This course explains how to design a SoC based on MicroBlaze, AMD proprietary IPs and/or custom IPs using EDK

Objectives

- The course describes how to build a complete Embedded System based on MicroBlaze Xilinx Processor
- Microblaze Implementation and Embedded Development Kit (EDK) with Xilinx Platform Studio (XPS) and Software Development Kit (SDK) tools are described to create a hardware platform and the software to execute to program it
- Xilinx Simulation tools are presented to debug Software and Hardware
- Impact utility is described to flash the bitstream
- Chipscope Pro is also used in order to display on-chip AXI transactions

Environment

- A PC with Xilinx Vivado v.2013.4 IDE
- Nexys-3 (Xilinx Spartan6-based) board or Nexys-4 (Xilinx Artix7-based) board
- Training manuals in PDF format (and print format for face-to-face trainings)

Pre-requisites

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

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.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - For remote trainings:
    - One Online Linux PC per trainee for the practical activities.
    - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
    - QEMU Emulated board or physical board connected to the online PC (depending on the course).
    - Some Labs may be completed between sessions and are checked by the trainer on the next session.
  - For face-to-face trainings:
    - 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

Starting with a simple "Hello Word" Project

- Tools Introduction :
  - The Vivado Design Suite
  - EDK
- Development flow introduction
  - Hardware Development
  - Software Development
  - Verification (Simulation and Debug)
  - Flashing with Impact

Exercise: *Creating the Hardware and Software to send strings on a serial port*

The MicroBlaze Logicore IP (MCS)

- Microcontroller system
  - Features
  - Overview
  - Designing with the core
- Microblaze architecture overview
  - Core block overview
  - Introduction and configurable features
  - Data type and Endianness
  - Register Bank (General Purpose Registers and Special Registers)
  - Pipeline
  - Branches
  - Privilege Modes
  - Hardware Exception
  - Breaks
  - Interrupts
  - Vector Table
  - Floating Point Unit
  - Caches and MMU Overview
  - Debug
  - MicroBlaze ABI
- MicroBlaze Interface Overview
  - MicroBlaze I/O Overview
  - AXI4 bus, AXI-Lite
  - AXI-Stream

- PLB (Peripheral Local Bus)
- LMB (Local Memory Bus)
- FSL (Fast Simplex Link Interface)
- XCL (Xilinx CacheLink Interface)
- Debug Interface
- MicroBlaze Core Configurability

Exercise: *Looking at the MicroBlaze core configuration wizard*
Exercise: *Area size analysis on different systems (with/without caches, AXI/PLB)*

Hardware Design- Xilinx Platform Studio (XPS)

- Introduction
- Platgen
- Simgen
  - Generating the Simulation Models and the Testbench
  - Loading an application
- XPS Interface Description
- MHS, MPD, PAO files Description
- Accessing peripheral information (Datasheet, MPD, etc.)
- Bus and Periphral Connexion
- Peripheral definition files
- Platgen tool

Exercise: *Enhancing the "Hello World" Platform (Adding Interrupt Controller, Timer, GPIO, RAM)*
- *Working with bus interfaces, Ports and Memory mapping*

Exercise: *Simulating the platform with ISim through ISE Project Navigator*
- *Analysing the Instruction fetching on the Instruction AXI bus*

Software Design- The Software Development Kit (SDK)

- Introduction
- Libgen
- Scanned repositories for software component
- Startup
- Linker Script
- MSS, MLD, MDD files description
- Board Support Package Specification
- Using Xilinx IP Drivers
- Interrupt Management
- Block RAM initialization (Data2Mem)
- Debugging (Memory, Registers, Disassembly views, etc.)
- Generating the Linker Script (Placing Code and Data in different memories)
- Executing the application from an External DDR3 Memory (using a bootloader)
- Profiling
  - Rebuilding the application and the BSP with the correct compiler flags to use profiling

Exercise: *Developing the software for the hardware platform created with XPS*
- *Timer Implementation*
- *Interrupt management*
- *Creating a Blinky*
- *Generating Interrupts from GPIOs*
- *Placing the application code/data in different memories*
- *Using Profiling*

Custom Peripheral Creation and Insertion

- Create or Import Peripheral (CIP) Wizard Introduction
- LogiCORE IP AXI Lite IPIF to connect a slave peripheral on the bus
- Creating HDL and MPD templates through XPS
- Directory Structure generated
- IP development and Simulation through ISE

  - Developing a user IP and a testbench
  - Understanding how to create software registers (memory mapped)
  - Simulating the IP
- Importing a Custom IP to XPS
- Connecting a Slave IP on the AXI-Lite Bus

Exercise: *Developing and Simulating a custom IP (controlling LED intensity with a PWM) through ISE*
Exercise: *Using the CIP wizard to implement this user IP into our platform (SoC)*
Exercise: *Developing the software application to program the IP through SDK*

The ChipScope Hardware Debugger

- Introduction to Chipscope Pro
- Implementing an AXI monitor into the design to analyze AXI4-Lite Bus transactions
- Retrieving the on-chip signals waveforms using Chipscope Pro Analyzer
- Clarifying trigger conditions

Exercise: *Connecting a Chipscope Analyzer to the AXI bus on our custom IP side and, using it to measure bandwidths*

Renseignements pratiques

Inquiry : 2 days