

presented by

ARM



Introduction & Update

UEFI Spring Plugfest – July, 2012
Andrew N. Sloss (ARM)

Agenda



- background
- terminology
- status
- sct
- eco-system
- hints and tips



.background



Facts



- Processors shipped in 2012
 - 7.9 Bu (4 Bu in 2009, 6 Bu in 2010)
- Processors shipped in total
 - 30+ Bu
- Processor licenses
 - 850+
- Semiconductor partners
 - 290+
- Process technology
 - 20nm – 250 nm
- Connected community members
 - 950+



Connected Community 950+



Silicon Partners

A large grid of logos for Silicon Partners, including companies like Intel, ARM, TI, and various semiconductor manufacturers.

Design Support Partners

A large grid of logos for Design Support Partners, including companies like Cadence, Synopsys, and various design tool providers.

Software, Training and Consortia Partners

A large grid of logos for Software, Training and Consortia Partners, including companies like Microsoft, Oracle, and various software and training providers.



.terminology



Terminology



Architecture
"ARMv7A"

Processor Micro-
Architecture
"Cortex-A15"

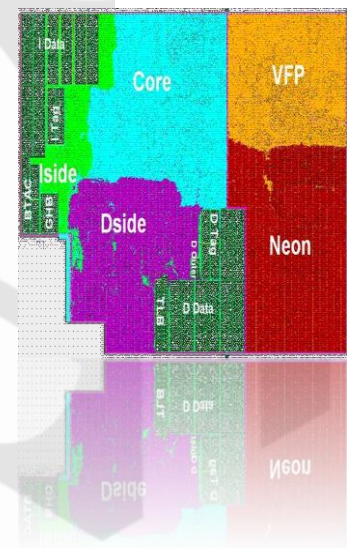
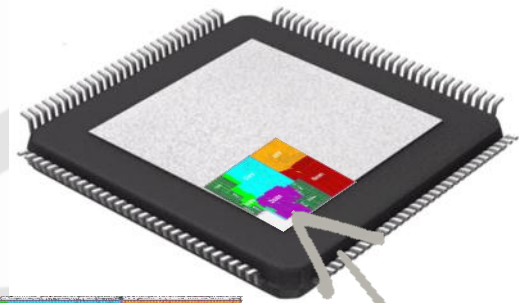
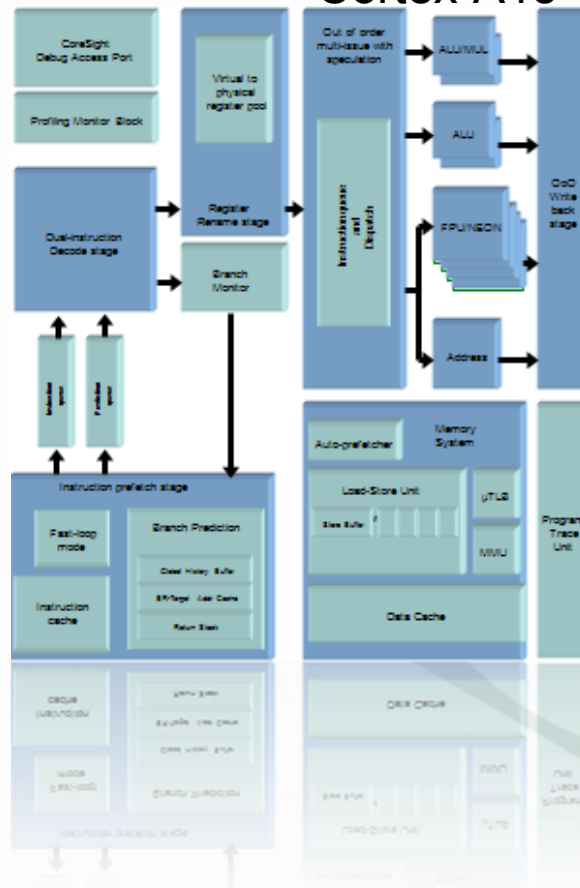
Processor Hard-Macro
Implementation

ARM® Architecture
Reference Manual
ARMv7-A and ARMv7-R edition



Copyright © 1986-1998, 2000, 2004-2010 ARM Limited. All rights reserved.
ARM DDI 0406B_errata_2010_Q3 (ID100710)

ARM logo (mirrored)





.status



Introducing UEFI on ARM



- Driving forces for UEFI on ARM
 - Processor and system complexity increasing
 - Support for existing OEMs that are developing ARM processor-based solutions using UEFI
 - Help standardize the boot procedure for ARM processor-based platforms
 - On-going ARM goal is to improve the hardware-software interface for Operating Systems that target the ARM architecture
- Advantages to ARM partners and OEMs
 - Write once per platform and saves costs in boot loader development/engineering
 - UEFI specification written down and peer reviewed
 - Tested UEFI drivers available from 3rd party peripherals providers
 - Provides an environment for manufacturing test

Introducing UEFI on ARM



- UEFI ARM Binding Sub-Team (ABST) was formed in 2008
 - Apple, ARM, HP & Microsoft are the current public members
 - UEFI Specification focuses on ARMv4 to ARMv7A
 - ABST is starting to work on virtualization and ARMv8 (Aarch64) bindings
- Specification 2.3.1 released - includes the ARM Bindings
 - Specifies the state of the processor & system post UEFI initialization
 - Defines the Runtime & Pre-Boot Services ABI (post boot services)
- Verification Tests
 - Oct'11 SCT 2.3 have been updated to support ARM
- ARM-UEFI supported commercially



BYOSOFT
百 敖 软 件



Innovation at the CORE™



Public Implementations



- Tianocore EDK2 project contains ARM platform support
- ARM Holdings now maintain the ARM packages
 - Since February 2011
 - With contributions from Apple, HP, Linaro, etc...
 - ArmPkg - Architectural and standard ARM peripheral support
 - ArmPlatformPkg - ARM standard development board support
 - Plus some other ARM related packages
- Future ARM development board support provided by Linaro
 - Specifically by the Linaro ARM Landing Team
- Linaro can potentially support boards for other member companies
 - Would be supported by their Landing Team

ARM Platform Status - EDK2



- ARM Hardware Platforms
 - BeagleBoardPkg (OMAP3530 SoC – Cortex A8)
 - ArmPlatformPkg/ArmVExpressPkg (ARM Versatile Express Board)
 - Cortex A9x4
- ARM Fast Model Support
 - ArmPlatformPkg/ArmRealViewEbPkg (ARM RealView Emulation Board)
 - Cortex A8 & Cortex A9x2)
 - ArmPlatformPkg/ArmVExpressPkg
 - Cortex A9 MPCore and A15 MPCore



ARM Platform Status - Other



- Other ARM-based Platforms exist outside of the EDK2 repository
- ARM Development Platforms
 - ArmPlatformPkg/ArmVExpressPkg
 - Cortex A5
 - Cortex A15x2 (Limited Hardware Availability)
 - ArmPlatformPkg/ArmTuscanPkg
 - Cortex A9x2
 - PCI Bus with SATA and USB 2.0 controllers
- Other Platforms
 - SamsungPlatformPkg/OrigenBoardPkg
 - Based on Cortex A9x2



.sct



ARM UEFI SCT Implementation



- Initial port was completed in August 2010
- ARM Holdings own ARM UEFI SCT implementation
 - Initially only available as a patch from UTWG documents area
 - Now integrated with main SCT package and available in UEFI member documents area
- Builds in both Windows and Linux environments
 - Primary toolchain is ARM CC (formerly RVCT)
 - ARM GCC also supported
- ARM SCT binary is tested on Beagleboard and ARM Versatile Express Platforms
- Can extend SCT build framework to write platform specific unit test cases



.eco-system



Growing Eco-system





.hints & tips



Hints & Tips (1)



- When writing a UEFI driver ensure you are not making the assumption you are running on a ARM Platform !
 - Risks are that your driver could not work on another platform even if it is a ARM-based platform (e.g.: different memory map or architectural controllers)
 - Benefit is you can reuse and test your driver on a wider range of devices
 - Hint 1: UEFI Specification provides methods to access architectural components
 - Hint 2: Avoid making your UEFI driver dependent on EDK2 Libraries

Hints & Tips (2) – Use cases



- Architectural Timer: On ARMv7, it exists at least two different timers (... and actually many exist!)
 - ARM SP804 DualTimer: Memory mapped controller
 - ARM Architectural Timer (used in the latest ARM Application Processors)
 - The solution is ...
 - use UEFI Boot Services (*see section 6 of the UEFI Spec*)
- Cache Management: difficult for engineers working on ARM architecture (compare to other architectures)
 - The solution is ...
 - stick with the UEFI Specification and use the Boot Services function `AllocatePages()` (*see section 6 of the UEFI Spec*)

Getting started



- Recommend obtaining a Cortex A8 development board
 - Available from <http://beagleboard.org>
- Recommend getting a copy of “Beyond the BIOS”
 - Available from Amazon
<http://www.amazon.com/Beyond-BIOS-Implementing-Extensible-Interface/dp/0974364908>
- UEFI Learning Center
 - http://www.uefi.org/learning_center/
- Intel UEFI Information
 - <http://software.intel.com/en-us/articles/about-uefi/>

Getting started



- Source tree can be found on SourceForge
 - EDK2 Source Tree
 - <http://sourceforge.net/apps/mediawiki/tianocore/index.php?title=EDK2>
 - BeagleBoard UEFI wiki
 - <https://sourceforge.net/apps/mediawiki/tianocore/index.php?title=BeagleBoardPkg>
 - Samsung Platform wiki
 - <http://sourceforge.net/apps/mediawiki/tianocore/index.php?title=SamsungPlatformPkg>
- Linaro (Boot Architecture)
 - <https://wiki.linaro.org/OfficeofCTO/BootArchitecture>
- Linaro (ARM Landing Team)
 - http://www.linaro.org/members/arm/ve_12.03#tab3



.summary



Summary



- UEFI brings a neutral boot loader capable of booting both open-source and non-open source Operating Systems
- BSD-like license allows for security and specialized code to be hidden and not released
- UEFI is a framework that grows and shrinks depending upon requirements
- Specification written down and peer-reviewed
- Lots of validated software already written for the PC industry e.g. PCI probing
- Ability to support a 3rd party peripheral eco-system
- Write-once, validate-once, support all OSes



Questions?



Thanks for attending the
UEFI Summer Summit 2012



For more information on
the Unified EFI Forum and
UEFI Specifications, visit
<http://www.uefi.org>



presented by

ARM