



RA0 - Cortex-A5 implementation

This course covers the ARM Cortex-A5 CPU

OBJECTIVES

- This course is split into 3 important parts:
 - Cortex-A5 architecture
 - Cortex-A5 software implementation and debug
 - Cortex-A5 hardware implementation.
- MMU operation under Linux is described.
- Interaction between level 1 caches, level 2 cache and main memory is studied through sequences.
- The exception mechanism is detailed, indicating how virtualization enables the support of several operating systems.
- The course also details the hardware implementation and provides some guidelines to design a SoC based on Cortex-A5.
- An overview of the Coresight specification is provided prior to describing the debug related units.
- The course explains the mechanisms dedicated to SMP implementation, exclusive resource management, snooping, software generated interrupt.

Labs are run under RVDS

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

PREREQUISITES

- Knowledge of ARM7/9 or having attended our course ARM fundamentals.
- This course does not include chapters on low level programming.
 - ACSYS offers a large set of tutorials to become familiar with RVDS, assembly level programming, compiler hints and tips.
- More than 12 correct answers to our Cortex-A5 prerequisites questionnaire.

Plan

First day

INTRODUCTION TO CORTEX-A5

- Cortex-A5 variants: single core vs multicore
- The 4 instruction sets

- Configurable options

ARM BASICS

- States and modes
- Exception mechanism
- Instruction sets

INSTRUCTION PIPELINE

- In-order pipeline operation
- Branch prediction mechanism
- Return stack

TRUSTZONE

- TrustZone conceptual view
- Secure to non secure permitted transitions
- Memory partitioning
- Interrupt management
- Boot sequence

INTRODUCTION TO MULTI-CORE SYSTEMS

- AMP vs SMP
- Boot sequence
- Exclusive access monitor
- Global monitor
- Spin-lock implementation
- Using events
- Basic concepts of RTOS supporting A5 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
- Stack in operation
- Interworking ARM and Thumb states
 - Demonstration of assembly sequences aimed to understand this new instruction set

MEMORY MANAGEMENT UNIT

- MMU objectives
- Address translation
- Page access permission, domain and page protection
- Utilization of memory barrier instructions
- Format of the external page descriptor table
- Tablewalk
- TLB organization
- Utilization of microTLBs
- Abort exception, on-demand page mechanism

- MMU maintenance operations
- Maintaining coherency of multiple TLBs

LEVEL 1 MEMORY SYSTEM

- Cache organization
- Supported maintenance operations
- Memory hint instructions
- Describing transient cache related transactions: line fills and line eviction
- 64-bit merging store buffer
- PMU related events

HARDWARE COHERENCY

- Snooping basics
- Snoop Control Unit: cache-to-cache transfers
- MOESI state machine
- Address filtering
- Understanding through sequences how data coherency is maintained between L2 memory and L1 caches
- Accelerator Coherency Port
- Enabling coherency mode

Third day

AMBA 3

- AXI
 - Topology
 - PL301 AXI interconnect
 - AXI channels, channel handshake
 - Support for unaligned data transfers
 - Transaction ordering, out of order transaction completion
 - Cortex-A5 external memory interface, ID encoding
- APB 3

HARDWARE IMPLEMENTATION

- Clock domains
- Reset domains
- Power control, dynamic power management
- Wait For Interrupt architecture
- Level 2 memory interface
- Exclusive L2 cache

PL310 LEVEL 2 CACHE

- Cache configurability
- Understanding through sequences how cacheable information is copied from memory to level 1 and level 2 caches
- Transient operations, utilization of line buffers LFBs, LRBs, EBs and STBs
- Cache event monitoring
- Describing each maintenance operation
- Cache lockdown
- Initialization sequence

PERFORMANCE MONITOR

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

Fourth day

INTERRUPT CONTROLLER

- Cortex-A5 exception management
- Interrupt virtualization
- Integrated timer and watchdog unit in MPCore
- Interrupt groups: SGI, PPI, SPI, LSPI
- Prioritization of the interrupt sources
- Distribution of the interrupts to the Cortex-A5 cores
- Generation of interrupts by software
- Detailing the interrupt sequence, purpose of Interrupt Acknowledge register and End-Of-Interrupt register

LOW POWER MODES

- Voltage domains
- Communication to the power management controller
- Standby and wait for event signals
- SCU power status register

CORESIGHT DEBUG UNITS

- Benefits of CoreSight
- Invasive debug, non-invasive debug, taking into account the secure attribute
- Connection to the Debug Access Port
- Debug facilities offered by Cortex-A5
- Event catching
- Debug Communication Channel
- ETM interface
- Cross-Trigger Interface, debugging a multi-core SoC

Renseignements pratiques

Duration : 4 days
Cost : 1950 € HT

