
Présentation

Cette formation de **deux jours** (conçue à l'origine par ALSE) enseigne les principes essentiels de la conception des circuits numériques.

Elle permet, indépendamment du langage HDL utilisé, de comprendre la technologie des circuits numériques (CPLD, FPGA et ASIC), mais surtout d'acquérir les **méthodes et bonnes pratiques** pour concevoir facilement des projets de toute complexité, et d'obtenir des résultats performants dont le comportement est **fiable**.

On y apprend comment implémenter efficacement différents types de structures (logiques, arithmétiques, contrôle, FSMs...) et comment **éviter les erreurs** qui conduisent à des résultats erratiques et génèrent des problèmes graves de fiabilité.

Ce cours s'inscrit dans le cadre d'une **action d'acquisition** ou de **perfectionnement des connaissances**.

Les participants peuvent utiliser au choix différents outils de Conception, Synthèse et Simulation (FPGA ou ASIC) durant les **exercices pratiques**, qui occupent environ **50 % du temps de la formation**

Ces **exercices** sont des éléments clés du succès de ces formations, et ils sont indispensables pour un réel apprentissage. Ils **assurent** également **un contrôle continu de l'acquisition des compétences**.

Les **Instructeurs ALSE** sont aussi et surtout des **Experts en Conception et Vérification** qui utilisent à journée entière les méthodologies qu'ils enseignent pour concevoir et vérifier des systèmes complexes. Ils savent partager leur savoir-faire avec passion, et sont particulièrement appréciés des participants.

Objectifs pédagogiques

- ◆ Comprendre l'architecture des Circuits Logiques Programmables (FPGA, CPLD) et des ASIC.
- ◆ Comprendre les limites de fonctionnement et causes d'aléas dans les circuits numériques.
- ◆ Apprendre à implémenter efficacement tous types de concepts et obtenir des circuits plus performants et plus fiables.
- ◆ Savoir comment décomposer les fonctions complexes pour une implémentation efficace.

Qu'apprendrez vous ?

- ◆ Les principes de base de la conception numérique.
- ◆ Une introduction aux méthodes de description HDL.
- ◆ Comment concevoir la logique combinatoire et les blocs séquentiels pour les composants logiques programmables et les ASICs, et en particulier les techniques de conception synchrone.
- ◆ Comment décrire et implémenter les structures fondamentales comme décodeurs, multiplexeurs, compteurs...
- ◆ Comment développer et implémenter des Machines à Etats Finis synchrones (FSM).
- ◆ Une présentation générale des méthodologies et flots de conception ASIC et FPGA.
- ◆ Une introduction aux principes et aux équations économiques des ASIC.

Connaissances Préalables (pré-requis)

La participation à cette formation ne demande pas de connaissance particulière préalable. Elle est idéale pour précéder une formation à un langage HDL, mais elle reste utile aux personnes qui le maîtrisent déjà. Une familiarité avec les principes de base de l'électronique reste souhaitable.

Supports de cours

Les manuels de cours Doulos sont réputés pour être détaillés, précis et faciles d'utilisation. Leur style, leur contenu et leur exhaustivité sont uniques dans le monde de la formation HDL. Ils sont souvent utilisés ensuite comme référence.

Sont compris dans la formation :

- ◆ Le Classeur du cours théorique, avec un Index. Il constitue un **Manuel de Référence** concis.
- ◆ Le **Cahier des Exercices** pratiques, qui permet de mettre en œuvre les connaissances théoriques.
- ◆ Les **fichiers** des exercices et des solutions.



Essential Digital Design Techniques **Programme***

Introduction

- ◆ Designing with programmable logic and ASICs
- ◆ Synchronous design techniques
- ◆ An Overview of HDL-Based Design
- ◆ First (Abel) and second generation HDLs
- ◆ VHDL and Verilog
- ◆ Design process using HDLs

Digital Design Fundamentals

- ◆ Representing bits and three-states ◆ Unsigned and signed (two's complement) numbers
- ◆ Static and dynamic definition of combinational logic
- ◆ Logic minimisation
- ◆ Avoiding asynchronous sequential logic

Synchronous Sequential Logic

- ◆ Principles ◆ Using D-type flip-flops
- ◆ Characterisation - timing constraints
- ◆ Timing violations and metastability issues
- ◆ Timing performance of synchronous systems
- ◆ Static timing analysis
- ◆ Other flip-flop types

Introduction to Programmable Logic

- ◆ Survey of programmable logic devices
- ◆ Selecting an appropriate device ◆ Importance of the internal structure
- ◆ Pull-ups; open collector; tristates and bi-directional ◆ Tristate bubble-up
- ◆ JTAG boundary scan

Inputs & Outputs

- ♦ I/O pin standards ♦ Pin assignment
- ♦ Transmission Lines and signal integrity
- ♦ Ground bounce and solutions
- ♦ Electromagnetic interference & compatibility (EMI & EMC)

Common Functions and their Implementation

- ♦ Encoders and decoders ♦ Priority encoders
- ♦ Multiplexers ♦ Tristates used as Muxes
- ♦ Parity generator
- ♦ Shift Registers
- ♦ Johnson (ring) "counters"
- ♦ Linear Feedback Shift Registers (LFSR) & PRNG

Arithmetic Structures

- ♦ Half and full adders ♦ Large adders ♦ Carry lookahead adder
- ♦ Pipelining ♦ Synthesis of adders
- ♦ Counters ♦ Wide counters
- ♦ Binary to BCD conversion
- ♦ Serial arithmetic
- ♦ Importance of synchronous design

Synchronous Finite State Machines and Memories

- ♦ Definition ♦ Graphical entry and symbolism
- ♦ Moore and Mealy structures
- ♦ Implementation
- ♦ States encoding and optimisation
- ♦ Using HDLs to design Finite State Machines
- ♦ Using memories ♦ Memory types

Introduction to ASICs

- ♦ ASIC types and technologies
- ♦ ASIC economics
- ♦ Design For Test
- ♦ Design process for ASICs

* : *programme donné à titre indicatif (certaines mise à jours et modifications peuvent ne pas y figurer)*