



## This course covers the Cortex-R5 / Cortex-R5F ARM cores

### Objectives

- This course is split into 3 important parts:
  - Cortex-R5 architecture
  - Cortex-R5 software implementation and debug
  - Cortex-R5 hardware implementation.
- Interaction between level 1 caches, TCM and main memory is studied through sequences.
- The course explains how to assign access permissions and attributes to regions by using the MPU.
- The exception mechanism is detailed, indicating how the VIC port can contribute to reduce interrupt latency.
- Sequences involving memory, cache and external master are used to explain the benefits of the ACP port.
- The course also details the hardware implementation and provides some guidelines to design a SoC based on Cortex-R5.
- An overview of the Coresight specification is provided prior to describing the debug related units.

*Labs are run under RVDS*

*A more detailed course description is available on request at [formation@ac6-formation.com](mailto:formation@ac6-formation.com)*

### Prerequisites and related courses

- Basic knowledge of the ARM architecture.
- Assembly-level programming notions

### Course material

- Printed training material is given to attendees during training.
- Precise and easy to use, it can be used as a reference afterwards.

### Environnement du cours

- Cours théorique
  - Support de cours au format PDF (en anglais) et une version imprimée lors des sessions en présentiel
  - Cours dispensé via le système de visioconférence Teams (si à distance)
  - 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 BASICS**

- States and modes
- Benefit of register banking
- Exception mechanism
- Instruction sets

#### **INTRODUCTION TO CORTEX-R5**

- Slave and master AXI ports
- Highlighting the new features with regard to Cortex-R4
- ARMv7-R architecture
- Exceptions
- System control coprocessor
- Configurable options
- Redundant CPU vs Twin-CPU

#### **INSTRUCTION PIPELINE**

- Prefetch unit
- Instruction cycle timing and interlock behavior
- Dynamic branch prediction mechanism: global history buffer
- Data Processing Unit
- Limited dual-issuing
- Global History Buffer
- Return stack

#### **EXCLUSIVE RESOURCE MANAGEMENT**

- Explaining issues when several processors share an exclusive resource
- Software aspects, load / store exclusive instructions
- Integrated local monitor
- Hardware aspects
- Using events to avoid to consume power while waiting for resource release

#### **MEMORY TYPES**

- Device and normal memory ordering
- Memory type access restrictions
- Access order
- Memory barriers

#### **MEMORY PROTECTION UNIT**

- ARM v7 PMSA
- Default memory map
- Cortex-R5 MPU and bus faults
- Region overview
- Setting up the MPU

## EXCEPTION MANAGEMENT

- Low Interrupt Latency
- Primecell VIC PL192
- VIC basic signal timing
- Connectivity: daisy-chained VIC
- Interrupt priority and masking
- Determining the cause of the fault through CP15 status registers
- Precise vs imprecise faults

## Second day

## LEVEL 1 MEMORY SYSTEM

- Cache basics
- Tag RAM and Data RAM organization
- Handling cache parity / ECC errors
- Cache maintenance operations
- Tightly Coupled Memories
- ECC/parity protection
- Preloading TCMs with ECC
- Using TCMs from reset
- Store buffer, merging data

## CACHE COHERENCY

- Hardware coherency vs software coherency
- ACP pass through interface,
- Virtual AXI peripheral interface region
- DMA into TCM
- Highlighting the difference between the  $\mu$ SCU and the Cortex-A SCU

## AXI PROTOCOL

- PL301 AXI interconnect
- AXI channels, channel handshake
- Transaction ordering
- Read and write burst timing diagrams
- AXI master interface attributes
- Write merging example
- AXI slave interfaces attributes
- Peripheral interfaces port attributes
- Accelerator Coherency Port interface
- Controlling an external cache

## HARDWARE IMPLEMENTATION

- Clock domains
- Reset domains
- Power control
- Maintaining caches and TCM powered while turning off the pipeline: dormant mode
- Power mode interaction with ACP
- Debugging the processor while powered down

**Third day****APB - ADVANCED PERIPHERAL BUS**

- Second-level address decoding
- Pinout
- APB3.0 new features

**PERFORMANCE MONITOR**

- Event counting
- Related interrupts
- Debugging a multi-core system with the assistance of the PMU

**LOW POWER MODES**

- Voltage domains
- Run mode, standby mode, dormant mode
- Communication to the power management controller

**CORESIGHT DEBUG UNITS**

- Benefits of CoreSight
- Invasive debug, non-invasive debug
- APBv3 debug interface
- Connection to the Debug Access Port
- Process related breakpoint and watchpoint
- Debug Communication Channel
- ETM interface
- Cross-Trigger Interface
- Debugging systems with energy management capabilities

**THUMB-2 INSTRUCTION SET (V7-A)**

- Introduction
- General points on syntax
- Data processing instructions
- Branch and control flow instructions
- Memory access instructions
- If&then conditional blocks
- Stack in operation
- Exclusive load and store instructions
- Memory barriers and synchronization
- Interworking ARM and Thumb states

**Renseignements pratiques**

**Renseignements : 3 jours**