



## V1 - Le langage VHDL

### Programmation de FPGAs en VHDL

#### Objectifs

- connaître les différentes méthodologie de modélisation des composants
- savoir exploiter les possibilités du langage
- modéliser des circuits combinatoires
- mettre en oeuvre des machines d'état
- réutiliser des composants
- savoir écrire en VHDL pour la simulation
- déterminer les chemins critiques et vérifier le respect des timings

*Les exercices de ce cours peuvent être réalisés au choix sur plateforme à base de FPGA Lattice ou Actel*

#### Matériel

- Un PC par binôme
- Exercices sur plateforme Lattice:
  - Environnement de développement ISPlever de Lattice
  - Synplify PRO pour la synthèse, Active-HDL pour la simulation
  - Carte à base de FPGA Lattice ECP2 ou MachXO
- Exercices sur plateforme Actel:
  - Environnement de développement Libero d'Actel
  - Synplify PRO pour la synthèse, ModelSim pour la simulation
  - Carte à base de FPGA Actel Fusion

#### Pré-requis

- Connaissances en technologie numérique
- Notions d'algèbre de Boole
- Quelques notions de programmation (quel que soit le langage) sont souhaitables.

#### Plan

##### Premier jour

##### **De la porte logique au FPGA**

- Structure d'un Circuit Intégré

- SSI (small scale integration), logique TTL
- MSI (medium scale integration), PALs, GALs, PLDs
- LSI (large scale integration), CPLDs
- VLSI (very large scale integration), ASICs, ASSPs, FPGAs
- Evolution des architectures logiques
- Contraintes des technologies
  - Méthode d'interconnexion (SRAM, Fuse, AntiFuse, Flash)
  - Réseaux d'horloges
  - Type d'éléments logiques
- Problèmes de timings

### **Les apports du VHDL**

- Intérêt d'une programmation en VHDL
- Différentes étapes de la conception
  - saisie
  - simulation
  - synthèse
  - adaptation à la technologie (mapping)
  - placement et routage
  - analyse des timings
  - génération des binaires (bitstream)

### **Concepts de base**

- Entités et architectures
- Ports, signaux et variables
  - types de données
  - attributs
- Différents styles d'architecture
- Environnements de simulation (Test Bench)

*Exercice : prise en main des outils de conception et découverte du flot de développement à l'aide d'exemples élémentaires*

### **Deuxième jour**

#### **Logique combinatoire en VHDL**

- Utilisation des bibliothèques IEEE
- Outils de modélisation de composants
- Traitements concurrents et séquentiels
  - affectations
  - processus
  - traitements conditionnels
- Opérateurs relationnels
- Opérateurs arithmétiques
- Surcharge des opérateurs
- Instantiation de composants

*Exercice : conception d'une logique de décodage d'adresses*

#### **Notion de bancs de test (TestBench)**

- Création d'un package
- Génération des vecteurs de test
  - intégration des vecteurs dans le source du bench

- lecture depuis un fichier
- stocker les résultats pour analyse ultérieure

*Exercice : test du décodeur d'adresses*

### Troisième jour

#### **La logique séquentielle**

- Rappels sur les bascules
  - problème des états métastables
- Gestion du reset
  - prise en compte des reset/presets asynchrones
- Buffers trois-états
  - interface avec des bus

*Exercice : conception d'une FIFO*

#### **Le test des modèles synchrones**

- Vérification des timings
  - lors de la simultaion
  - tests de faisabilité du placement routage
  - simulation après placement routage

*Exercice : vérification de la FIFO à vitesse élevée*

#### **Fonctions et procédures**

- Passage de paramètres à des fonctions
- Surcharge des fonctions
- Différences entre procédures et fonctions

*Exercice : conception et test d'un arbitrage de bus*

### Quatrième jour

#### **Les machines d'états**

- Rappels sur les machines de Mealy et de Moore
  - représentations graphique
  - implémentation
  - traduction en VHDL
- Principes de conception d'une FSM à 2 process
- Reset d'une machine d'états
- Utilisation de la simulation pour vérifier la conception
- Optimisation de l'utilisation des ressources

*Exercice : conception d'une interface SRAM burst*

#### **Optimisation de la conception**

- Optimisation de l'utilisation des ressources
- Optimisation des traitements
  - temps de traitement

- fréquence de traitement
- concept de pipeline

*Exercice : Optimisation des modèles précédents*

## Conception de systèmes complexes

- Découpage hiérarchique
- La librairie work
- Création de librairies
- Packages

*Exercice : exemple de l'interface PCI*

## Renseignements pratiques

**Duration : 4 days**

**Cost : 1900 € HT**

**Prochaines sessions : du 10 au 13 April 2012**



SARL au capital de 15400€ - SIRET 449 597 103 00026 - RCS Nanterre - NAF 722C - Centre de Formation : 19, rue Pierre Curie - 92400 Courbevoie  
Siège social et administration : 21, rue Pierre Curie - 92400 Courbevoie - Tél. 01 41 16 80 10 - Fax. 01 41 16 07 78

Last site update: Mon 13 Feb 2012 02:28:18 PM CET

<http://www.ac6-formation.com/>