

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

Environnement du cours

- Cours théorique
 - Support de cours au format PDF (en anglais) et une version imprimée lors des sessions en présentiel
 - Cours dispensé via le système de visioconférence Teams (si à distance)
 - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
 - Les activités pratiques représentent de 40% à 50% de la durée du cours
 - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
 - Exemples de code, exercices et solutions
 - Pour les formations à distance:
 - ▶ Un PC Linux en ligne par stagiaire pour les activités pratiques, avec tous les logiciels nécessaires préinstallés.
 - ▶ Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique
 - ▶ Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante.
 - Pour les formations en présentiel:
 - ▶ Un PC (Linux ou Windows) pour les activités pratiques avec, si approprié, une carte cible embarquée.
 - ▶ Un PC par binôme de stagiaires s'il y a plus de 6 stagiaires.
 - Pour les formations sur site:
 - ▶ Un manuel d'installation est fourni pour permettre de préinstaller les logiciels nécessaires.
 - ▶ Le formateur vient avec les cartes cible nécessaires (et les remporte à la fin de la formation).
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours

- Au début de chaque session (demi-journée en présentiel) une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

Audience visée

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

Modalités d'évaluation

- Les prérequis indiqués ci-dessus sont évalués avant la formation par l'encadrement technique du stagiaire dans son entreprise, ou par le stagiaire lui-même dans le cas exceptionnel d'un stagiaire individuel.
- Les progrès des stagiaires sont évalués de deux façons différentes, suivant le cours:
 - Pour les cours se prêtant à des exercices pratiques, les résultats des exercices sont vérifiés par le formateur, qui aide si nécessaire les stagiaires à les réaliser en apportant des précisions supplémentaires.
 - Des quizz sont proposés en fin des sections ne comportant pas d'exercices pratiques pour vérifier que les stagiaires ont assimilé les points présentés
- En fin de formation, chaque stagiaire reçoit une attestation et un certificat attestant qu'il a suivi le cours avec succès.
 - En cas de problème dû à un manque de prérequis de la part du stagiaire, constaté lors de la formation, une formation différente ou complémentaire lui est proposée, en général pour conforter ses prérequis, en accord avec son responsable en entreprise le cas échéant.

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 Peripheral 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

- *Analyzing 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

Renseignements : 2 jours