

Objectives

- Understand ARM v8-R architecture and Cortex-R52/R52+ features.
- Learn pipeline behavior and instruction execution.
- Master exception handling and memory systems.
- Explore virtualization and safety features.
- Implement synchronization and debug techniques.

Prerequisites

- Basic knowledge of ARM architecture
- Familiarity with embedded systems
- Experience in assembler programming (optional)

Target audience

- Software developers working with ARM architecture
- System architects
- Embedded systems engineers

Environnement du cours

- Cours théorique
 - Support de cours imprimé et au format PDF (en anglais).
 - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique.
- Au début de chaque demi-journée une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Modalités d'évaluation

- Les prérequis indiqués ci-dessus sont évalués avant la formation par l'encadrement technique du stagiaire dans son entreprise, ou par le stagiaire lui-même dans le cas exceptionnel d'un stagiaire individuel.
- Les progrès des stagiaires sont évalués par des quizz proposés en fin des sections pour vérifier que les stagiaires ont assimilé les points présentés
- En fin de formation, une attestation et un certificat attestant que le stagiaire a suivi le cours avec succès.
 - En cas de problème dû à un manque de prérequis de la part du stagiaire, constaté lors de la formation, une formation différente ou complémentaire lui est proposée, en général pour conforter ses prérequis, en accord avec son responsable en entreprise le cas échéant.

Plan

First Day

ARM v8-R Architecture overview

- Core registers
- Exception model
- Instruction sets
- Memory model
- Virtualization

Introduction to Cortex-R52/R52+

- Overview
- Memory System
 - TCM Memory
 - Level-1 caches
 - Direct access to internal memory
 - AXIM interface
 - Low-latency peripheral port
 - Flash interface
 - AXIS interface
 - Error detection and handling
- Safety and Configurable options
 - DCLS
 - Spin-lock

Pipeline and ISA

- Pipeline
- Instruction execution
- Conditional instructions
- Flag-setting instructions
- Timings
 - Instructions cycle timings
 - Base instructions cycle timings
 - Pipeline Behavior
 - Skewing
 - Dual-issuing
 - Load / Store
 - Division and square root
 - Floating-point and Advanced SIMD Multiply accumulate instructions
 - Instructions with exceptional behavior
- A32 and T32 instructions

Second Day

Exceptions Model

- Exception state
- Exception levels
- Reset state in ARMv8-R
- Interrupt
 - Controller

- Handling
- Virtualization

Level 1 memory system (Cache and TCM)

- Cache
 - Cache basics: organization, replacement algorithm, write policies
 - Cache organization
 - Write with allocate policy
 - Understanding transient cache line load / store: linefill buffers, eviction buffer
 - Cache maintenance operations
- TCM
 - Tightly Coupled Memories, address decoding
 - ITCM and DTCM configuration
 - Accessing the TCMs from the AXI slave interface
 - ECC protection, TCM internal error detection and correction
 - Preloading TCMs with ECC
 - Using TCMs from reset
- Memory ordering
- Memory Barriers
- Shared Resource management

Memory Protection Unit (MPU)

- Memory protection overview
- MPU regions
- Virtualization support

Third Day

GICv3

- Introduction
- Fundamentals
- Configuring the GIC (SPI, PPI, SGI ...)
- Handling Interrupts
- Configuring LPIs
- Virtualization

ARMv8-R Virtualization with safety considerations

- Introduction to virtualization
 - Basic concepts
 - Virtual interrupts
 - Trapping exceptions
 - Trapping instructions and register access
 - PMSA (MPU)
 - Virtual timers
 - Virtual machine IDs
 - Backup registers
- Examples of virtualization
 - Guest OS switcher
 - OS monitor
 - Virtual interrupts

Synchronization overview

- Inter-Processor Interrupts
- Cluster ID

- Exclusive access monitor, implementing Boolean semaphores
- Global monitor
- Spin-lock implementation
- Using events

Performance Monitoring Unit (PMU)

- Event interface
- Counters
- Authentication signals and PMU behavior

Debug

- Debug overview (External debug, self-hosted debug ...)
- Cross trigger
- Embedded Trace Macrocell (ETM)

Renseignements pratiques

Renseignements : 3 jours