



HX5 - AMD Zynq All Programmable SoC: Hardware and Software Design

This course explains how to design a System on Chip (SoC) based on the AMD Zynq-7000 All Programmable SoC

Objectives

- Describing how to build a complete Embedded System based on the Zynq (Processing System with an ARM Cortex-A9MP Core + FPGA)
- Describing the Zynq Implementation, the Vivado IP Integrator tool and the Software Development Kit (SDK) tools to create a hardware platform and the software to program it
- Working with AMD (Xilinx) tools like Chipscope and the SDK Remote Debugging to debug the Software and the Hardware
- Booting the Linux Kernel on the platform and Executing Linux OS based Applications
- Creating a User IP and the corresponding Linux Driver and integrating it to the System

Prerequisites

- 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 ramène 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.

Plan du cours

Zynq Device Overview

- Processing System
- Programmable Logic
- Interfacing and signals
- Interconnects
- Memory
- Interrupts

Embedded System Design: Starting with a simple « Hello world » project

- Tools Introduction :
 - Vivado Design Suite
 - SDK
- Development Flow Introduction
 - Hardware Development
 - Software Development
 - Verification (Simulation and Debug)
 - Downloading the bitstream

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

Embedded System Design Using the PS and the Programmable Logic

- Adding an existing IP (from the AMD Xilinx Library) to the design
- Dealing with interrupts
- Developing with SDK
- Debugging with SDK
- Software profiling

Exercise : Enhancing the previous Platform (Adding Interrupt Controller, GPIO, RAM)

Exercise : Developing the software dealing with interrupts

Chipscope - Hardware Debug

- 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

Linux Booting and Application debugging

- The linux kernel
- Linux booting, boot Methods
- Linux OS based application software

Exercise : Linux Booting through different methods

Exercise : Debugging a Linux application using SDK Remote profiling

System Design with a DMA and the Processing System High Performance Slave Port

- Integrating the AXI CDMA
- Standalone Application
- Linux OS based Application

Exercise : Running a Standalone CDMA Application

Exercise : Running a Linux CDMA Application

Custom Peripheral (IP) Creation and Insertion

- Creating a Peripheral IP
- Importing the Peripheral
- Linux Base Device Driver Development
- Loading Module into running kernel
- Application execution

Exercise : Creating our own Intellectual Property and Device Driver for Linux OS; and executing the application