The Role UEFI Technologies Play in ARM Platform Architecture

Spring 2017 UEFI Seminar and Plugfest
March 27 - 31, 2017
Presented by Dong Wei (ARM)
Agenda

- ARM Ecosystem Update
- Specification Updates
- SBSA/SBBR
- SBSA/SBBR Tests
- Questions
  - ODM/OEM/ISV Badge Program?
  - UEFI Driver Binary Format
Section Heading

ARM Ecosystem Update
Economics

• What are the ARM numbers?
  – Silicon with ARM IP shipped in 2016: 16.7 Bu
  – Cumulative total shipped: 100+ Bu
  – Processor + GPU licenses: 1400+
  – Licensees: 450+
  – Foundry partners: 5+
  – Process technology: 7 – 250 nm
  – Connected community members\(^1\): 1000+

\(^1\) Important for a collaborative business model
Connected Community
Specification Updates
ACPI Next

- New introduction chapter
- NUMA SRAT (System Resource Affinity Table) support for ITS (Interrupt Translation Service)
- CPPC (Collaborative Processor Performance Control) Support for multiple PCC (Platform Communication Channels)
- Processor Properties and Topology Table (PPTT)
- Extended PCC subspaces – bidirectional interface between the OSPM and the platform
- SDE (Software Delegated Exception) hardware error notification and SDEI (SDE Interface) table
- IORT, and ARM ACPI Table, will have an update soon
- Heterogeneous Memory Attribute Table (HMAT)
- NVM Label, ARS (Address Range Scrubbing) Updates, Translate SPA (System Physical Address), Platform RAS Capabilities Updates, ARS Error Injection
PSCI

- **Power State Coordination Interface** is the ARM standard for core and system power management
  - Supported by all major OSs, UEFI and ACPI
- Expect to release PSCI v1.1 in 17Q2
  - Improves reset support, and allows implementing system warm resets
ARM Trusted Firmware (TF)

- Standardized ARMv8-A EL3 firmware
  - Optional trusted boot firmware

- BSD licensed, contributions welcome
  - No CLA (Contributor License Agreement) needed

- Reusable reference code
  - Including PSCI…

https://github.com/ARM-software/arm-trusted-firmware
ARM TF and PSCI

- AArch64 and AArch32 library
- Mostly generic with thin platform layer
- Supports all mandatory PSCI v1.0 functions
  - and most optional ones
- Latest TF v1.3 adds
  - Power state residency statistics functions
  - Instrumentation of key PSCI operations
- TF implementation will track specification
ARM TF Runtime Stack

Normal World

<table>
<thead>
<tr>
<th>EL0</th>
<th>Guest OS</th>
<th>App</th>
<th>App</th>
</tr>
</thead>
<tbody>
<tr>
<td>EL1</td>
<td>OS Kernel</td>
<td>App</td>
<td>App</td>
</tr>
<tr>
<td>EL2</td>
<td>Hypervisor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EL3</td>
<td>SMC Dispatcher</td>
<td>PSCI Core Interface</td>
<td>ARM System IP library</td>
</tr>
<tr>
<td></td>
<td>PSCI Platform</td>
<td>SoC SMC calls</td>
<td></td>
</tr>
</tbody>
</table>

Trusted World

<table>
<thead>
<tr>
<th>Trusted Execution Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusted App</td>
</tr>
<tr>
<td>Trusted OS Kernel</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

UEFI Plugfest – March 2017
ACPI View

• A UEFI Shell utility
  – Provides a human readable output of the installed ACPI tables
  – Similar to SmbiosView
  – Provides extensive interface to validate ACPI tables
  – Useful for firmware developers to diagnose ACPI table issues that cause an OS to fail to boot
  – Assists in prototyping implementations against specification proposals
  – ARM initiated, collaborations welcome
  – https://github.com/tianocore/edk2-staging
Platform Architecture

• Base System Architecture (BSA)
  – Defines hardware requirements
• Base Boot Requirements (BBR)
  – Defines firmware requirements
• These specifications require a minimum set of hardware and firmware implementations that will ensure OS and firmware will interoperate
SBSA/SBBR

• SBSA/SBBR are the BSA/BBR for the enterprise systems
  – Developed using feedback from vendors across the industry (Silicon vendors, OSVs, Hypervisor vendors, BIOS vendors, OEMs and ODMs)
  – SBBR defines the required, recommended and optional UEFI, ACPI and SMBIOS interfaces
• SBSA are SBBR are now available at https://developer.arm.com/  
  – Current versions are SBSA v3.0 and SBBR v1.0  
  – No click through license required
SBSA/SBBR Compliance Tests

• SBSA test suite covers
  – SBSA PE properties
  – SBSA defined system components
  – SBSA rules for PCIe integration
    • Based on the PCIe specification
    • Based on standard OS drivers with no quirks enabled

• SBBR test suite covers
  – UEFI testing based on the UEFI SCT
  – ACPI testing based on FWTS
  – SMBIOS testing
SBSA Tests

• Provided as open source
  – Apache v2 License
• Built on top of a Platform Adaptation Layer
  – ARM will support one based on UEFI and ARM Trusted Firmware
  – A silicon vendor can also port to a bare metal environment
SBBR Tests

• From 3 sources (all open source)
  – UEFI SCT* (ARM will upstream into SCT)
  – FWTS (ARM + Linaro will upstream)
  – Standalone (ARM provides through github and packages into LuvOS image)

• Note: UEFI SCT is currently for UEFI member only. Would like to see it open source
Unified Release

• A unified software release, to tie all of these deliverables together with the enterprise FVP model
• Planned for future
SBSA/SBBR Tests Release

- Overarching github including SBBR
  - https://github.com/ARM-software/arm-enterprise-acs

- SBSA github
  - https://github.com/ARM-software/sbsa-acs
SBASA/SBBR Roadmap

2016 Q4

2017 Q1

2017 Q2

2017 Q3

2017 Q4

Future

SBASA – alpha
SBASA (source + binary)

SBBR – alpha
SBBR beta

SBASA – beta * PCIe testing alpha

SBASA/SBBR EAC

SBASA/SBBR update

SBASA/SBBR Compliance process

SBASA/SBBR update

SBASA-1.next

SBASA-3.next

SBBR-1.next

SBBR-next-next

SBASA-4.0

Specs

SBASA/BBR Test suites
Questions to the ARM Community
SBSA/SBRR Certification

• To improve the out-of-box experience for OS vendors and system users, ARM received feedback that a badge program certifying the SBSA/SBRR Compliance can be useful.

• Feedback?
UEFI Driver Binary Format

• EBC is a cross-architecture solution
  – One driver image for all ISAs
  – Open-source EBC Interpreter for ARM upstreamed to tianocore
• However,
  – Benefit cannot be realized if x86 uses its native format, unless more ISAs become relevant
  – No supported EBC Compiler
  – No Secure Boot Signing for EBC Drivers
• Can the industry come together to solve these problems?
  – If not, propose that ARM AArch64 native binary format be used for UEFI Drivers on ARM systems
  – Feedback?
Summary
Conclusion

• UEFI Technologies play significant roles in the ARM Platform Architecture
• ARM SBBR requires UEFI, ACPI and SMBIOS implementations
• SBSA/SBBR Tests can be used for compliance tests
• Drive closure on a remaining questions
Thanks for attending the Spring 2017 UEFI Seminar and Plugfest

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

presented by

ARM