UEFI ARM Update

Presented by Mitch Ishihara

presented by

ARM®

UEFI Plugfest – October 2014

Agenda

- Economics
- Objectives
- Status Overview
  - Specifications
  - Implementation
  - Testing
- Technology
- Resources

Economics

Economics

What are the ARM numbers?

Processors shipped in 2013	: **10+ Bu** (~8.7 Bu`12)
Processors shipped in total	: **50+ Bu**
Processor licenses	: **1000+** (850'12)
Semiconductor partners	: **310+** (290'12)
Process technology	: **10 – 250 nm**
Connected community members	: **1000+** (950'12)

Economics (1000+)

Objectives

Why UEFI on ARM?

Driving forces for UEFI on ARM

- Processor and system complexity increasing
- Support existing partners' ARM processor-based UEFI solutions
- Help standardize boot process for ARM processor-based platforms
- Improve hardware-software interface for OS that targets the ARM architecture

Advantages to ARM partners and OEMs

- Write once per platform, saving costs in boot loader development
- UEFI specification peer reviewed and published
- Tested UEFI drivers available from 3rd party peripherals providers
- Provides an environment for manufacturing tests

ARM UEFI Vision

Provide standard ARM architectural support

- Correctness in implementation within ARMv7-A and ARMv8-A architectures
- Future Proof through standardization (rather than proprietary) reference software
- Focus on reducing fragmentation and overall partner support costs

Provide reference ports of UEFI for ARM development platforms

Support BIOS (and other) partners' UEFI development

- Directly and through organizations such as Linaro

ARM UEFI Engineering

UEFI support for the ARM Architecture

- Maintain ARM packages and docs in Tianocore EDK2 repository
- Implement support for new ARM architectures, CPUs and system IP
- Implement common UEFI features or applications for ARM
- Maintain SCT for ARM and validate on standard platforms
- Align with relevant ARM Platform Design Documents (PDDs)

UEFI support for ARM platforms

- Porting for new ARM development platforms
- Maintained within EDK2 (for standard platforms) or other neutral repository

Help partners with UEFI platform code management and development

Juno ARM Development Platform

ARMv8-A Architecture

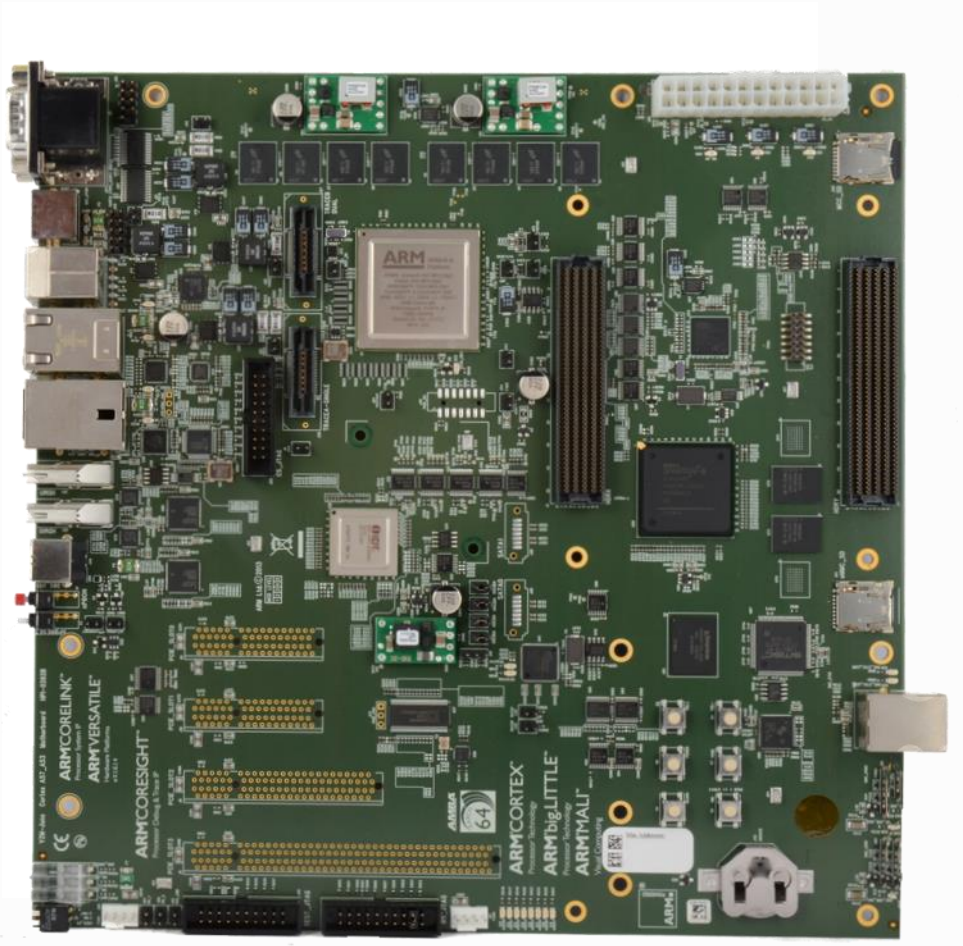
Cortex-A57 and Cortex-A53 big.LITTLE

Reference firmware

ARM Trusted Firmware

UEFI Tianocore EDK2

UEFI support booting any rich OS



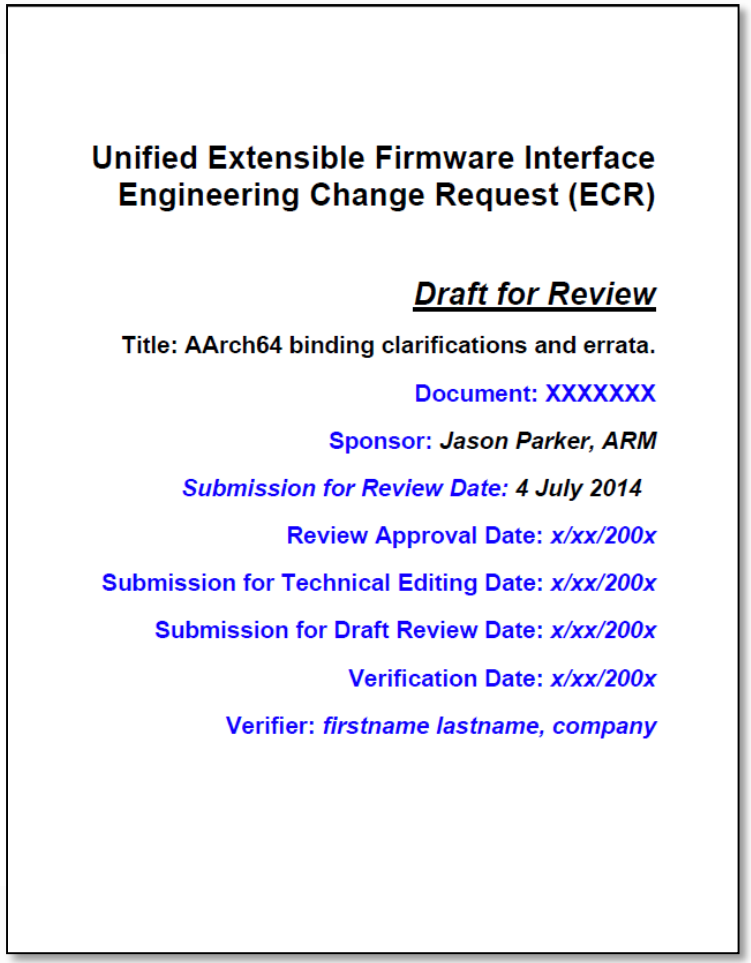
Status Overview

UEFI Specification Updates

UEFI Forum ARM Binding Sub-Team (ABST) activities

Approved UEFI Specification 2.4 AArch64 binding clarifications and omissions errata

- Boot configuration requirements (modes, registers, and bit settings)
- Memory alignment restriction to enable 64K page mappings at runtime.
- Functionality to allow use of runtime Services from either EL1 or EL2 exception levels.
- Fixes and runtime usability improvements for AArch64 systems.
- Queued for publication in next update to UEFI Specification



Unified Extensible Firmware Interface Engineering Change Request (ECR)

Draft for Review

Title: AArch64 binding clarifications and errata.

Document: XXXXXXX

Sponsor: *Jason Parker, ARM*

Submission for Review Date: 4 July 2014

Review Approval Date: *x/xx/200x*

Submission for Technical Editing Date: *x/xx/200x*

Submission for Draft Review Date: *x/xx/200x*

Verification Date: *x/xx/200x*

Verifier: *firstname lastname, company*

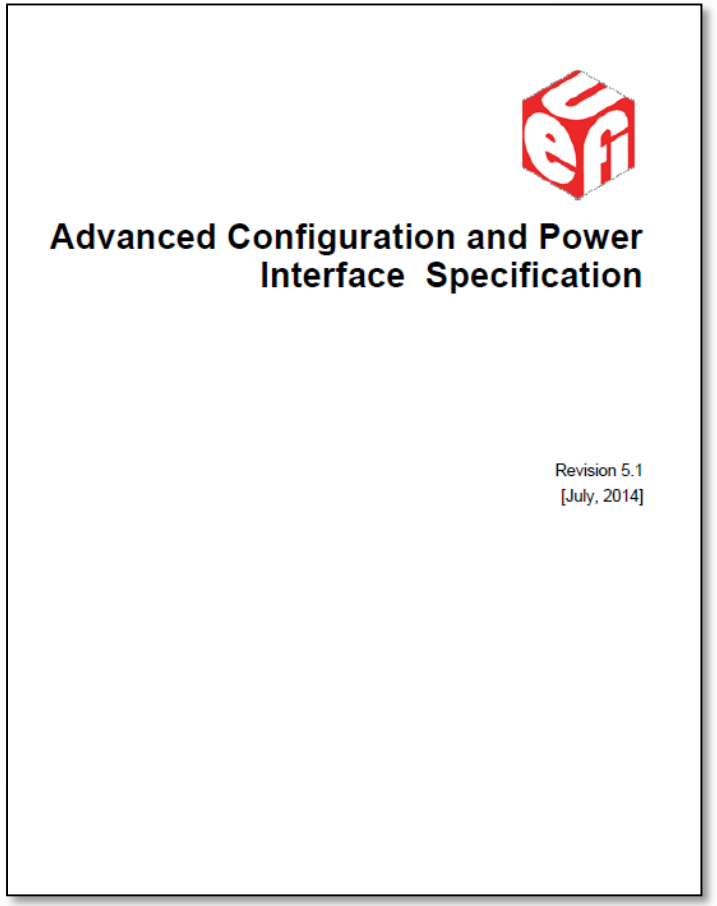
ACPI Specification Updates

ACPI 5.1 adopted July 22nd addressing the following for ARM:

- ✓ Virtualization support for GICv2m, GICv3 for alignment with SBSA
  - Improved GIC Architecture description and compatible with SBSA Level 1
- ✓ Added PSCI support
- ✓ Added platform/system memory mapped Generic Timer support
  - Alignment with SBSA Level 1
- ✓ Added support for SBSA Level 1 Generic Watchdog Timer
- ✓ Support for clock management and other Device Specific Data features of an SoC

Active ACPI work on support for:

- Idle management
- CPU topology
- SMMU or IO topology
- GIC Interrupt Translation Service

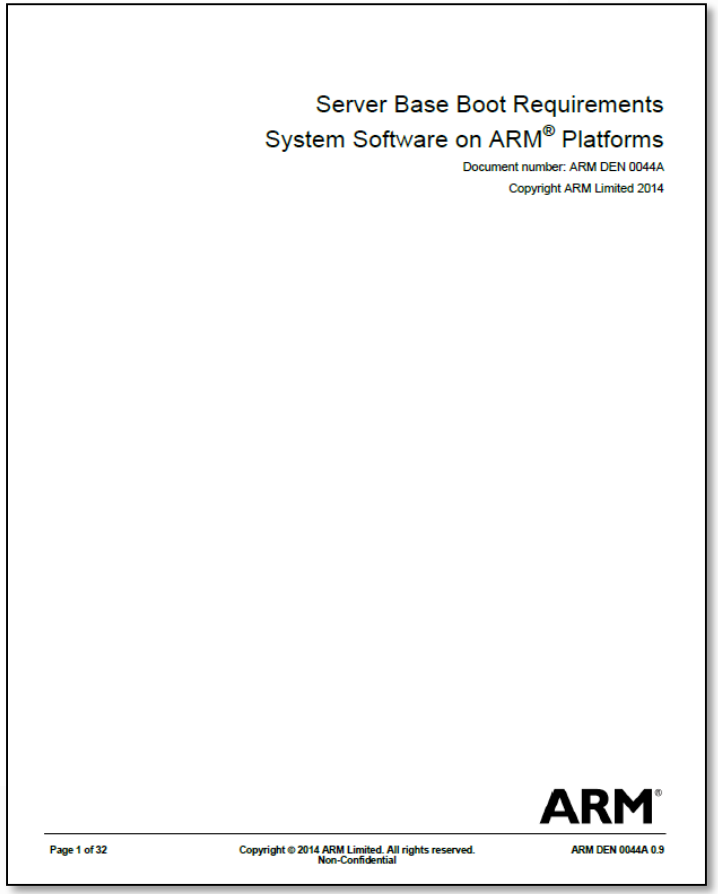


ARM Specification Updates

ARM boot architecture

Published ARM Server Base Boot Requirements

- Targets SBSA-compliant 64-bit ARMv8 servers
- Defines base firmware requirements for out-of-box OS or hypervisor support
- UEFI Specification 2.4B or later
  - Boot services, Runtime services, protocols
- ACPI Specification 5.1 or later
  - ACPI Tables: mandatory, recommended, optional
  - ACPI Methods and Objects


Server Base Boot Requirements
System Software on ARM® Platforms
Document number: ARM DEN 0044A
Copyright ARM Limited 2014
ARM
Page 1 of 32
Copyright © 2014 ARM Limited. All rights reserved.
Non-Confidential
ARM DEN 0044A 0.9

EDK2 Implementation Updates

Tianocore EDK2

- Pushed and stabilized AArch64 code into the EDK2 repository
- Juno support (AArch64 platform, ACPI support)
- Enabled VirtIo support on ARM Fixed Virtual Platform models
- Android FastBoot support
- UEFI Runtime Services support
  - Documentation: http://tianocore.sourceforge.net/wiki/ArmPkg/Runtime
- Optimization (size & speed)

Future Tianocore EDK2

- Enabling LLVM tool chain
- More optimization!

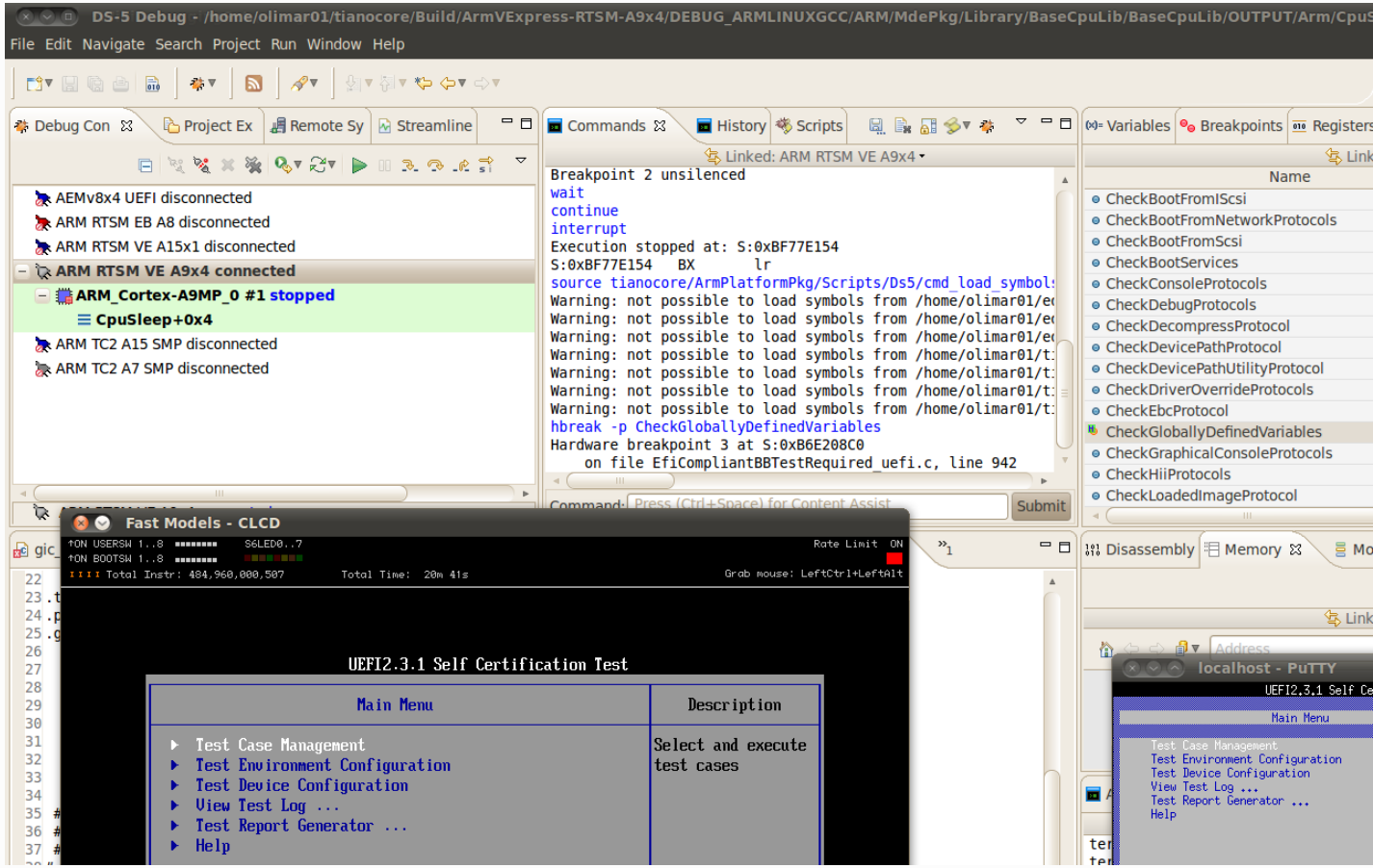
SCT Implementation Updates

SCT "new code base"

- Aligned with EDK2
- Enables both EDK Shell and UEFI Shell 2.0
- Defect fixes
- Optimization (size & speed)

UEFI v2.4B SCT "release candidate"

- Test and send feedback to UTWG



New Features in ACPI 5.1 for ARM

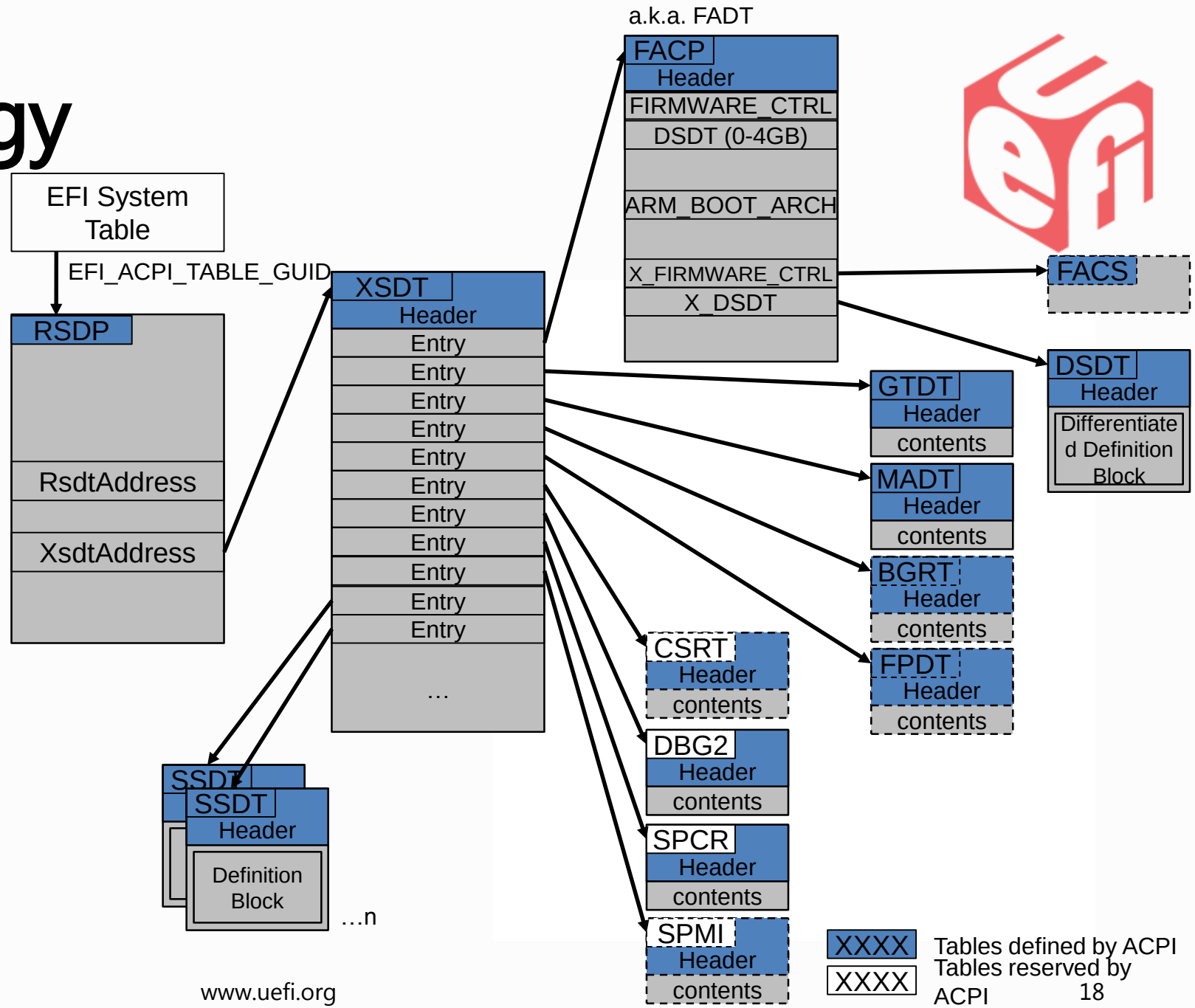
Technology

ACPI Technology

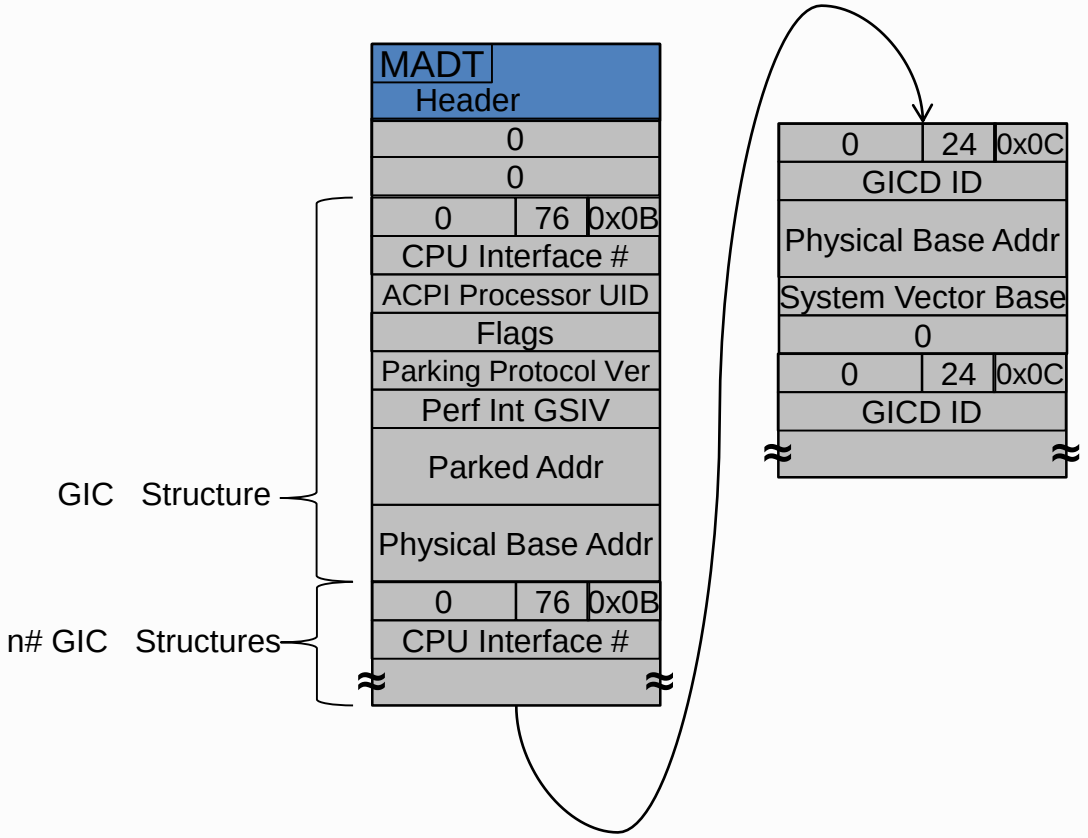
Static tables and primary runtime interpreted control methods provided by system firmware to the OS for system configuration, power management and error handling

Processor architecture agnostic

Refer to SBBR for ARMv8 server ACPI requirements



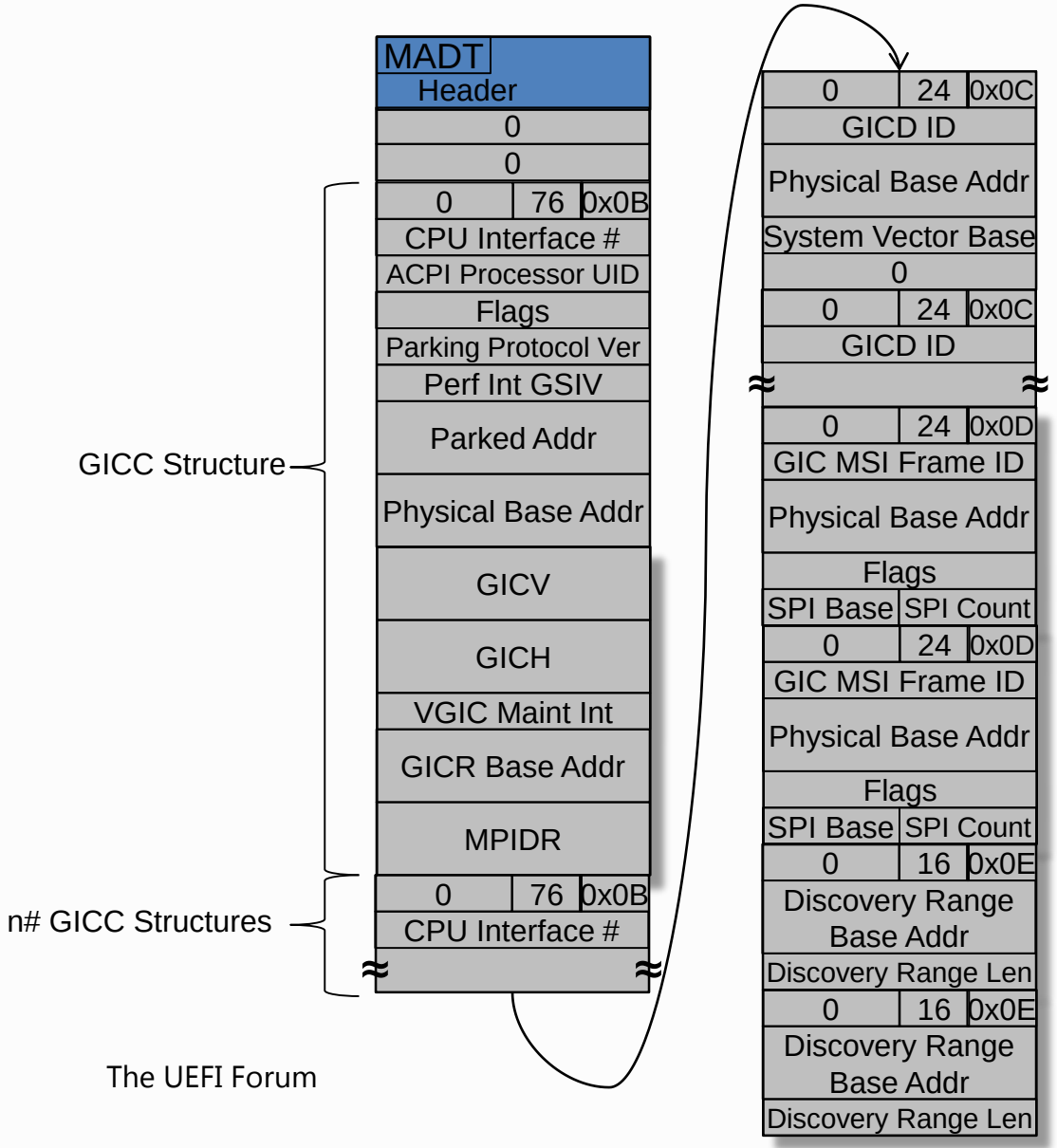
ACPI 5.0: GIC



Basic ARM Generic Interrupt Controller Architecture support

- Missing support for alignment with SBSA

New Features in ACPI 5.1: GIC



Updated Generic Interrupt Controller support

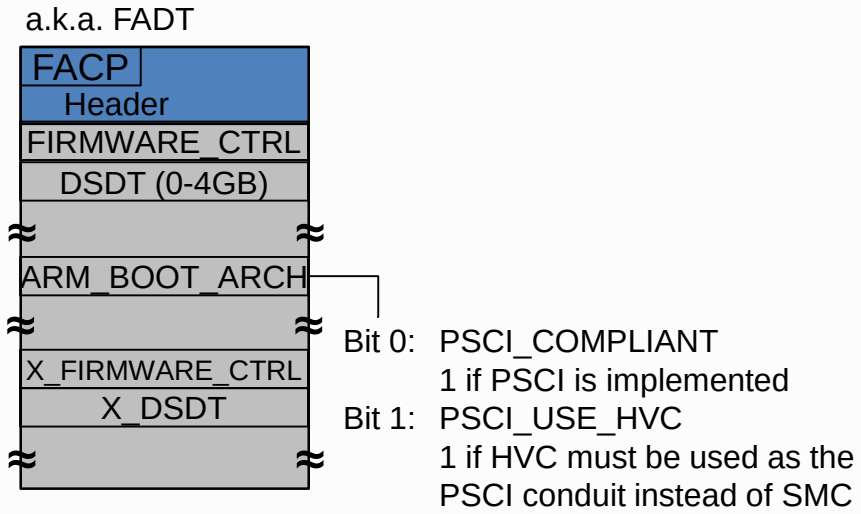
GIC Support has been extended to cover:

- GICv2 virtualization
- GICv2m (optionally required in SBSA Level 1)
- Partial support for GICv3
  - Redistributors are supported
  - Interrupt Translation Service work in progress
- Improved consistency with "ARM ARM" language

Now called GICC and GICD structures of the MADT

To do: Add ITS support

New Features in ACPI 5.1: PSCI



Support for PSCI

PSCI discoverability is provided by a new ARM Boot Flags field in FADT

MADT provides ways of identifying every core

- Enables the use of PSCI for:
  - Secondary core boot
  - Dynamic addition/removal of cores (hot-plug)
- Creates a path for use in idle management

To do: Use of PSCI in idle management. This will be worked on as part of the more generic idle management support for ARM

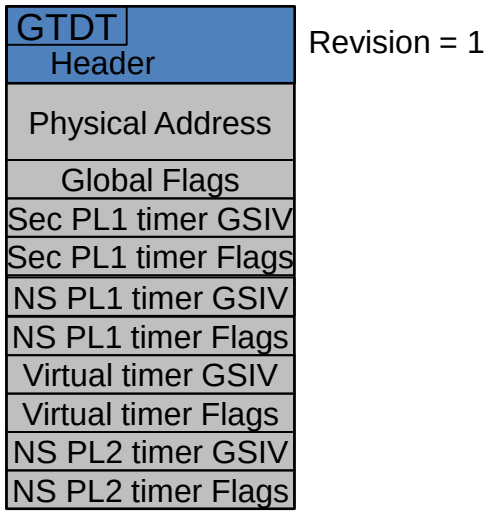
Power State Coordination Interface (PSCI)
http://infocenter.arm.com/help/topic/com.arm.doc.den0022b/index.html

ACPI 5.0: Generic Timer

Limited support for the Generic Timer Architecture

GTDT described timers that were implemented at the time and cannot describe:

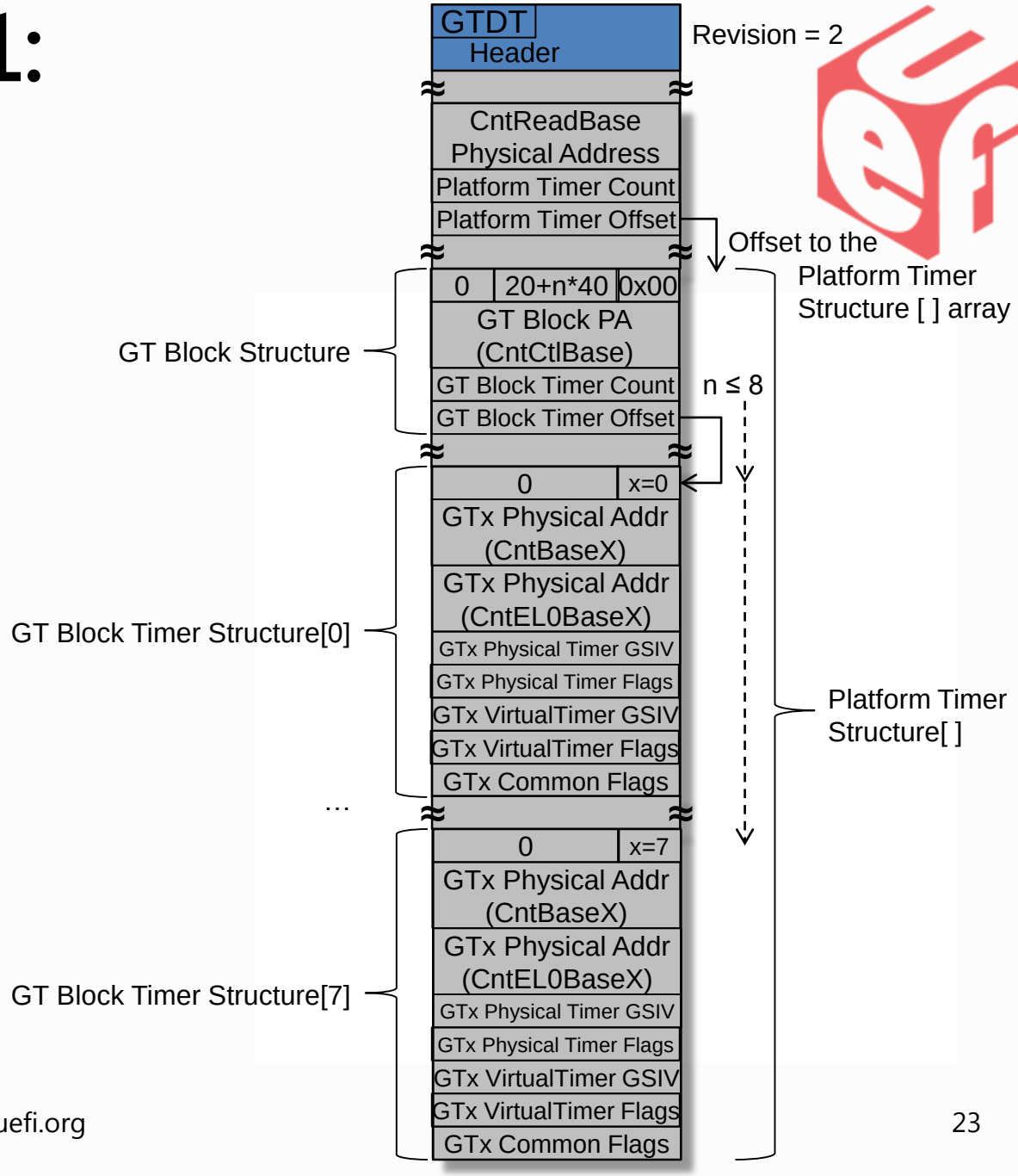
- Always-on per processor timers
- Memory-mapped platform timers
- Platform watchdog timers



New Features in ACPI 5.1: Generic Timer

Extended support for Generic Timer Architecture

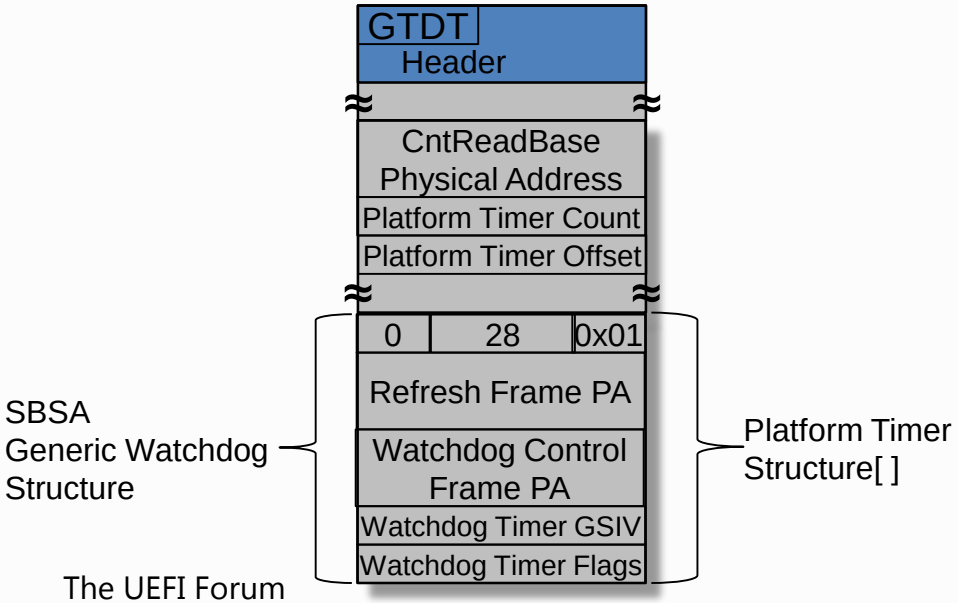
- It is now possible to describe platform memory mapped timers that are compliant with the ARMv7 or ARMv8 Generic Timer Architecture
  - Covered by extension to the GTDT table in the Platform Timer Structure[ ]
  - Secure or non-secure via GTx Common Flags
  - Always-on Capability via GTx Common Flags
- This is a requirement for SBSA Level 1 systems



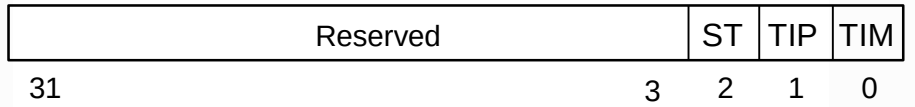
New Features in ACPI 5.1: Generic Watchdog Timer

Support for SBSA Level 1 Generic Watchdog Timer

- Covered by extension to the GTDT table in the Platform Timer Structure[ ]



Watchdog Timer Flags



Bit	Name	Description
0	Timer Interrupt Mode	Indicates the mode of the timer interrupt 1: Edge triggered 0: Level Triggered
1	Timer Interrupt Priority	Indicates the polarity of the timer interrupt 1: Active low 0: Active high
2	Secure Timer	Indicates whether the timer is secure or non-secure 1: Secure 0: Non-secure
31:3	Reserved	Must be zero

New Features in ACPI 5.1: Cache Coherency Attribute (_CCA)

A device identification object specifies whether a device and its descendants support hardware managed cache coherency

_CCA returns

- 0 - The device does not have hardware managed cache coherency
  - Software managed to ensure stale or invalid data is not accessed from the caches
- 1 - The device has hardware managed cache coherency

Allows platform designers to provide hardware cache coherency support on an as-needed basis for cost and performance reasons, without requiring new drivers to have knowledge of the platform

Provides flexibility in the firmware to indicate to the OS what support is provided in the platform

Optional Features in ACPI 5.1: Device Specific Data (_DSD)

An optional object used to describe device properties to device drivers

_DSD returns a variable-length package of Device Data Descriptor structures
- UUID and Data Structure tuples

UUIDs may be created by governing bodies (e.g. PCI SIG, UEFI Forum), OEMs or hardware vendors

UUID and data structure pairs are published via http://www.uefi.org/acpi

This method will help us to provide more generic solutions in clock control or other bespoke features

Resources

Resources

SCT

- How to contribute: http://tianocore.sourceforge.net/wiki/ArmPkg/HowToContributeSct
- Documentation to build/run/debug SCT: http://tianocore.sourceforge.net/wiki/ArmPkg/Sct
- GitHub source: https://github.com/UEFI/UEFI-SCT/
- UEFI SCT 2.4B ARM+AArch64 for Taipei Plugfest 2014 http://www.uefi.org/sites/default/files/resources/UEFI-SCT-ARM-AARCH64-TaiPeiPlugfest2014.zip

ARM Server Base Boot Requirements (SBBR)

- http://infocenter.arm.com/help/topic/com.arm.doc.den0044a/index.html

Summary

Summary

UEFI provides an OS agnostic boot loader that grows and shrinks depending upon requirements

UEFI Forum specifications written down and peer-reviewed

ARMv8-A AArch64 support for UEFI today

- Tightening of UEFI Specification AArch64 bindings
- ARM pushed and stabilized AArch64 code into the EDK2 repository

Testing UEFI v2.4B SCT underway for Q1 2015 release

ACPI Specification progresses for ARM in 2014

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

presented by

ARM®