



## Logique Programmable

### Programmation et mise en oeuvre d'électronique programmable

La réalisation de systèmes embarqués combine de plus en plus de l'électronique programmable et du logiciel. Ces deux composantes du système contribuent de façon critique au bon fonctionnement du système et doivent être conçues et maîtrisées non seulement séparément mais également dans leurs interactions. **ac6-formation** propose des formations pratiques pour vous permettre de maîtriser la mise en oeuvre de composants de logique programmable et leur interactions avec la composante logicielle de vos systèmes.

Vous pouvez visualiser les descriptifs détaillés des différents cours en utilisant la barre de navigation ci-dessus. Vous pouvez également cliquer sur les références des cours dans les descriptions ci-dessous.

**V0 - Programmable component fundamentals** This training is intended to professional who want to use or maintain programmable components

This training is intended to professional who want to use or maintain programmable components

**ALT1 - CYCLONE-V CORTEX-A9 HARD PROCESSOR SYSTEM** This course covers the hard IPs present in Cyclone-V Altera FPGA family, based on ARM Cortex-A9 CPU

**H1 - Lattice Mico32 - Processeur embarqué** Mise en oeuvre et programmation d'un processeur Soft-Core dans un FPGA

De plus en plus d'applications embarquées sont basées sur des FPGA de grande capacité, et intègrent une partie logicielle nécessitant d'installer un processeur dans le FPGA. Cependant cette architecture présente un certain nombre de particularités, autant au niveau de l'architecture matérielle que de la programmation. Cette formation couvre l'ensemble de la démarche d'installation d'un processeur et de programmation de celui-ci, soit directement soit avec un OS embarqué. Les exercices pratiques se feront sur un FPGA de chez Lattice Semiconductor avec le CPU embarqué Mico32 et micrium uC/OSII ou uClinux.

**H2 - Lattice Diamond** Maîtriser l'outil Diamond pour optimiser et debugger un FPGA

Les outils de programmation de FPGA sont des environnement complexes, prenant en charge de nombreuses tâches différentes. Il peut donc être délicat de les utiliser de façon optimale et d'en tirer le meilleur parti. Ce cours vous permettra de maîtriser l'outil Diamond de Lattice Semiconductor pour programmer vos FPGA, les simuler et les mettre au point, que ce soit en VHDL, en Verilog ou en mélangeant les deux langages.

**H6 - Lattice - PCIe 1.1 x1, x4 IP core** This course describes the implementation of the Lattice PCIe core present in ECP2M, ECP3 and SCM FPGA families

**HX1 - Xilinx - Virtex-5 FXT Embedded Processor Block** This course covers the parameterizing of the Embedded Processor Block present in Virtex-5 FXT Xilinx FPGAs.

**HX2 - Xilinx - Spartan-6 / Virtex-6 Integrated PCI Express Block** This course covers the implementation of the Xilinx PCIe block.

**HX3 - Xilinx - Designing with Ethernet MAC logicores** This course covers the implementation of the Ethernet MAC Xilinx logicores.

**HX4 - Xilinx - Microblaze implementation** This course explains how to design a SoC based on MicroBlaze, Xilinx proprietary IPs and/or custom IPs using EDK

This course explains how to design a System on Chip (SoC) based on MicroBlaze, Xilinx proprietary IPs and/or custom IPs using Embedded Development Kit (EDK)

(This Training is mainly based on labs)

**HX5 - Zynq All Programmable SoC: Hardware and Software Design** This course explains how to design a System on Chip (SoC) based on the Zynq-7000 All Programmable SoC

**V1 - VHDL Language** FPGA Programming with VHDL and Simulation (through the training Xilinx, Lattice or Actel FPGA are targeted)

**V2 - Advanced VHDL for FPGA** Acquire a strong design methodology with the best of VHDL for simulation and synthesis

This training is intended to engineers having basic knowledge in VHDL who are willing to acquire a strong designing methodology, and to take the best of VHDL language for digital system simulation and/or synthesis (through the training, Xilinx FPGAs will be targeted)

**V3 - SystemC** Conception et Simulation de systèmes en SystemC

La conception de systèmes électroniques embarqués est de plus en plus complexe et il devient difficile de concevoir séparément le matériel et le logiciel. Le langage SystemC a été conçu pour permettre de concevoir et simuler un système dans son ensemble, aussi bien ses parties matérielles que logicielles et ce même avant de l'avoir partitionné.