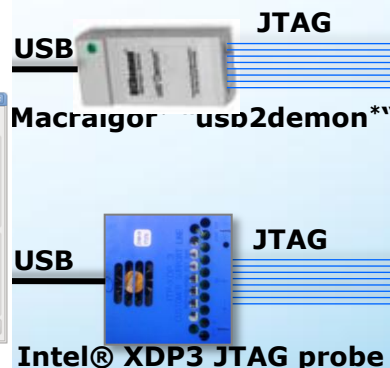
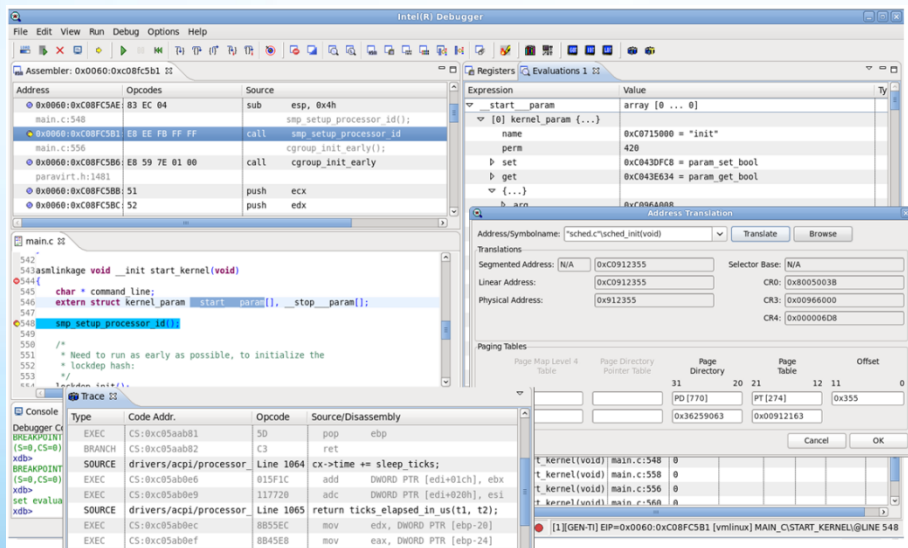
When you stop/hit a breakpoint all non-frozen threads will be stopped and when you continue all non-frozen threads will run.

Intel® JTAG Debugger – Overview



Features

- Bitfield editor supporting Intel® Atom™ processor based CPUs and SoCs
- Flashing and peripheral register support for Intel Atom processor CE4200
- Access to page translation and descriptor tables
- Dynamically loaded kernel module debug
- On-Chip trace support
- SMP run control support
- Host OS: Linux*; Target OS: Linux awareness
- Platforms: Intel Atom processor Z5xx, Z6xx, E6xx, Nxxx, and CE4xxx

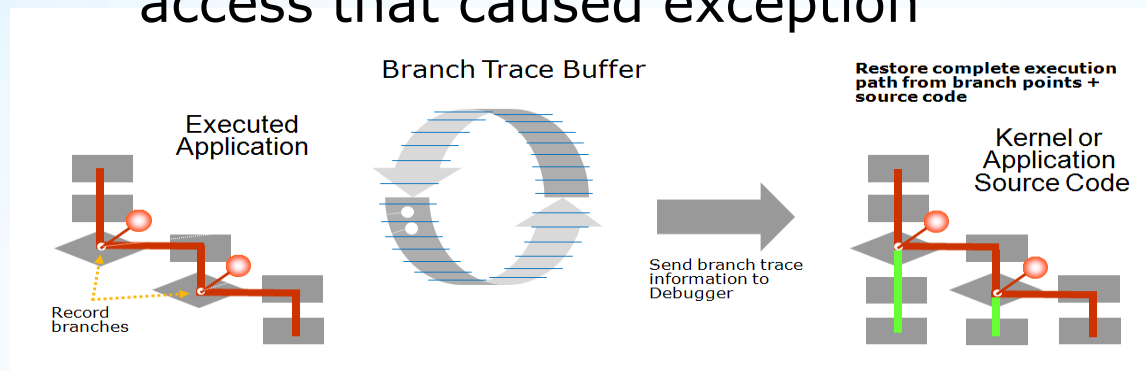


Intel® Atom™ Processor CE4200

Intel® JTAG Debugger – Instruction Trace Support



1. Kernel or device driver causes exception
2. Read **execution trace** at assembler level to identify access that caused exception



Type	Code Addr.	Opcode	Source/Disassembly
SOURCE	arch/i386/kernel/irq.c:do_IRQ(class *) Line 113		}
BRANCH	CS:0xc010594e		
BRANCH	CS:0xc01037fa		
EXEC	CS:0xc0102d40		
EXEC	CS:0xc0102d45		
EXEC	CS:0xc0102d47		
EXEC	CS:0xc0102d4b		
EXEC	CS:0xc0102d4f		
BRANCH	CS:0xc0102d5d		

Type	Code Addr.	Opcode	Source/Disassembly
SOURCE	kernel/profile.c:profile_hit(int,void*)	Line 383	
BRANCH	CS:0xc011a00e	C3	ret
SOURCE	arch/i386/kernel/apic.c:smp_apic_timer_interrupt(class *) Line 1199		
EXEC	CS:0xc010f1c9	83C408	add esp, 0x8h
SOURCE	arch/i386/kernel/apic.c:smp_apic_timer_interrupt(class *) Line 1198		
BRANCH	CS:0xc010f1cc	E98FF20000	jmp irq_exit
SOURCE	include/asm/thread_info.h:irq_exit(void)	Line 92	
EXEC	CS:0xc011e460	B800F0FFFF	mov eax, -0x1000
EXEC	CS:0xc011e465	21E0	and eax, esp
SOURCE	kernel/softirq.c:irq_exit(void)	Line 167	
EXEC	CS:0xc011e467	8168140000100	sub DWORD PTR [eax+irq_exit+07h], 0x10000h
SOURCE	kernel/softirq.c:irq_exit(void)	Line 168	

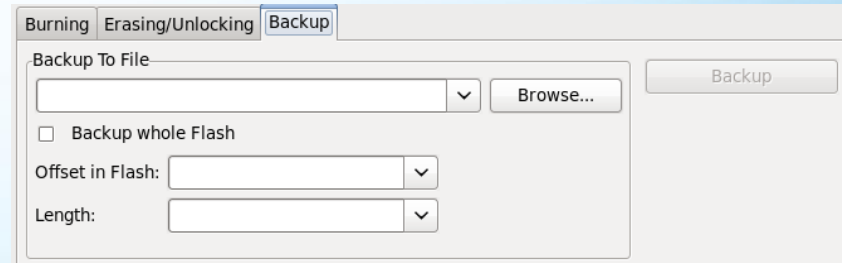
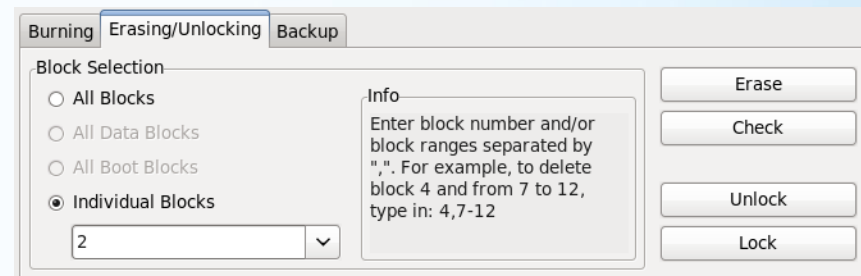
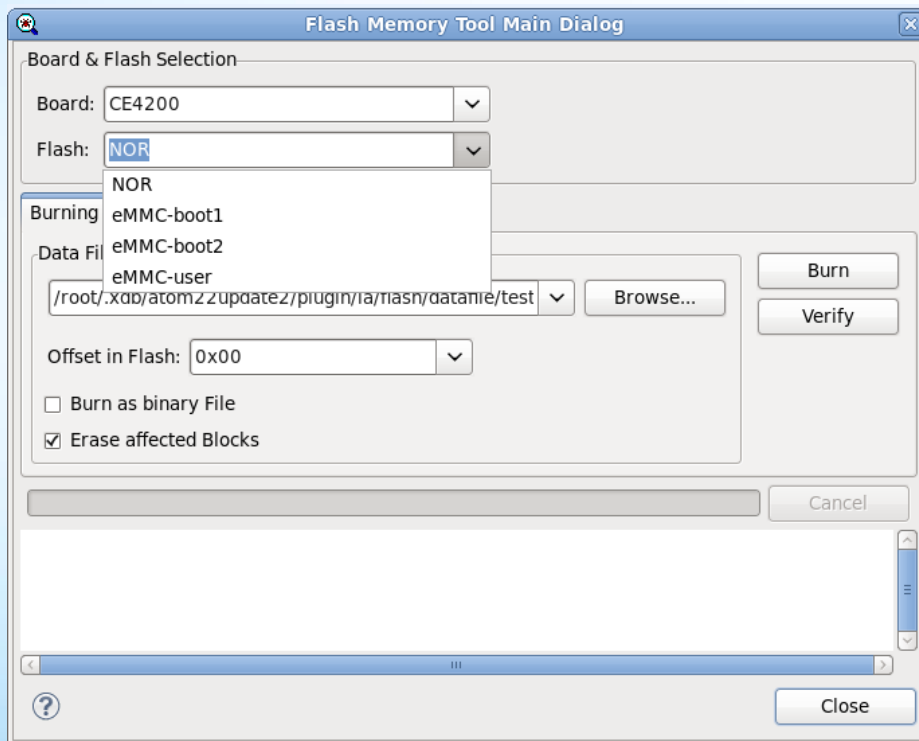
**Identify root-cause
of system runtime
failures**

Flashing Target Images and Bootloader



Intel® C++ JTAG Debugger Flash Memory Tool

- Flash binary and hex files
- Erase/unlock/lock blocks
- Backup flash contents into binary file on host
- Ideal for BIOS update



***Update software stack
components
from within the debugger***

Intel® JTAG Debugger – CPU & Chipset-Specific Register Access



Register Windows

Register	Value	Description
TC_CFG_D31IP	0x00000000	Device 31 interrupt
TC_CFG_D27IP	0x00000000	Device 27 interrupt
TC_CFG_D2IP	0x00000000	Device 2 interrupt
TC_CFG_D26IP	0x01000000	Device 26 interrupt
TC_CFG_D25IP	0x01000000	Device 25 interrupt
TC_CFG_D24IP	0x01000000	Device 25 interrupt
TC_CFG_D23IP	0x01000000	Device 23 interrupt
TC_CFG_D03IP	0x00000000	Device 3 interrupt
TC_CFG_D31IR	0x0032	Device 31 interrupt
TC_CFG_D27IR	0x1032	Device 27 interrupt
TC_CFG_D26IR	0x1032	Device 26 interrupt
TC_CFG_D25IR	0x1032	Device 25 interrupt
TC_CFG_D24IR	0x1032	Device 24 interrupt

Bitfield Editor: TC_CFG_D31IR

Original Value: 0x0032
Register Value: 0x0032
Group Value: 0x02
IAPR 0:3

Register Layout:

ICPR — 15 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 IBPR

Interrupt A Pin Route
Description:
3140h Device 31 interrupt route

Bits 15 to 12:
Indicates which physical pin INTD# uses for device 31.

Bits 11 to 8:
Indicates which physical pin INTC# uses for device 31.

Close Restore Set

Console
Debugger Commands
xdbg> run
xdbg_R>
program stopped: BREAKPOINT ID=2 at "MAIN_C\START_KERNEL \@LINE 700" [task=GEN-TI (0)]
xdbg_R>

Breakpoints
Id Address Fun
0xc08fc sta
0x00100
0xc08fc start_k mai 0

[1][GEN-TI] EIP=0x0060:0xc0476CAA [vmli...TRACE_POWER_C\TRACE_POWER_END\@LINE 118

- Graphical representation
- Runtime manipulation
- Each bit fully documented

Full SoC Chipset Awareness

Page Translation Table – Insight into Memory Configuration



- Instant and simple resolution and translation between physical and virtual address space
- Identify why memory access failed

Address Translation

Address/Symbolname:

Translations

Segmented Address: Selector Base:

Linear Address: CR0:

Physical Address: CR3:

CR4:

Paging Tables

Page Map Level 4 Table	Page Directory Pointer Table	Page Directory	Page Table	Offset
Index:		31 20 21 12 11		
Value:		PD [769]	PT [9]	0x472
		0x3624C063	0x00409163	

Modify Page-Table Attributes

Original Value: 0x0163

Register Value:

Group Value:

P 0:0

Register Layout:

Page is present in memory

Description:

Page-Table Attributes

Present Bit (P Bit)

The Present Bit indicates whether the page frame address in a page table entry maps to a page in physical memory. If the bit is set, the page is in memory.

Paging

Index	Value	Page Map Level 4 Table	Page Directory Pointer Table	Page Directory	Page Table	Offset
PD [768]	0x00000000			31	20 21	12 11
PD [769]	0x00000000			PD [769]	PT [9]	0x472
PD [770]	0x00000000			0x3624C063	0x00409163	
PD [771]	0x00000000					
PD [772]	0x00000000					
PD [773]	0x00000000					
PD [774]	0x00000000					
PD [775]	0xC1C00000 to 0xC1FFFFFF			0x01C00000		P=0 A=1 D=1 PS=1(4MB) G=1 PAT=0
PD [776]	0xC2000000 to 0xC23FFFFFF			0x02000000		D=0 A=1 D=1 PS=1(4MB) G=1 PAT=0
PD [777]	0xC2400000 to 0xC27FFFFFF			0x02400000		D=0 A=1 D=1 PS=1(4MB) G=1 PAT=0
PD [778]	0xC2800000 to 0xC2BFFFFFF			0x02800000		P=1 R/W=1 U/S=0 PWT=0 PCD=0 A=1 D=1 PS=1(4MB) G=1 PAT=0
PD [779]	0xC2C00000 to 0xC2FFFFFF			0x02C00000		P=1 R/W=1 U/S=0 PWT=0 PCD=0 A=1 D=1 PS=1(4MB) G=1 PAT=0
PD [780]	0xC3000000 to 0xC33FFFFFF			0x03000000		P=1 R/W=1 U/S=0 PWT=0 PCD=0 A=1 D=1 PS=1(4MB) G=1 PAT=0

Take control of memory configuration issues and memory leaks

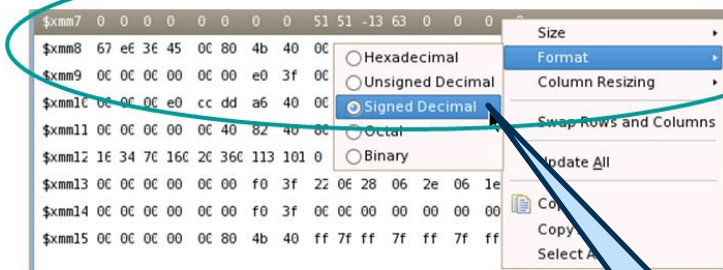
Vector Register View and Evaluation



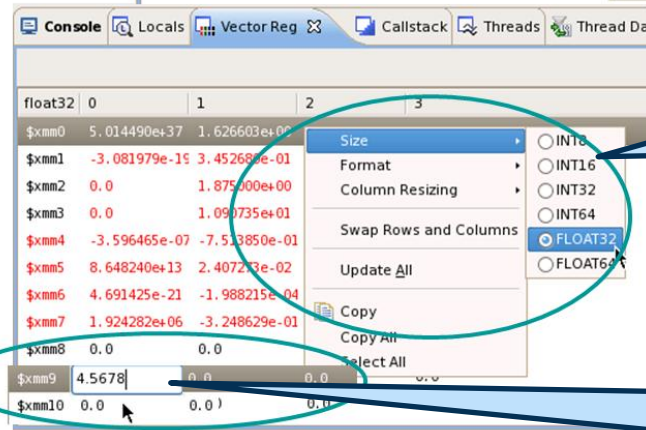
- View and modify registers in the same value grouping
- Drag & drop source code into vector evaluation window
- View vector register usage
- Modify variable value directly in vector register SSE x.0, SSSE3, AVX

Select and drag & drop.

The screenshot shows two source code files, video.cpp and sphere.cpp, in a code editor. Line 110 in sphere.cpp, which contains the code `if (disc <= 0.0) return;`, is highlighted in blue. A green arrow points from this line to the Vector Register View window.



Reformat Variable Size & Type



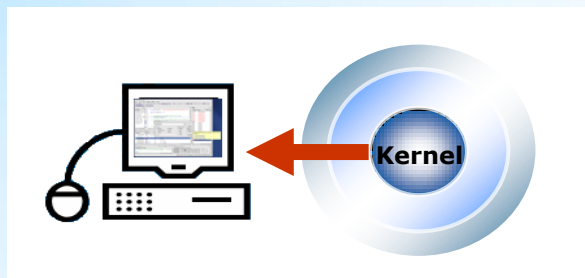
Modify Values on the Fly

Vector Register View of Source Code

Track and debug SSE register usage

Linux* OS Awareness – System Debug

Debug dynamically loaded device drivers without instrumentation



sum1.c

```
66 return count;
67 }
68
69 static int _init sum_init(void) {
70     struct proc_dir_entry *proc_arith;
71     struct proc_dir_entry *proc_arith_sum;
72     int y = foobar(10);
73     y = y;
74     proc_arith = proc_mkdir("arith", 0);
75     if (!proc_arith) {
76         printk(KERN_ERR "cannot create /proc/arith\n");
    }
```

Kernel Threads

Name	pID	State	Flags
swapper	0	runnable	FREE_PAGES
init	1	stopped	SUPERPRIV
kthreadd	2	stopped	FORKNOEXEC FREE_PAGES
ksoftirqd/0	3	stopped	FORKNOEXEC FREE_PAGES

Assembler: 0x0060:0xc01032b1

Kernel Modules

Module-Name	Stop at Init	Stop at Exit	Debug ID	Kernel Object File	Size	Init-Method	Cleanup-Method	Arguments
sum1	YES	YES	1	/home/demo/labs/exam	0xd88	0xf88fe000	0xf88bb0d0	
soundcore	no	no	N/A	file not found	0x2144	N/A	0xf8863538	
snd	no	no	N/A	file not found	0xd734	N/A	0xf885206c	
iusbc	no	no	N/A	file not found	0x6f08	N/A	0xf886a7cd	
pci_hotplug	no	no	N/A	file not found	0x76c0	N/A	0xf88d3a78	

Module Settings

Module Name: own_kernel_mod

☒ Stop at Init ☒ Stop at Exit

Help OK Cancel

Console

Debugger Commands

Source directories deleted

Source path substitution rules deleted

xdb>

LOAD : "/home/demo/labs/vmlinux"

- Monitor kernel modules and system threads
- Access status information
- Debugging of Linux* memory images

In-depth device driver and OS layer views

Summary



- Intel® Embedded Software Development Tool Suite for flexible support of many embedded development scenarios
- Develop and optimize your software stack with Intel® Embedded Software Development Tool Suite
 - Intel® C++ Compiler
 - Intel® Integrated Performance Primitives
 - Intel® VTune™ Amplifier XE
 - Intel® Application and System Debugger
- Full system debug support from board design through platform and software stack optimization and validation
- Extensive support for Intel® Atom™ processor family

www.intel.com/software/products/atomtools

Intel® Premier Support

- Intel® Premier Support includes:

- Technical support direct from the experts at Intel
- Online access to Intel Premier Support Website
- Issue submission and tracking
- Product updates and related downloads
- FAQ's, user forums, and proactive notices

<http://premier.intel.com>

**"Registering for support was easy,
and we value the security of
knowing that Intel is there to help,
even though we haven't need it so
far."**

*Rob Hoffmann
Director of Marketing
NewTek, Inc.*

- Primary support for products below:
 - Intel® Parallel Studio 2011
 - Intel® Parallel Studio XE 2011
 - Intel® C++ Studio XE 2011
 - Intel® Cluster Studio XE 2011
 - Intel® Embedded Software Development Tool Suite for Intel® Atom™ Processor
 - Intel® Composer XE 2011
 - Intel® Inspector XE 2011
 - Intel® VTune™ Amplifier XE 2011
 - Intel® Integrated Performance Primitives
 - Intel® Math Kernel Library
 - Intel® MPI library
 - Intel® Threading Building Blocks
 - Intel® Cilk Plus
 - Intel® Graphics Performance Analyzer
 - Intel® Trace Analyzer and Collector

Take Aways & Next Steps



Software Development Tools targeting the entire software stack on Intel® Atom™ Processor Based Devices

Product details and download at

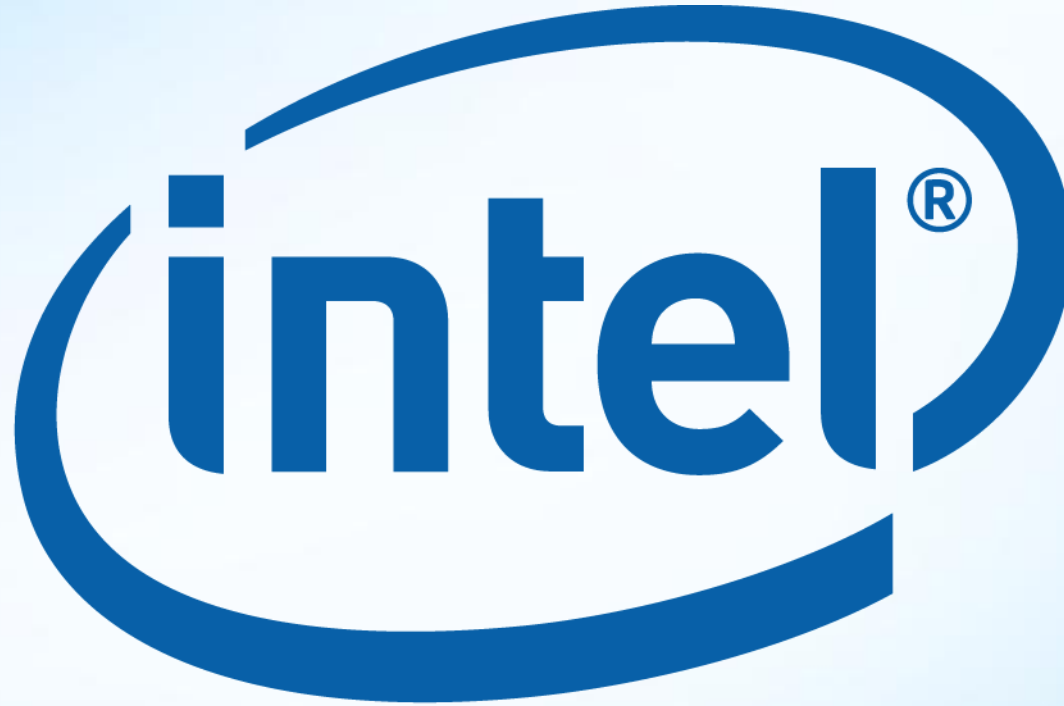
<http://software.intel.com/en-us/articles/intel-tools-for-intel-atom-processors/>

Questions:

EmbeddedDevTools@intel.com

Next Steps: Try the products ...

**Download a free 30-day evaluation at:
www.intel.com/software/products**



Software

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.

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, reference www.intel.com/software/products.

Copyright © , Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Core, VTune, and Cilk are trademarks of Intel Corporation in the U.S. and other countries. *Other names and brands may be claimed as the property of others.

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