



## V3 - SystemC

### *Conception et Simulation de systèmes en SystemC*

#### Objectifs

- Comprendre les avantages de la simulation système avec SystemC
- Maîtriser les différents niveaux de modélisation
  - Modèles transactionnels
  - Modèles du logiciel
  - Modèles du matériel
  - Modèles synthétisables
- Modélisation du système:
  - UTF (UnTimed Functional model)
  - TF (Timed Functional model)
- Modélisation du matériel:
  - BCA (Bus Cycle Accurate model)
  - PCA (Pin Cycle Accurate model)

Ce cours est accompagné de nombreux exercices réalisés en binômes et décrit la version 2.2 de SystemC

#### Matériel

- Un PC par binôme avec
  - Visual Studio 2005
  - SystemC-2.2

#### Pré-requis

- Connaissance de base du langage C++ (voir par exemple le cours [L3 - C++](#))

#### 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
- Au début de chaque demi-journée 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.

# Course Outline

## Premier jour

### Rappels sur le langage C++

- La programmation objet
  - Classes et objets
  - Attributs
- Méthodes et opérateurs
  - Surchage
  - Constructeurs et Destructeurs
  - Méthodes virtuelles
  - Références
  - Paramètres par défaut
- Gestion de la mémoire
  - Les opérateurs new et delete
- Les espaces de nommage
- Les entrées/sorties standard

### Aspects avancés de C++ pour SystemC

- Templates de classes et de fonctions
  - Définition de templates
  - Contraintes
  - Instanciation automatique
  - Instanciation manuelle
- Conversions entre types
  - Conversions implicites
  - Conversions définies par l'utilisateur
  - Opérateurs de copie et initialisations
  - Conversions montantes et descendantes (up cast/down cast)
- Exceptions

## Deuxième jour

### Introduction à SystemC

- Les bases de SystemC
  - Objectifs du langage
  - Historique
  - Avantages et inconvénients de SystemC
- La modélisation au niveau des transactions (TLM)
- Le flot de coception avec SystemC
  - Modèle algorithmique
  - Modèle TLM
  - Partitionnement matériel/logiciel
  - Synthèse directr ou traduction en HDL
  - Simulation des différents modèles
- L'architecture de SystemC
  - Canaux de communication
  - Eléments de structure
  - Types de données

- Moteur de simulation

## Éléments de base du langage SystemC

- Éléments de structure
  - Modules, Ports et Signaux
  - Canaux primitifs
- Création de la structure
  - Instanciation des modules
  - Connexion des ports
- Processus et gestion du temps
  - Méthodes et Threads
  - Evènements
  - Sensibilité statique ou dynamique
  - Temps et horloges
  - Processus dynamiques

## Troisième jour

### Simulation d'un modèle SystemC

- Démarrage et arrêt de la simulation
- Élaboration du modèle
  - Phases d'élaboration statique
  - Phases d'élaboration dynamique
  - Notion de chercheur d'évènement (event finder)
  - Callbacks d'élaboration
- Les phases de simulation
  - Notifications d'évènements
  - Attentes sur évènement et triggers
  - Files d'évènements
- Techniques de debug
  - Affichages et traces
  - Gestion d'erreurs
  - Trace de signaux cachés et de variables

## Quatrième jour

### Modèles précis au niveau bus et bit

- Modèles *Bus Cycle Accurate* et *Pin Cycle Accurate*
- Modélisation des bus
  - Interfaces et canaux de communication
  - Interfaces maîtres et esclaves
  - Méthodes d'interface (bloquantes et non-bloquantes)
  - Utilisation d'évènements dans les canaux
  - Sensibilité dynamique pour un canal
- Modélisation de bus multi-ports
  - Politiques de connexion de ports
- Modèles précis au niveau bit
  - Types de données totalement spécifiés
  - Affectation et troncature
- Types logiques et vecteurs
  - Sélection de bits et de tranches
  - Concaténation de types
  - Résolution de types
- Types entiers et virgule fixe

## Modélisation par raffinement des modèles

- Raffinement d'algorithmes
  - Utilisation de modèles UTF puis TF avant partitionnement matériel/logiciel
  - Annotations de timings
- Différentes méthodes de raffinement
  - Raffiner la structure
  - Raffiner les données
  - Raffiner les communications
- Raffinement des canaux
  - Le concept d'adaptateur
  - Réalisation d'un adaptateur
  - Création d'un *event finders* spécialisé