Intel® Atom™ Processor E6xx Series Software Introduction

FOR 2012 INTEL CUP ESDC
Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

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

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

64-bit computing on Intel architecture requires a computer system with a processor, chipset, BIOS, operating system, device drivers and applications enabled for Intel® 64 architecture. Performance will vary depending on your hardware and software configurations. Consult with your system vendor for more information.

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM) and, for some uses, certain computer system software enabled for it. Functionality, performance or other benefits will vary depending on hardware and software configurations and may require a BIOS update. Software applications may not be compatible with all operating systems. Please check with your application vendor.

Intel® Hyper-Threading Technology requires a computer system with a processor supporting HT Technology and an HT Technology-enabled chipset, BIOS and operating system. Performance will vary depending on the specific hardware and software you use. For more information including details on which processors support HT Technology, see here. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Copies of documents which have an order number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel’s website at http://www.intel.com. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copyright © 2008, Intel Corporation. All rights reserved.
Ensure completeness of software platform ingredients for ISG’s product lines
Firmware Solutions
Traditional BIOS vs. Boot Loader

**BIOS**
Open Box Designs
(Requires Flexibility)

- Standard OS compatibility
- Feature richness
- Open to many use cases
- Multiple boot paths
- Extra services and support

**Boot Loader**
Closed Box Designs
(Static Hardware Configurations)

- Custom OS & applications
- Basic Intel® architecture initialization
- Quick and small
- Single use case
- Limited boot options
- No frills
- Royalty free
- No hand-holding
Spectrum of System Initialization Firmware

- Memory
- Display
- ROM Update
- Power Management
- Legacy Compatibility
- Network
- OS Services
- System Management
- Storage
- Security
- Dynamic Setup
- Virtualization
- Reset
- Bus
- Simple I/O
- Peripheral Drivers
- Intel® BLDK Provides Flexibility to Scale System Initialization for Embedded Systems

Intel® BLDK:
- BIOS
- RISC Init
- Intel® Boot Loader Development Kit (Intel® BLDK)
UEFI Platform Initialization Overview

- UEFI specifies how firmware boots OS loader
- UEFI’s Platform Initialization (PI) 1.2 Architecture specifies how Driver Execution Environment (DXE) Drivers and Pre-EFI Initialization (PEI) Modules (PEIMs) initialize Si and the platform
- Intel® UDK2010 is an implementation of PI and UEFI specifications

Diagram:
- UEFI Specification
  - Platform Drivers
  - Silicon Component Modules
  - Hardware
- PEI/DXE PI Foundation
- Modular components
- Human User
  - GUI
  - Application
  - Libraries
  - Drivers
  - Network
  - OS
  - Firmware
  - Hardware

Full system stack (user -> hardware)
Intel® BLDK Codebase Overview

- Intel® BLDK is a reference implementation for Intel Customer Reference Boards (CRBs)

- Intel BLDK codebase
  - Distributed as a combination of source files and binary modules
  - Modular codebase allows for reuse of source code for different platforms
  - Compatible with latest UEFI standards
  - CRB reference implementations are publicly available

Intel® BLDK for Intel® Atom™ Processor E6xx Series with Intel® Platform Controller Hub EG20T is available for download
The primary purpose of system firmware is to initialize the platform and boot to the UEFI shell or an operating system.
# Platform Debug Methodologies

<table>
<thead>
<tr>
<th>Debug Method</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port 80h</td>
<td>• Simple</td>
<td>• Limited information</td>
</tr>
<tr>
<td></td>
<td>• Low overhead</td>
<td>• No execution tracing / flow control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• May require several build iterations to isolate failure</td>
</tr>
<tr>
<td>Serial Out</td>
<td>• Simple</td>
<td>• Additional hardware and initialization</td>
</tr>
<tr>
<td></td>
<td>• Low overhead</td>
<td>• No execution tracing / flow control</td>
</tr>
<tr>
<td></td>
<td>• More information than Port 80h</td>
<td></td>
</tr>
<tr>
<td>Software Debugger</td>
<td>• Freely available</td>
<td>• Debug agent on target</td>
</tr>
<tr>
<td></td>
<td>• Source level debug</td>
<td>• Cannot debug all flows</td>
</tr>
<tr>
<td></td>
<td>• Execution tracing / flow control</td>
<td></td>
</tr>
<tr>
<td>Hardware Debugger</td>
<td>• Source level debug</td>
<td>• Requires purchase of JTAG debugger</td>
</tr>
<tr>
<td></td>
<td>• Ability to step through code</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Greater visibility to HW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Ability to debug complex execution paths (CPU init, SMM)</td>
<td></td>
</tr>
</tbody>
</table>
Operating System Support
Intel® Atom™ Processor E6xx Series OS Support

• Microsoft*
  – Windows Embedded Standard 7*

• Linux*
  – Yocto
Why Yocto?

- Wide base of Developers
- Ease of use
- Feeding Developer traits

Innovations:
- Mainstream Linux
- Ingredients Proliferation
- Consistent Delivery mechanism

Website: http://www.yoctoproject.org
Video and Graphics
Different Graphics Drivers – Different Markets

- **Intel® Graphics Media Adapter (GMA) driver** is the best known Intel graphics driver
  - For Desktop and Laptop applications
  - Sometimes used in embedded for PC-like applications

- **Intel® Embedded Graphics Drivers (IEGD)**
  - All integrated graphics embedded chipsets/processors
  - Embedded features
  - Long Life support (>5 years)

- **Intel® Embedded Media and Graphics Drivers (EMGD)**
  - Embedded Intel® Atom processor focused
  - Long Life support (>5 years)
Windows* Miniport / Display Driver Arch

D3D/DDraw Application/Game

D3D/DDraw Runtime

Win32 (GDI32)

I/O Manager

DDI Calls

Display Driver

D3D DDI

DDRAW DDI

GDI DDI

Video Port

Video Miniport

Vendor supplied

OS System Supplied Service

User Mode

Kernel Mode

DRIVER
The port driver is the graphics driver module that enables display output devices.

All supported SDVO transmitters will be controlled through port driver.
Customizing – Configuration Editor (CED) Tool

- CED is the Configuration EDitor for embedded graphics drivers.
- It provides a straight-forward GUI interface for manipulating all the driver options.
- It creates the appropriate files to install the driver for all of the build options (Windows, Linux, VBIOS, EFI, etc.)
Power and Performance Optimization
## Performance Optimization

### Multicore
- Use multi-core software development tools
- Use multi-core enabled performance libraries
- Multithread for parallelism
- Put threads that share data on cores that share cache

### Tools
- Intel® Software Development Products
- Analysis tools
- Threading Tools and Libraries
- Debuggers

### Compiler Performance Features
- Use an Intel® Atom™ processor targeting option (-xSSE3_ATOM)
- Use automatic vectorization (-vec)
- Use the -O3 option for aggressive loop and memory optimization
- Use interprocedural optimization (IPO)
- Use profile guided optimization (PGO)

**Example:**
- `-O3 -ipo -no-prec-div -xSSE3_Atom -prof_gen -prof_use`
Processor Power States

- **P0** - CPU active at highest frequency (HFM)
- **Pn** - CPU active at lowest frequency (LFM)
- **C0** - CPU active (In any P-state)
- **C1** - Core clock is Off
- **C3/C4** - Reduced Voltage, Partial L2 cache flush
- **C6** - Core Off, L2 cache flush, state saved to SRAM

The deeper the sleep, the longer it takes for the processor to wake up.
# Software Recommendations for Power Optimization

<table>
<thead>
<tr>
<th>Power Guidelines</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hurry up and get idle</td>
<td>Reduce interrupts</td>
</tr>
<tr>
<td>Improve cache locality</td>
<td>Use timers effectively</td>
</tr>
<tr>
<td>Manage memory efficiently</td>
<td>Be power-aware and power-smart</td>
</tr>
<tr>
<td>Avoid polling and busy waits</td>
<td>Use multithreading</td>
</tr>
</tbody>
</table>
Thread Optimization for Power Efficiency

Stagger thread start - execute in parallel - sleep longer
Migrating to Intel® Architecture
Migration Design Guide

Part one
1. Port the code to the target operating system
2. Execute the code correctly on one Intel® architecture core
3. Optimize the code for performance on one Intel architecture core

Part two
4. Apply multi-core software design updates
5. Optimize the software design for multi-core Intel architecture

Steps 4 and 5 are optional
Endianness and Execution Difference

Example foo.c

```c
#include <stdio.h>
int a = 0x12345678;
int main()
{
    char *ap = (char *) &a;
    printf("%2x %x\n", *ap, a);
    return 0;
}
```

Different results on BE and LE machines!

Endian neutral code is portable
## Architecture Differences PowerPC* vs. Intel® Architecture

<table>
<thead>
<tr>
<th>Instructions</th>
<th>PowerPC* and Intel® architecture instructions are very different. For some instructions there is no one to one (PowerPC to Intel architecture) equivalent. Refer to <a href="https://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-software-developer-manuals.html">Intel® 64 and IA-32 Architectures Software Developer Manuals</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment</td>
<td>PowerPC instructions are all 4 bytes in size and must be aligned on 4 byte boundaries. Intel architecture instructions vary in size and therefore do not require alignment. On PowerPC a bool is 4 bytes. On Intel architecture, a bool is 1 byte. Make the code portable by changing the PowerPC boolean data to an unsigned 32-bit integer.</td>
</tr>
<tr>
<td>Vector Oriented Instructions</td>
<td>PowerPC uses Altivec* instructions. Intel architecture uses Intel® Streaming SIMD Extensions (Intel® SSE)</td>
</tr>
<tr>
<td>Divide-by-zero</td>
<td>For Integer divide-by-zero, PowerPC simply returns zero. On Intel architecture, executing this operation is fatal.</td>
</tr>
<tr>
<td>Calling Conventions Specified by ABI</td>
<td>Arguments are passed in registers for PowerPC. For Intel architecture, arguments are passed on the stack. Intel architecture has fewer registers than PowerPC and therefore local variables may be stored on the stack as well.</td>
</tr>
<tr>
<td>Byte order (Endianness)</td>
<td>PowerPC is bi-endian (primarily configured as big-endian), Intel architecture is little-endian</td>
</tr>
<tr>
<td>Bit Fields</td>
<td>PowerPC is “down bit ordered”. Intel architecture is “normal bit ordered” aka “up bit ordered”</td>
</tr>
</tbody>
</table>

*Other names and brands may be claimed as the property of others.*
# Architecture Differences ARM* vs. Intel® Architecture

<table>
<thead>
<tr>
<th>Instructions</th>
<th>ARM* and Intel® architecture instructions are very different. For some instructions there is no one to one (ARM to Intel architecture) equivalent. Refer to Intel® 64 and IA-32 Architectures Software Developer Manuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment</td>
<td>Pointer alignment, e.g. 1 byte aligned on x86, type dependent on ARM, e.g. 4-byte integer must be 4 byte aligned. Structure size and alignment. E.g. a struct with 3 characters on x86 is 3 bytes; it is 4 bytes on ARM. Intel architecture instructions vary in size and therefore do not require alignment.</td>
</tr>
<tr>
<td>Vector Oriented Instructions</td>
<td>ARM uses Vector Floating Point (VFP) instructions, Advance SIMD (NEON), DSP Enhanced Instructions. Intel architecture uses Intel® Streaming SIMD Extensions (Intel® SSE).</td>
</tr>
<tr>
<td>Signed vs. unsigned char</td>
<td><code>char</code> is signed on x86 and unsigned on ARM. <code>CHAR_MIN</code> and <code>CHAR_MAX</code> have different values on x86/ARM. gcc compiler can force all <code>char</code> types to be signed: <code>-fsigned-char</code></td>
</tr>
<tr>
<td>Calling Conventions Specified by ABI</td>
<td>Arguments are passed in registers and on the stack for ARM. For Intel architecture, arguments are passed on the stack.</td>
</tr>
<tr>
<td>Byte order (Endianness)</td>
<td>ARM is bi-endian; Intel architecture is little-endian</td>
</tr>
<tr>
<td>Bit Fields</td>
<td>ARM is bit ordered depending on endian selection. Intel architecture is “normal bit ordered” aka “up bit ordered”</td>
</tr>
</tbody>
</table>

*Other names and brands may be claimed as the property of others.
Intel® Embedded Design Center provides technical resources for Intel’s newest embedded processors and chipsets including confidential documentation, training, online support and step-by-step guidance. Take a tour.

- Step-by-step
- Hardware platforms
- Software
- Online training
- Applications
- Embedded support
- Embedded solutions advisor
- Embedded platform roadmap
- Embedded board planner

- Intel® embedded community
- Discussion forums
- Linux driver for US15W SMBus controller? in Software & Tools Discussions
- Request more extensions on Embedded Board Planner in Hardware Discussions
- Problem with setting rotation modes using IECO driver on

- Embedded blogs
- Roving Reporter: Adapting to the Extremes of Rugged Design in Hardware Blog
- Roving Reporter: PC/104 and beyond – EPIC, EBX, SUMIT, and more in Hardware Blog
- Roving Reporter: Touch-Screen Automation, Simplified

Embedded events
- Intel Developer Forum
  Monday, September 13, 2010
- Austin Real-Time & Embedded Computing Conference
  Tuesday, September 14, 2010
- Arrowfest 2010 – Chicago
  Thursday, September 16, 2010

More Events

Learn more
- Intel® product technologies
- Where to buy
- About the Intel® EDC
Break Away with Intel® Atom™ Processors
A Guide to Architecture Migration

Get Your Free Intel Press Book
Break Away with Intel® Atom™ Processors: A Guide to Architecture Migration

For details visit:
www.intel.com/go/breakaway-offer
References

• Intel® Atom™ Processor Technical Documentation - Low Power Features Chapter
• Intel® Embedded Alliance Solutions Center
  – http://www.intelcommsalliance.com/kshowcase/view/?iid=subhdr+ptr_embe
  d_alliance
• Intel® Embedded Design Center (EDC) http://edc.intel.com/
• Intel EDC: Migrating to Intel Architecture
  • http://edc.intel.com/Step-by-Step/Migration/
• Intel Press Book: “Break Away with Intel® Atom™ Processors” - Buy it at:
• Intel Product Information http://ark.intel.com/
• Intel® Software Development Products
• Intel® Software Network http://software.intel.com

• Intel® Tools for Intel® Atom™ Processors