



<http://www.alse-fr.com>

# Formation Pratique Conception Intel FPGAs Intel® Quartus® Prime



**Objectifs** Cette **première partie** permet de maîtriser le flot de conception des FPGAs Intel et les très nombreux outils intégrés à la suite Intel® Quartus® Prime software (souvent méconnus ou mal utilisés). Cette partie comprend aussi des concepts importants, dont les fondamentaux de l'Analyse de Timing, et assure ainsi une préparation à la deuxième partie de la formation (sur Les Timings et l'optimisation).

Chaque chapitre est suivi d'un **exercice pratique indispensable** pour un réel apprentissage, qui occupe environ **50 % du temps de la formation**.

Ces exercices **assurent** aussi un **contrôle continu de l'acquisition des compétences**.

Cette formation sera **sanctionnée** par la délivrance d'une **fiche de présence**.

**Pré-requis** Connaissances de base solides en électronique numérique et en conception FPGA.

**Durée** **Deux (2) jours** intenses soit **14 heures**

**Format** **Enseignement en ligne par Instructeur** à distance qui donne le cours, répond aux questions, surveille la bonne exécution des exercices, et fournit une aide individuelle si nécessaire.

**Matériel** Requier un PC Windows ou Linux avec un écran confortable sur lequel sera installé Quartus Prime Standard, ainsi qu'un accès stable à Internet et micro, haut-parleurs (ou casque) et une webcam (optionnelle). Un environnement calme et non perturbé est indispensable.

**Exercices** Ils seront exécutés sur le PC du stagiaire avec un **kit FPGA** (qui peut être prêté). La version gratuite (Lite) de Quartus Prime Standard peut suffire pour cette première partie.

ALSE est depuis 1996 un Organisme de Formation Professionnelle Continue déclaré auprès de la DRIEETS sous le numéro 26.21.01281.21, référencé dans le DataDock et **Certifié Qualiopi** « Actions de Formation ». Cette formation est donc possible dans le cadre de l'article L920-1 du Code du travail.



Cette formation ne **comprend PAS** les sujets suivants :

Conception des circuits numériques, Conception avec un Langage HDL (VHDL, Verilog, SystemVerilog), Vérification avec les langages HDL et Méthodologies, Conception de Systèmes Embarqués, SoPC Nios et NiosV, Conception de systèmes avec Platform Designer (ex Qsys), Compilation incrémentale, Reconfiguration partielle, Optimisation pour les architectures HyperFlex, le Langage Tcl/Tk...

**Toutes ces compétences peuvent être acquises au travers de nos autres formations.**

# Programme de la Formation « Quartus Prime » <sup>1</sup>

## Journée 1

- **Introduction** à Intel® Quartus® Prime software et au flot de conception.  
Présentation rapide des familles FPGA, des versions de Quartus et du support des OS  
L'Interface Utilisateur Graphique et le flot général.  
Méthodes d'entrée de données de conception (HDL, Megawizard, Schémas graphiques, Memory Editor...)  
Création et gestion d'un projet (archivage, databases), les fichiers clés d'un projet Quartus.  
*Exercice pratique complet sur kit FPGA.*
- **Les outils de base du flot**  
Assignment Editor, Assignations des broches,  
Utilisation de l'éditeur de contraintes, vérification dynamique,  
Pin Planner, Import/Export CSV,  
Placement-Routage, contrôle du Fitter.  
*Exercice pratique sur maquette.*
- **Principes fondamentaux de l'Analyse Timing et Introduction aux Contraintes SDC**  
Rappel de l'analyse timing basée sur les phénomènes physiques.  
Introduction et terminologie TimeQuest – Timing Analyzer / SDC.  
Guide de Survie pour le langage SDC, expliquant les contraintes essentielles (et suffisantes) pour contraindre un design ou un bloc Full Synchrone avec entrées / sorties Asynchrones.
- **Modes de Configuration des FPGAs**  
Les modes de configuration.  
Téléchargement du FPGA, chaînes JTag FPGA et mixtes,  
Programmation Flash indirecte (JIC).  
JTag Server & programmation à distance via Ethernet.  
*Exercice pratique sur maquette (contraintes SDC, analyse Timing et Téléchargement)*
- **Utilisation avancée de Quartus Prime**  
Bases de données version-compatibles,  
Archivage intelligent des projets,  
Création et comparaison de Révisions,  
Analyse par les Netlist Viewers : RTL, Technology, et FSM.  
Cross-probing, suppression intelligente des messages,  
Vérification automatique des règles de conception avec le Design Assistant (DRC).  
Les conseillers (Advisors).  
*Exercice pratique.*

<sup>1</sup> NB : Contenu et répartition sont fournis à titre indicatif et sont susceptibles d'être modifiés ou adaptés par l'instructeur.

# Journée 2 <sup>1</sup>

- **Automatisation du flot de Conception** : travailler Mieux et Plus vite !  
Les tâches du flot peuvent être automatisées et sécurisées par des scripts en ligne de commande ou (mieux) en langage Tcl/Tk : Création automatique de projet, gestion des fichiers, archivage, nettoyage, compilation, création des bitstreams, test automatique des résultats, etc... La richesse de l'API Tcl de Quartus permet de tout envisager.  
*Exercice pratique complet (avec scripts Tcl) sur kit FPGA.*
- **Chip Planner.**  
Un outil très puissant pour comprendre et analyser l'implémentation exacte de votre projet dans le FPGA. Permet également la modification manuelle de très bas niveau (assez largement déconseillée).  
Concepts, utilité, mise en œuvre, cas d'utilisations et limitations.
- **Estimation et Optimisation de Consommation**  
Utilisation de PowerPlay, estimation précoce, affinée par simulation, injection de vecteurs et analyse statistique. Utilisation de **ModelSim** ou **QuestaSim**, utilisation du GUI, automatisation par script Tcl.  
Options d'optimisation de la consommation, le conseiller Power Advisor.  
*Exercice pratique* : utilisation de ModelSim (ou QuestaSim) pour une Estimation précise de consommation
- **In-System Memory Contents Editor**  
Un excellent outil, très largement méconnu mais pourtant simple d'emploi et qui rend des services inestimables pour faciliter la mise au point de vos applications ! Concepts, Mise en Œuvre pratique, Applications.  
*Exercice pratique sur kit FPGA*
- **In-System Sources & Probes :**  
Comme l'ISMCE, cet outil pratiquement irremplaçable est lui aussi très utile !  
Concepts, domaines d'applications, Mise en Œuvre.  
*Exercice pratique*
- **SignalTap**  
Principes de l'Analyse Logique Temps Réel Embarquée de vos projets. Création des fichiers STP, types de buffers, méthode de stockage, choix des triggers, triggers avancés, visualisation des signaux HDL de type FSM, °Plugins, différents types d'acquisition.  
*Exercice pratique complet sur kit FPGA*
- **SignalTap Avancé**  
Les options avancées de SignalTap qui permettent d'en augmenter considérablement l'utilité et le champs d'application : qualifieurs pour le stockage, triggering avancé, spécification par machine d'états, les différents types d'acquisition. Recompilation incrémentale. Mise en œuvre du Power-Up Trigger. Scripting.  
*Exercice pratique sur kit FPGA*
- **SignalProbe, Logic Analyzer Interface (optionnel)**  
Toujours des outils d'instrumentation (via JTag) qui peuvent aider à la mise au point et l'analyse embarquée temps réel de vos projets. Concepts et applications.
- **SSN Analyzer (optionnel)**  
Concept, utilité, mise en œuvre de cet outil d'estimation et de vérification d'intégrité de signal.

*Cette formation est conçue pour être suivie de la  
Formation Avancée « **Timing** » (pages suivantes).*

<sup>1</sup> NB : Contenu et répartition sont fournis à titre indicatif et sont susceptibles d'être modifiés ou adaptés par l'instructeur.



<http://www.alse-fr.com>

# Formation Pratique Conception Intel FPGAs Timings



**Objectifs** Cette formation dense de deux jours est réservée utilisateurs ayant déjà acquis toutes les bases car elle permet d'aborder les techniques plus avancées : Analyse et Contraintes Timing approfondies avec TimeQuest / Timing Analyzer (sur 2 jours).

Ces techniques sont indispensables pour tous les projets d'une certaine complexité, notamment ceux qui implémentent des interfaces rapides (interfaces mémoires synchrones, ADC, DAC, xGMII, etc...) avec contraintes timings complexes et multiples domaines d'horloges. Cette formation est alignée sur la dernière version des outils de conception.

Chaque chapitre est suivi d'un exercice pratique indispensable pour un réel apprentissage, qui occupe environ 50 % du temps de la formation.

Ces exercices assurent aussi un contrôle continu de l'acquisition des compétences.

Cette formation sera sanctionnée par la délivrance d'une fiche de présence.

**Pré-requis** Pratique très fluide de l'ensemble des outils Intel-FPGA et si possible participation à la formation Quartus de « Essentielle » de 2 jours.

**Durée** Deux (2) jours soit 14 heures ; Enseignement en ligne (à distance) par Instructeur

**Matériel** Requier un PC Windows ou Linux avec un écran confortable, ainsi qu'un accès stable à Internet avec une bande passante correcte et un micro+casque.

## Programme de la Formation « Avancée »<sup>1</sup>

### Journée 1 – Timing Analyzer Essential (ex TimeQuest)

➤ **Intel Timing Analyzer, et les contraintes SDC**

**Partie I – Introduction, Présentation des Concepts et de l'Interface Utilisateur.**

Rapide Introduction et terminologie TimeQuest.

Méthodologie d'utilisation de TimeQuest depuis l'interface graphique et à partir des fichiers SDC.

Contraindre son design, Générer des rapports timings, comprendre et savoir interpréter les rapports issus de TimeQuest. Flot de base avec Quartus.

*Exercice pratique*

➤ **Partie II – Rapports Timing & Contraintes d'horloges**

Rapports Timings Avancés, Analyse de la représentation en waveform, scripts Tcl, custom reporting.

Cross-probing avec les outils du flot Quartus (Chip Planner, Technology Map Viewer, Constraints Editor...)

Contraintes sur les différents types d'horloges : horloges de base, horloges dérivées, horloges virtuelles, PLL. Analyses Core et Latency, incertitudes, prise en compte du jitter.

*Exercice pratique*

➤ **Partie III – Contraindre les I/Os, Interfaces Synchrones**

Contrainte des interfaces Combinatoires, des interfaces Synchrones, délais min/max sur les I/Os, utilisation des Horloges Virtuelles, prise en compte de la charges sur les I/Os, Analyse avancée des I/Os.

*Exercice pratique*

➤ **Partie IV – Chemins Asynchrones et Exceptions Timing**

Concepts avancés, Analyse Recovery-Removal, chemins Asynchrones et faux chemins,

Analyse et contrainte des traversées de domaines d'horloges,

Problématiques de skew, multiplexage d'horloges,

Exceptions multi-cycles.

*Exercice pratique*

<sup>1</sup> NB : Contenu et répartition sont fournis à titre indicatif et sont susceptibles d'être modifiés ou adaptés par l'instructeur.

## Journée 2 – Timing Analyzer – Utilisation Avancée <sup>1</sup>

- **Optimisation des Timings** – Optimisation à la synthèse, Timing Driven Synthesis, WYSIWIG Resynthesis, Synthèse Physique, Retiming, Register duplication, Options du Fitter.
- **Design Space Explorer**. Concept et utilité. (optionnel)
- **Exceptions Timings**  
Concepts avancés sur les cas d'utilisation des exceptions Multi-Cycles en Setup et Hold.  
Gestion des exceptions et de leurs priorités dans Timequest.  
Analyse des Clock Enable pour logique multi-cycle, utilisation des Fanouts/Through. Rapports Timings associés.  
*Exercice pratique optionnel*
- **Interfaces Source-Synchrones – SDR**  
Introduction Concept Interfaces Sources Synchrones. Schémas SDR et DDR, Center-aligned et Edge-aligned.  
Contrainte des interfaces Source-Synchrones SDR Input et Output. Analyse des différentes méthodes de contraintes (FPGA Centric ou Board System), Horloges Virtuelles, Gestion des PLL. Rapports Timings associés.  
*Exercice pratique*
- **Interfaces Source-Synchrones – DDR (optionnel)**  
Contrainte des interfaces Source-Synchrones DDR Input et Output. Analyse des différentes méthodes de contraintes (FPGA Centric ou Board System) Gestion des PLL, des DDIO. Gestion des Exceptions.  
Rapports Timings associés.  
*Exercice pratique optionnel*
- **Design avec Feedback externe**  
Concept, analyse et contraintes associées. Gestion des cas Horloges et Data. Rapports Timings associés.  
*Exemple pratique : contraintes d'une Flash Quad-SPI rapide*
- **Utilisation du Tcl : scripting et SDC avancé**  
Utilisation du Tcl pour analyse timing avancée, Command-Line, Rapports custom.  
Utilisation du langage Tcl pour création de contraintes SDC avancées.

<sup>1</sup> NB : Contenu et répartition sont fournis à titre indicatif et sont susceptibles d'être modifiés ou adaptés par l'instructeur.