The Next Steps in the Evolution of ARM Cortex-M

Joseph Yiu
Senior Embedded Technology Manager
CPU Group

ARM Tech Symposia China 2015
November 2015
Trust & Device Integrity from Sensor to Server
Device Security Fundamentals

Separation
- Isolate trusted resources from non-trusted
- Isolate non-trusted software
- Reduce attack surface of key components

Trusted Software
- Provision of security services
- Small, well reviewed code

Trusted Hardware
- Hardware assist for cryptography
- Secure access validation built into SoC
Bringing Security to the Smallest Devices

Tomorrow

ARMv8-M architecture
The ARM architecture for ARM® Cortex®-M processors

Provides a security foundation with TrustZone®

New AMBA® 5 AHB5 specification
Extends the security foundation through the ultra-low power SoC
ARMv8-M: Taking Embedded to the Next Level

Security

Taking TrustZone security to the smallest devices

Productivity

Making scalable software development even easier

Bringing security within reach of all developers
Introducing ARMv8-M
ARMv8-M Sub-profiles

Scalable architecture

- ARMv8-M **Baseline**:  
  - Lowest cost, smallest, ARMv8-M implementations.

- ARMv8-M **Mainline**:  
  - For general purpose microcontroller products  
  - Highly scalable  
  - Optional DSP and floating-point extensions.
## ARMv8-M Baseline Performance & Scalability

Instruction set feature uplift for baseline microcontroller

<table>
<thead>
<tr>
<th>Feature</th>
<th>Key benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware divide</strong></td>
<td>Faster integer divide operation in hardware.</td>
</tr>
<tr>
<td></td>
<td>Removes need for library code.</td>
</tr>
<tr>
<td><strong>Compare and branch</strong></td>
<td>Combined compare-with-zero and branch.</td>
</tr>
<tr>
<td></td>
<td>Faster control code.</td>
</tr>
<tr>
<td><strong>Long branch</strong></td>
<td>Long non-linking branch to compliment branch with link.</td>
</tr>
<tr>
<td></td>
<td>Enables support for cross unit tail calls.</td>
</tr>
<tr>
<td><strong>Wide immediate moves</strong></td>
<td>Pointer and large immediate creation without needing a literal load.</td>
</tr>
<tr>
<td></td>
<td>Provides a linking mechanism for execute-only code.</td>
</tr>
<tr>
<td><strong>Exclusive accesses</strong></td>
<td>Load-link / store-conditional support for semaphore use.</td>
</tr>
<tr>
<td></td>
<td>Enables common semaphore handling between CPUs.</td>
</tr>
<tr>
<td><strong>Interrupt active bits</strong></td>
<td>Active status of all interrupts individually tracked.</td>
</tr>
<tr>
<td></td>
<td>Offers dynamic re-prioritization of interrupts.</td>
</tr>
</tbody>
</table>
ARMv8-M Mainline Variants

Comprehensive instruction set support with optional DSP and floating-point extensions

- Retains Baseline fundamentals.

- Adds extensive 32-bit instruction set
  - ~ 40% performance uplift over Baseline.

- Optional integer digital signal processing (DSP) extension
  - ~ 80 saturating arithmetic and SIMD operations.

- Optional floating-point (FP) extension
  - ~ 45 instructions, IEEE754 compatible single, and/or double precision floating-point operations.
Memory Protection and Watchpoints

Improved programmability and flexibility

- ARMv8-M adopts base and limit style comparators for regions
  - Replaces previous power-of-two size, sized aligned scheme
    - Simplifies software development, encouraging creation of safer software
    - Accelerates programming, potentially reducing context switch times.

- MPU configurable down to 32-byte granularity.

- Debug variable watchpoints also enhanced to support more flexible scheme.
Introducing
ARM TrustZone for ARMv8-M
ARM TrustZone Technology

Bringing ARM security extensions to the embedded world

- Optional security extension for the ARMv8-M architecture
  - Security architecture for deeply embedded processors
  - Enables containerisation of software
  - Simplifies security assessment of embedded devices.

- Conceptually similar and compatible with existing TrustZone technology
  - New architecture tailored for embedded devices
    - Preserves low interrupt latencies of Cortex-M
    - Provides high performance cross-domain calling.
ARMv8-M Additional States

Existing handler and thread modes mirrored with secure and non-secure states

- Secure and Non-Secure code run on a single CPU
  - For efficient embedded implementation.

- Secure state for trusted code
  - New Secure stack pointers for robust operation
  - Addition of stack-limit checking.

- Dedicated resources for isolation between domains
  - Separate memory protection units for Secure and Non-secure
  - Private SysTick timer for each state.

- Secure side can configure target domain of interrupts.
ARMv8-M Interrupt Security

High-performance interrupt handling with register protection

- Subject to priority, Secure can interrupt Non-secure and vice versa
  - Secure can boost priority of own interrupts
  - Uses current stack pointer to preserve context.

- Uses ARMv7-M exception stacking mechanism
  - Hardware pushes selected registers.

- Non-secure interruption of Secure code
  - CPU pushes all registers and zeroes them
    - Removes ability for Non-secure to snoop Secure register values.
Security Defined by Address

All transactions from core and debugger checked

- All addresses are either Secure or Non-secure.

- Policing managed by Secure Attribution Unit (SAU)
  - Internal SAU similar to MPU
  - Supports use of external system-level definition
    - E.g. based on flash blocks or per peripheral.

- Banked MPU configuration
  - Independent memory protection per security state.

- Load/stores acquire NS attribute based on address
  - Non-secure access attempts to Secure address = memory fault.
High Performance Cross-Domain Calls

Efficient microcontroller focussed implementation

- Security inferred from instruction address
  - Secure memory considered to hold Secure code.

- Direct function calls across boundary
  - High performance and high security
  - Multiple entry points
  - No need to go via “monitor” for transitions.

- Uses Secure Gateway instruction “SG”
  - Only permitted in special Secure memory with Non-secure-callable attribute (NSC).
TrustZone for ARMv8-A

Secure transitions handled by the processor to maintain embedded class latency
Cross-Domain Function Calls

An assembly code level example

- **Guard instruction (SG)** polices entry point
  - Placed at the start of function callable from non-secure code.
- **Non-secure  → secure branch faults if SG isn’t at target address**
  - Can’t branch into the middle of functions
  - Can’t call internal functions.
- **Code on Non-secure side identical to existing code.**
A Simplified Use Case

Composing a system from Secure and Non-secure projects

- Non-secure project cannot access Secure resources.
- Secure project can access everything.
- Secure and Non-secure projects may implement independent time scheduling.
Microcontroller System

With TrustZone technology

- Security driven from master
  - Dynamically from an ARMv8-M CPU
  - Statically from a simple DMA.
- Propagated by AHB5 interconnect
  - Compatible with existing Cortex-A.
- Enables selective access
  - Individual flash pages
  - Regions of memory
  - Peripherals.
ARMv8-M Ecosystem Development Underway

ARMv8-M provides the standard for the extensive Cortex-M ecosystem to create the security solutions needed in a connected world

Contact us to start your ARMv8-M development
ARMv8-M: Security in Small, Real-time Embedded

- Optimised for small real-time processors
  - Low, deterministic interrupt latency
  - Efficient – every cycle counts

- Hardware based security state switch
  - No hypervisor code and processing overhead

- Fully programmable in C
  - Easy to program, easy to debug

- Transition via a standard function call
  - Transparent to the software developer

System Security

© ARM 2015
ARMv8-M: Increased Software Productivity

- Improved scalability
  - Continuum across product family

- Easier, standardised device protection
  - TrustZone security
  - Simplified MPU

- Enhanced debug
  - Improved trace
  - More flexible breakpoints/watchpoints
The Next Steps in the Evolution of Cortex-M

**ARMv8-M**

*Provides a continuum of performance and compatibility*

**ARM TrustZone Technology**

*Simplifies and accelerates security in the microcontroller space*

**AMBA 5 AHB5**

*Extends security to the system*
Thank you

- The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.
- Copyright © 2015 ARM Limited