



White Paper
Mark Oliver
Senior Systems Engineer
Intel Corporation

ARM* to Intel® Atom™ Microarchitecture - A Migration Study

November 2011

326398-001



Introduction

At Intel, our engineers do not perform migrations from ARM* to Intel Architecture®. We strive to create the best microprocessors and chipsets, based on x86 architecture and enable developers to use them by assuring the best resources and information is available. To help us understand the challenges of an ARM to IA migration, we enlisted the services of an independent engineering and design firm that had an actual migration task at hand. This paper will describe their experience and how they successfully migrated to IA.



The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. www.intel.com/embedded/edc.



Contents

Introduction	2
Contents	4
Why Migrate?	5
Defining Requirements	7
System architecture	7
Security	7
External resource availability	7
Assessment	8
Software Considerations	8
Application Porting	9
Summary and Conclusions	9



Why Migrate?

Not all existing ARM designs are good candidates for migration to IA. Older, deeply embedded designs using an RTOS or custom OS with an app written in a non-portable language are the most difficult to migrate. This is especially true if you have no need to add new functionality or increase processor/execution speed.

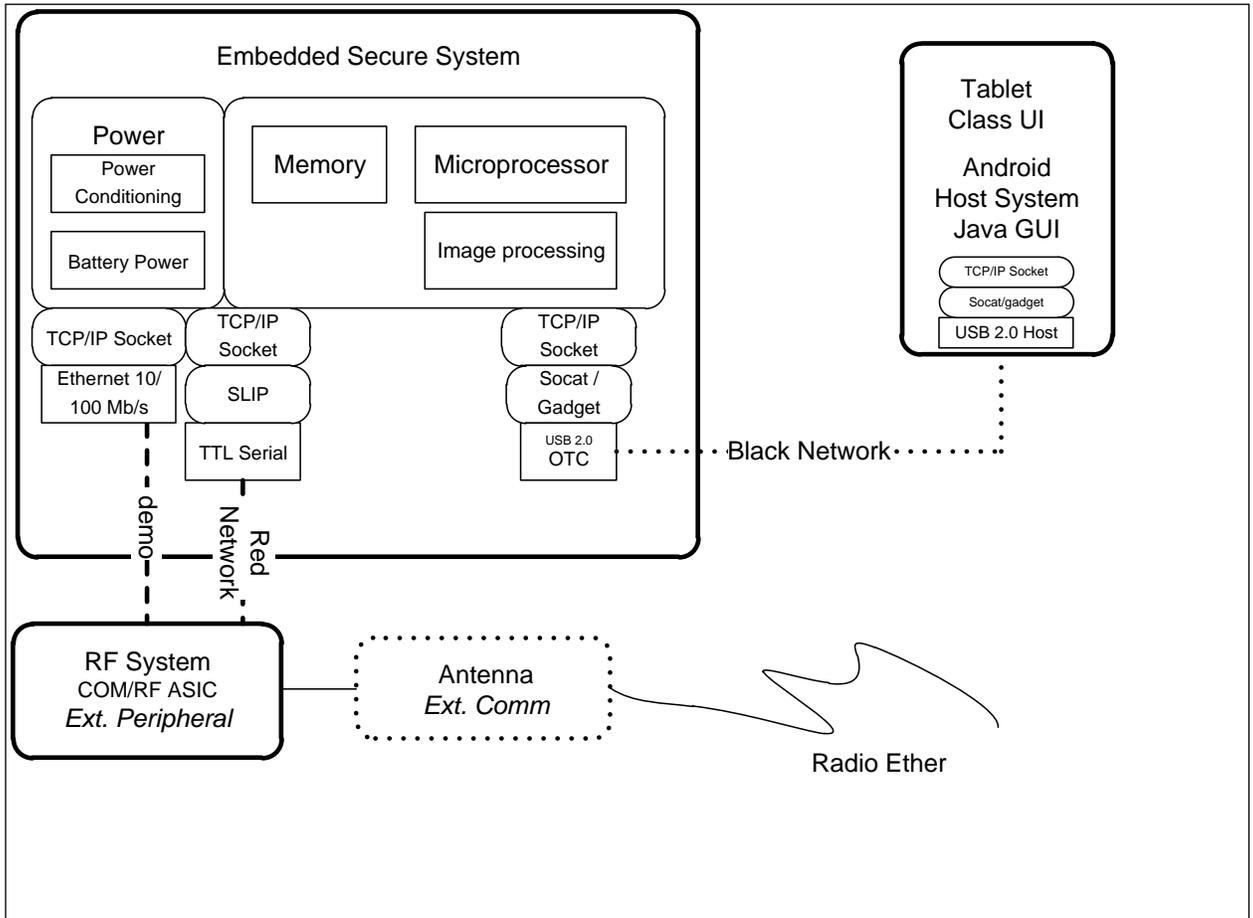
Apply the following criteria to determine if it makes sense in your situation.

- Can your application benefit from added functionality, increased execution speed or computational power?
- Will these added features and functionality allow you to adjust your bill-of-materials cost structure?
- Can your power budget comprehend the additional requirement and additional power rails?

The firm in the study found several reasons to migrate their existing ARM design to Intel Atom microarchitecture. To understand, let's look at their application. They had an existing commercial tablet computer that needed to have all data communications secured, both incoming and outgoing. To accomplish this, they created a "backpack" device in a shell case that fit on the tablet or smartphone. They disabled the internal radios in the tablet, and routed the secure (encrypted) data and communications over USB to the backpack device.



Figure 1. Block Diagram



In this example, both the CPU in the tablet and the backpack device are ARM architecture based. Their intention was to migrate the ARM system in the backpack to IA using the Intel Atom processor (see Figure 1).



Defining Requirements

The firm defined their requirements and applied them to IA in the following areas of consideration.

System architecture

- The secure and non-secure systems must be separated.
- Battery life must allow portable one-day use between charges.
- IA cost adders for BIOS, additional voltage regulators, and clocks must be minimized.

Security

The target application is of a sensitive nature that might be used in the military or government sectors. With Intel® Virtualization Technology under consideration, could a virtualized system be secured?

External resource availability

- Documentation – A combination of documents from the Intel EDC, IA developer community, and Linux Wikis were found most helpful.
- Reference hardware – No Intel manufactured boards were available to the firm. However, they were able to acquire two development boards based on Intel Atom microarchitecture, created through a joint effort between Intel and Emerson*.
- Software tools – The firm worked through the wide variety of tools available that can build a board support package (BSP) and port an OS to custom hardware. Some were included with the OS and others needed to be procured separately. JTAG tools were readily available for the physical transfers.
- Power budget – The Intel Atom processor required a considerable amount more power than the ARM* CPU it was replacing. This would require a larger battery and add extra weight and bulk.
- Cost – The Intel Atom processor was not on price parity with the ARM-based processor it would replace. The cost was closer when utilizing Intel VT, where one Intel Atom processor would replace two ARM based processors.
- Form factor – The Intel Atom processor must fit on a small form factor board in the backpack. Thermal requirements must be achieved with the use of a moderate sized passive heat spreader.



Assessment

During this process, several decisions were shaped. Intel VT was not known to the firm, but they discovered it during their initial investigation. The firm determined that they could secure the system while utilizing virtualization. A single-core Intel Atom processor provided all the compute and encryption power needed for the application. A tablet based on the Intel Atom processor and running Android* was judged to provide a good user experience, with cycles to spare for the virtualized encrypt/decrypt application. This made the migration simpler in the following ways:

- It negated the need for BIOS or bootloader development. They found a proprietary hypervisor and kernel solution that was suitable for the secure system portion.
- Power and cost targets were reduced as well. The price of a second board, case, battery, and supporting silicon were eliminated.

At this point, they looked at the remainder of the advantages of the Intel Atom microarchitecture.

Performance

- Intel Atom microarchitecture would allow them to augment existing functionality and add new features, making their solution more scalable.

IA ecosystem and infrastructure

- Wide choice of development tools and drivers.

Power profile

- An OS that utilizes Intel Atom processor power states and Intel SpeedStep® negate the additional active power. Atom executes tasks faster and returns to low power or standby faster. This also helps to reduce thermal power.

Software Considerations

It was found during the requirement stage, that the tools needed to migrate or create a system on IA are varied and abundant. The firm looked at boot loaders, operating systems and BSP tools, along with compilers and tools to aid in migration of their crypto application.



Bootloaders

They evaluated the Intel® Boot Loader Development Kit (BLDK) and correctly identified that it would be more useful for those developers creating their own board. They also looked at Coreboot* and came to the same conclusion. The developer must have schematic level detail for the target system board. This finding coupled with their decision to use a consumer tablet made this unnecessary. The board's native BIOS was used for further development.

OS and BSP

Ubuntu Linux*: It was found to be very straightforward and easy to install, but not inherently secure.

WindRiver* Linux: They summarized it as "normal Linux" with proprietary kernel configuration tools.

Green Hills Integrity* was found to have the right level of security, but would require more porting effort than Linux.

The conclusion was that Linux is very good at abstracting the hardware it's running on. That, coupled with the ease of moving over their Java-based app from the ARM system made Linux the clear choice. The BSP was found to be unneeded with this solution.

Application Porting

Because the proprietary application was recent code written in C, it ran as-is on the secure Linux kernel. No migration was necessary, but the firm planned to use the Intel® Embedded Software Development Tool Suite for Intel® Atom™ Processor to optimize their application for execution speed.

A key learning was that because of standardization of drivers, especially for cross-platform peripherals like USB, "abstraction is the key to migration".

Summary and Conclusions

Here are the key points that were discovered in the course of the study.

- The available documentation and hardware was more "PC-like" than a traditional embedded developer might expect.



- The IA learning curve was manageable, even for a small engineering firm. Even for the smaller customer, online help was available from the Intel EDC, the IA developer community and Wikis.
- Discovering VT was the turning point in this circumstance. The ability to merge the secure and less secure components on one platform CPU, allowed them to reduce the bill-of-materials, associated cost, weight, size, and power profile.
- Software migration was straightforward utilizing a Linux OS, Android/Java apps, and their proprietary application written in C.
- Not every embedded ARM design is a candidate for direct migration to IA, but with basic system architecture consideration, it becomes feasible in more cases.
- The more IA features and performance you can utilize, the larger the advantage for Intel Atom microarchitecture.
- The IA resource ecosystem lends itself well to embedded development.



The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. www.intel.com/embedded/edc.

Author

Mark Oliver is a Systems Engineer with the Intelligent Connected Devices Group at Intel Corporation.

Acronyms

ARM*	Advanced RISC Machine
CPU	Central processing unit
IA	Intel® Architecture
RISC	Reduced Instruction Set Computing
RTOS	Real-Time Operating System
VT	Intel® Virtualization Technology



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.

This paper is for informational purposes only. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Core Inside, i960, Intel, the Intel logo, Intel AppUp, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, the Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. logo, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, InTru, the InTru logo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX, Moblin, Pentium, Pentium Inside, skool, the skool logo, Sound Mark, The Journey Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2011 Intel Corporation. All rights reserved.