ARM CORTEX-M1 SYSTEM DESIGN

Ref : 004735A **Duration : 4 days**

OBJECTIVES

- This course takes an in depth look at the considerations you will need to take into account when designing a system containing a Cortex-M1 processor core
- It is aimed at :
  - Software engineers who not only want to obtain details of how to write software to run on the Cortex-M1, but also wish to obtain an understanding of hardware design issues
  - Hardware engineers who need to understand how to design Cortex-M1 based systems, but also wish to obtain an understanding of the issues of writing software to run on that system

RELATED COURSES

- Formation 1 (référence)
- Formation 2 (référence)

PARTNERS

- This training course is approved by ARM

PREREQUISITES

- A basic understanding of microprocessors and microcontrollers is recommanded
- A basic understanding of digital logic or hardware / ASIC design issues would be useful but not essential
- A basic understanding of assembler or C programming would be useful but not essential
- A basic awareness ARM cores is useful but not essential

PRACTICAL LABS

- Las can be run under 2 possible environments : Eclipse/RVDS or Keil IDE

APPROVED TRAINING CENTRE ARM

Contact

Tel : +33 (0)5 62 13 52 32
Fax : +33 (0)5 61 06 72 60
training@mvd-fpga.com

Course also available customized

Next sessions, see : *http://www.mvd-fpga.com/en/formationsCalend.html*

TOPICS

ARM Cortex-M1 CORPORATE INTRODUCTION

- ARM architectural summary
- Architecture challenge, meeting the challenge with profiles
- ARM instruction set evolution
-

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

DEVELOPMENT TOOLS OVERVIEW

- RVCT compilation tools
- Codewarrior for RVDS
- RVD debugger
- RVISS simulator
- JTAG run control unit

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

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
- Workbook : introductory tutorial for Cortex-M1

INTERRUPTS

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

EXCEPTIONS

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

MEMORY TYPES

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

EMBEDDED SOFTWARE DEVELOPMENT WITH Cortex-M1

- Placing code, data, stack and heap in the memory map, scatterloading
- Tailoring the C library to your target
- Reset and initialisation
- Building and debugging your image

- Long branch veneers
- Workbook : Retargeting the standard C library functions, handling interrupts

INVASIVE DEBUG

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

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

INTEGRATION

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

IMPLEMENTATION

- Implementation flow
- Configuration options
- RTL Validation

DOCUMENTATION

Training manuals will be given to attendees during training in print.