# RA2 - Cortex-A9 implementation

*This course covers both Cortex-A9 single and multiple core high-end ARM CPUs*

## Objectives

- This course is split into 3 important parts:
  - Cortex-A9(MP) architecture
  - Cortex-A9(MP) software implementation and debug
  - Cortex-A9(MP) hardware implementation
- MMU operation under Linux is described.
- Spin-lock implementation in a multicore system is also detailed
- Interaction between level 1 caches, level 2 cache and main memory is studied through sequences.
- The exception mechanism is explained, indicating how virtualization enables the support of several operating systems.
- An overview of the Coresight specification is provided prior to describing the debug related units.
- The course also describes the hardware implementation and provides some guidelines to design a SoC based on Cortex-A9.
- Cache coherency is detailed, including cache tag mirrors, the advantage of connecting DMA channels to ACP and the sequences that have to be used to modify a page descriptor.

A more detailed course description is available on request at training@ac6-training.com

## Prerequisites and related courses

- Knowledge of ARM7/9 or having attended our course ARM fundamentals.
- This course does not include chapters on low level programming.
- Related courses:
  - VFP programming, RC0 - VFP programmingcourse
  - NEON programming, RC1 - NEON-v7 programmingcourse

    .

## Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

## Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

# Course Outline

## First day

### INTRODUCTION TO CORTEX-A9

- Cortex-A9 variants
- New memory-mapped registers in MPCore
- The 3 instruction sets
- Configurable options

### ARM BASICS

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

### INSTRUCTION PIPELINE

- Superscalar pipeline operation
- Branch prediction mechanism
- Return stack
- Predicted and non-predicted instructions

### TRUSTZONE

- Secure to non secure permitted transitions
- L1 and L2 secure state indicators, memory partitioning
- Interrupt management when there is a mix of secure and non-secure interrupt sources
- Boot sequence

### INTRODUCTION TO MULTI-CORE SYSTEMS

- AMP vs SMP
- Boot sequence
- Exclusive access monitor
- Spin-lock implementation
- Using events
- Basic concepts of RTOS supporting A9 SMP architecture

## Second day

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

- General points on syntax
- Branch and control flow instructions
- Memory access instructions
- Exception generating instructions
- If&then conditional blocks
- Interworking ARM and Thumb states
- Demonstration of assembly sequences aimed to understand this new instruction set

## MEMORY MANAGEMENT UNIT

- Page access permission, domain and page protection
- Page attributes, memory types
- Utilization of memory barrier instructions
- Format of the external page descriptor table
- TLB lockdown
- Abort exception, on-demand page mechanism
- MMU maintenance operations
- Using a common page descriptor table in an SMP platform, maintaining coherency of multiple TLBs

## LEVEL 1 MEMORY SYSTEM

- Virtual indexing, physical tagging for instruction cache
- Supported maintenance operations
- Write-back write allocate cache allocation
- Memory hint instructions PLD, PLI, PLDW, data prefetching
- Describing transient cache related transactions: line fills and line eviction
- 4-entry 64-bit merging store buffer

## HARDWARE COHERENCY

- Snooping basics: CLEAN, CLEAN & INVALIDATE and INVALIDATE snoop requests
- Snoop Control Unit: cache-to-cache transfers
- MOESI state machine
- Understanding through sequences how data coherency is maintained between L2 memory and L1 caches
- Accelerator Coherency Port

## AMBA 3

- AXI
  - Topology: direct connection, multi-master, multi-layer
  - PL301 AXI interconnect
  - Separate address/control and data phases
  - AXI channels, channel handshake
  - Transaction ordering
  - Read and write burst timing diagrams
  - Cortex-A9 external memory interface, ID encoding
- APB 3

## HARDWARE IMPLEMENTATION

- Clock domains
- Reset domains
- Wait For Interrupt architecture
- AXI master interface attributes
- Exclusive L2 cache
- AXI sideband information

## PL310 LEVEL 2 CACHE

- AXI interface characteristics
- Exclusive mode operation
- Understanding through sequences how cacheable information is copied from memory to level 1 and level 2 caches
- TrustZone support
- Power management
- Cache event monitoring

- Describing each maintenance operation
- Cache lockdown
- Interrupt management

## PERFORMANCE MONITOR

- Event counting
- Selecting the event to be counted for the 6 counters
- Debugging a multi-core system with the assistance of the PMU

# Fourth day

## INTERRUPT CONTROLLER

- Cortex-A9 exception management
- Interrupt virtualization
- Integrated timer and watchdog unit in MPCore
- Interrupt groups: STI, PPI, SPI, LSPI
- Legacy mode
- Prioritization of the interrupt sources
- Distribution of the interrupts to the Cortex-A9 cores
- Detailing the interrupt sequence
- Spurious interrupt

## LOW POWER MODES

- Voltage domains
- Cortex-A9 power control
- Communication to the power management controller
- SCU power status register

## CORESIGHT DEBUG UNITS

- Invasive debug, non-invasive debug
- APBv3 debug interface
- Connection to the Debug Access Port
- Process related breakpoint and watchpoint
- Program counter sampling
- Event catching
- Debug Communication Channel
- PTM interface, connection to funnel
- Debug registers description
- Cross-Trigger Interface, debugging a multi-core SoC