



## RM1 - Cortex-M1 implementation

*This course covers the Cortex-M1 ARM core targeting FPGA SoCs*

### Objectives

- This course is split into 3 important parts:
  - Processor architecture
  - Software implementation
  - Hardware implementation.
- A tutorial has been developed by ACSYS to facilitate the understanding of Cortex-M1 low level programming, therefore labs can be replayed after the course.
- The course explains how to design a SoC based on Cortex-M1, clarifying the operation of the interconnect and the debug facilities integrated in the CPU.

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

### Prerequisites

- Knowledge of ARM7/9.
- 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 Cortex-R prerequisites questionnaire.

### 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

#### ARM Cortex-M1 INTRODUCTION

- Programmer's model
- Fixed memory map
- Privilege, modes and stacks
- Memory Protection Unit
- Interrupt handling
- Nested Vectored Interrupt Controller [NVIC]
- Power management
- Debug

## ARM Cortex-M1 CORE

- Datapath and pipeline
- Write buffer
- Bit-banding
- System timer
- State, privilege and stacks
- System control block
- Different level of debug implementation

## EXCEPTIONS

- Exception behavior, exception return
- Non-maskable exceptions
- Privilege, modes and stacks
- Fault escalation
- Vector table

## OVERVIEW OF THUMB-2 INSTRUCTION SET

- Data processing instructions
- Branch and control flow instructions
- Memory access instructions
- Exception generating instructions
- If...then conditional blocks
- Exclusive load and store instructions
- Accessing special registers
- Memory barriers and synchronization

## Second day

## INTERRUPTS

- Interrupt entry / exit, timing diagrams
- Tail chaining
- Interrupt response, pre-emption
- Interrupt prioritisation
- Interrupt implementation configurability, impact on core size

## MEMORY TYPES

- Memory types, restriction regarding load / store multiple
- Device and normal memory ordering
- Access order
- Memory barriers

## INVASIVE DEBUG

- Cortex-M1 debug features
- Monitor mode
- Flash patch and breakpoint features
- Data watchpoint and trace
- DWT registers
- AHB-Access Port

## INTEGRATION

- Functional Integration
- Clocking
- Reset
- AHD and Debug interfaces
- Synthesis, Place and Route
- Sign-Off

## Third day

## IMPLEMENTATION

- Implementation flow
- Configuration options
- RTL Validation
- Synthesis
- Place and route
- Qualification

## AMBA3.0 INTERCONNECT SPECIFICATION

- Purpose of this specification
- Example of SoC based on AMBA specification
- Differences between AMBA2.0 and AMBA3.0

## AHB - ADVANCED HIGH PERFORMANCE BUS

- Centralized address decoding
- Address gating logic
- Arbitration, bus parking
- Single-data transactions
- Sequential transfers
- Retry response
- Split response
- AHB-lite specification

## APB - ADVANCED PERIPHERAL BUS

- Read timing diagram
- Write timing diagram
- Operation of the AHB-to-APB bridge
- APB3.0 new features

## AHB CORTEX-M1 PORTS

- Clocking and reset
- Bus interfaces , AMBA-3 compliance
- Debug interface, AHB-AP programming interface
- Connection to the TPIU