This course describe the Microcontroller Subsystem (MSS) of SmartFusion2 Microchip FPGAs

Objective

- Get an Overview on the Cortex-M Architecture
- Understand the Cortex-M Software implementation and debug
- The Microchip Implementation and Embedded Development Kit (EDK) with Libero SoC and software Integrated Development environment (IDE) tools are described to create a hardware platform and the software to execute to programme it
- Describe SmartFusion MSS Architecture, I/Os and understand the SmartFusion2 FPGA Fabric Interface
- Become familiar with the MSS Peripherals

Prerequisites

- Basic knowledge of processor and FPGA technology
- Knowledge of VHDL and C languages

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
    - One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
    - An installation and test manual is provided to allow preinstallation of the needed software.
    - The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- 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.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented

- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

First Day

Libero SoC

- Microchip FPGA & SoC overview
- Libero SoC overview
- Create and Design
- Constraint management
- TestBench and Simulations
- Programme and Debug
- Microchip (Microsemi) tool’s

Exercise: *Creating a Libero Project for Firmware*
Exercise: *Libero Project and SoftConsole*

Cortex-M3 Architecture Overview

- V7-M Architecture Overview
- Core Architecture
  - Harvard Architecture, I-Code, D-Code and System Bus
  - Write Buffer
  - Registers (Two stack pointers)
  - States
  - Different Running-mode and Privileged Levels
  - System Control Block
  - Systick Timer
  - MPU Overview
- Programming
  - Alignment and Endianness
  - CMSIS Library
- Exception/ Interrupts Mechanism Overview
  - Vector Table
  - Interrupt entry and return Overview
  - Tail-Chaining
  - Pre-emption (Nesting)
  - NVIC Integrated Interrupt Controller
  - Exception Priority Management
  - Fault escalation
  - Debug Interface

Second Day

SmartFusion2 Cortex-M3 Processor

- System Level Interface
- Integrated Configurable Debug
- Cortex-M3 Processor Core Peripherals
  - Nested Vectored Interrupt Controller
  - System Control Block
  - System Timer
  - Memory Protection Unit

- Cortex-M3 Processor Description
  - Programmers Model
  - Memory Model
  - Exception Model
  - Fault Handling
  - Power Management
- Cortex-M3 Processor Instruction Set
  - CMSIS Functions
  - Memory Access Instructions
  - General Data Processing Instructions
  - Saturating Instructions
  - Branch and Control Instructions
- Cortex-M3 Processor Peripherals
  - System Control Block
  - System Timer (Systick)
  - Memory Protection Unit

Exercise: *Cortex-M3 Mode Privilege*
Exercise: *Cortex-M3 Exception Management*
Exercise: *Cortex-M3 MPU*
Exercise: *Cortex-M3 Real-Time Operating System (FreeRTOS)*

MSS Cache Controller

- Cache Matrix
- Memory Mapping
- Memory Maps and Transaction Mapping
- Cache Locked Mode
- How to use cache Controller

Exercise: *Cache Controller Configuration*

Embedded NVM (eNVM) and SRAM (eSRAM) Controllers

- Functional Description
- Security
- How to use eNVM
- SYSREG Control Registers

High Performance DMA Peripheral and Controller

- DMA Controller Initialization
- DMA Controller Operations
- How to use HPDMA
- HDMA Controller Register Map
- Peripheral DMA Architecture Overview
- How to use PDMA
- PDMA Register Map

Fabric Interface Controller

- Architecture Overview
- Functional Description
- AHB-Lite Options
- Implementation Considerations
- How to Use FIC

Third Day

Reset Controller

- Power-On Reset Generation Sequence
- Power-Up to functional Time Data
- CoreResetP Soft Reset Controller
  - Reset Topology
  - Implementation
- How to use the Reset Controller

AHB - Advanced High Performance Bus

- Centralized address decoding
- Address gating logic
- Address pipelining
- Sequential transfers
- AHB-Lite Specification

Connectivity and Communication

- Universal Serial Bus OTG Controller
- Ethernet MAC
- CAN Controller
- MMUART Peripherals
- Serial Peripheral Interface Controller (SPI)
- Inter-Integrated Circuit Peripherals (I^2C)
- MSS GPIO
- Communication Block
  - Architecture Overview
  - CoreSysServices Soft IP
  - How to use the communication block
- Real-Time Controller
- System Timer
- Watchdog Timer
- ECC System Service

Exercise: *Programming using UART interface*
Exercise: *MSS CAN drivers and APIs*
Exercise: *Programming using USB OTG Controller Interface*
Exercise: *Using ECC System Service*
Exercise: *FreeRTOS and LWIP Project*

Renseignements pratiques

Inquiry : 3 days