R1 - ARM7/9 implementation

This course covers ARM7TDMI and ARM966/946/926 cores.

Objectives

- This course takes an in depth look at the considerations you will need to take into account when designing a system containing either an ARM7TDMI family or ARM9TDMI family processor core.
- Some information on the latest generation of ARM processor cores, such as the ARM9E-S family is also included.
- It is aimed at:
  - Software engineers who not only want to obtain details of how to write software to run on the ARM, but also wish to obtain an understanding of hardware design issues
  - Hardware engineers who need to understand how to design ARM based systems, but also wish to obtain an understanding of the issues of writing software to run on that system.

For on-site courses, labs can be run under 3 possible environments : CodeWarrior/ADS/AXD, Eclipse/RVDS or GNU/Lauterbach simulator.
For open courses, labs are run under Eclipse/RVDS.
A more detailed course description is available on request at training@ac6-training.com

Prerequisites

- A basic understanding of microprocessors and microcontrollers.
- 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 of the ARM is useful but not essential.

Environnement du cours

- Cours théorique
  - Support PDF (en anglais), imprimé en présentiel ; à distance via Teams.
  - Assistance du formateur tout au long de la formation.
- Chaque session débute par un point avec les stagiaires.

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Plan du cours

First day

The ARM architecture

- ARM operation modes
- The ARM registers set
- Program Status Registers
- Exception handling
- Instruction sets

ARM processor core

- ARM7TDMI core signals
- The ARM7TDMI instruction pipeline
- ARM7TDMI memory interface
- ARM9TDMI datapaths
- ARM9TDMI pipeline

Second day

ARM and Thumb instruction sets

- Conditional execution and flags
- Branch instructions
- Single register data transfer
- Block data transfer
- Stack management
- Register access in Thumb
- ARM architecture V5TE new instructions

ARM / Thumb interworking

- Branch exchange example
- Mixing ARM and Thumb subroutines
- ARM to thumb veneer
- Thumb-to-ARM veneer
- Interworking calls

Exception handling

- Exception priority
- Vector table instructions
- Chaining exception handlers
- Register usage in exception handlers
- Example C interrupt handler
- Software managed interrupt controller
- Issues when reenabling interrupts
- Invoking SWIs
- Data abort with memory management

Third day

Compiler hints and tips

- Automatic optimization
- Instruction scheduling
- Tail-call optimization
- Loop termination
- Inline assembler
- Stack usage
- Global data layout

Initializing cached processors

- Cache basics
- Cache flushing
- Write buffer
- Memory management
- TLB and translation tables

- Memory protection, MPU configuration steps
- System control coprocessor
- Tightly coupled memory

Embedded sofware development

- ROM/RAM remapping
- Exception vector table
- Reset handler
- C library initialization
- Scatterloading
- Linker placement rules
- Long branch veneers
- C library functionality
- Placing the stack and heap

Fourth day

AMBA bus architecture

- AHB Protocol
- AHB Connection Architectures
- AHB Workbook

ARM debug solutions

- Debugging with multiICE
- Watchpoints, hardware breakpoints, software breakpoints
- Semihosting
- EmbeddedICE-RTT logic
- Instruction trace, data trace
- Trace capture