

Advanced Logic Synthesis for Electronics

<http://www.ALSE-FR.com>



Systeme Embarqué Nios II & Qsys ?

Maîtrisez... grâce à notre formation :

> SoPC Intel-FPGA < Essential + Expert

Pour les ingénieurs désireux d'adopter la technologie de Systèmes embarqués sur FPGA qui s'est imposée depuis plusieurs années, la courbe d'apprentissage reste assez ardue, car la technologie est de plus en plus riche et sophistiquée, et les projets de plus en plus ambitieux.

La formation de base (« **Essential** ») de trois jours est le point de départ incontournable pour tout profil d'utilisateur, et elle assure une bonne compréhension de l'ensemble de la méthodologie. Elle est destinée aussi bien aux architectes et concepteurs Hardware qu'aux ingénieurs de développement logiciel embarqué (*Embedded Software*).

La Formation « **Expert** » est largement orientée Hardware et Système. Elle est destinée aux architectes et ingénieurs qui sont appelés à développer des nouveaux systèmes et des nouvelles architectures, à créer des périphériques spécifiques et à optimiser les performances de systèmes complexes.

**Ne démarrez pas un projet System on Programmable Chip sans avoir suivi
(au moins) la formation SoPC Essentielle !**

Si vous êtes déjà utilisateur de cette technologie, *mieux vaut tard que jamais* !

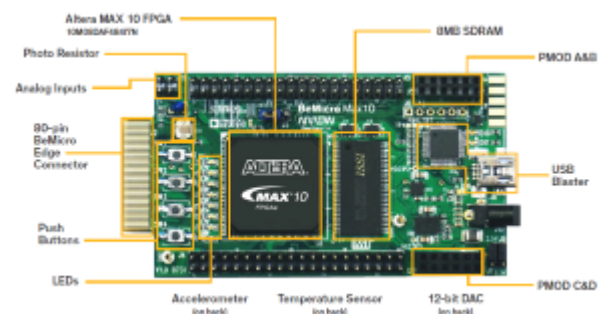
Ces formations vous permettront de gagner du temps, de corriger vos erreurs, d'optimiser vos projets, de mettre en œuvre de meilleures solutions, d'obtenir des meilleures performances et une meilleure fiabilité. Vous découvrirez également comment bien utiliser la *fabric Network On Chip* générée par **Qsys**.

Le succès de ces formations tient pour une bonne part aux nombreux exercices pratiques et concrets qui permettent d'appliquer réellement les concepts, pas-à-pas, et de les tester sur carte.

Les aspects matériels et logiciels sont couverts par ces formations et il est très important que les ingénieurs des deux cultures s'y retrouvent ! Même si certaines sections peuvent sortir des cœurs de métier respectifs, cette ouverture et cet échange représente en fait une énorme opportunité et un réel avantage pour converger vers des solutions optimales et des performances exceptionnelles, hors de portée des solutions purement logicielles ou purement matérielles.

Ce mélange de cultures est la condition indispensable pour que des projets complexes trouvent des solutions optimisées et efficaces.

Évidemment, ce que vous apprendrez est complètement indépendant du composant ou de la carte que vous utiliserez ensuite ! Cependant, nous avons porté tous nos **exercices pratiques** sur un Kit FPGA très peu coûteux (**BeMicro Max10**) représenté ci-contre, qui est disponible en ligne pour environ 30 €uros.





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

Formation Pratique Intel-FPGA System on Programmable Chip Nios II & Qsys



Contenu Cette formation est proposée en **deux parties** pour un total de cinq jours.

La **première partie** de trois jours concentre tout le savoir-faire **fondamental** et **essentiel** pour démarrer la conception de Systèmes avec Processeur Embarqué : Présentation des solutions Intel FPGA, construction de la plate-forme hardware, choix du processeur, choix et intégration des périphériques matériels, des interconnexions et des flots de communication etc... ainsi que les bases de l'utilisation de l'environnement de développement logiciel Nios II EDS / Eclipse.

La **deuxième partie**, fortement orientée Hardware et destinée aux **utilisateurs avancés**, va plus loin en exposant les concepts hardware détaillés (système d'interconnexion *Network On Chip* **Qsys** et différents bus), la création de périphériques custom complexes, jusqu'à l'écriture de drivers dans la couche d'abstraction matérielle.

Cette formation est **indispensable** pour construire un Système et des Applications performants et fiables en tirant bénéfice des riches possibilités de l'architecture SoPC (construction de flots de données et d'une *fabric* efficaces, paramétrage des interconnexions et de flux, ajout de nouvelles instructions personnalisées au processeur ou encore mise en œuvre de l'accélération matérielle).

Les aspects Matériel **et** Logiciel sont abordés dans ce stage pour permettre aux ingénieurs hardware et software d'avoir une vue complète des deux mondes, ce qui est **indispensable** pour concevoir un système embarqué efficace et optimisé.

Théorie et Pratique alternent à travers de très nombreux exercices décrits pas-à-pas, implémentés et testés sur un Kit Intel-FPGA facilement disponible : **le BeMicro Max 10**.

Ces exercices **indispensables** pour un réel apprentissage, **assurent aussi un contrôle continu de l'acquisition des compétences**.

Pré-requis	Connaissances de base en langage C et de préférence en Conception FPGA Intel (VHDL).
Durées	Trois jours soit 21 heures, deux jours soit 14 heures, ou cinq jours soit 35 heures.
Prix	1 950 € HT (<i>Essential</i>), 1 290 € HT (<i>Expert</i>) ou 2 990 € HT (<i>Complet</i> , 5 jours, sur facture unique)
Lieu	Sur notre site de Paris 13ème ou sur site (demander un devis pour le nombre de stagiaires)
Horaires	9h30 → 18h, 9h → 18h, 9h → 17h30, soit 21 heures effectives (pour les trois premiers jours) 9h → 18h, 9h → 17h00, soit 14 heures effectives (pour les deux derniers jours).

Objectifs

Partie I « Essential »

- Acquérir le **savoir-faire essentiel** pour la Conception et la Programmation de Systèmes On Chip avec Processeur(s) embarqué(s) sur FPGAs Intel.
- Apprendre à concevoir la plate-forme hardware avec l'outil **Qsys**
- Découvrir la Méthodologie logicielle et la couche d'abstraction matérielle (HAL Nios II).
- Maîtriser l'environnement de développement logiciel Nios II EDS / Eclipse.

Partie II « Expert »

- **Maîtriser les concepts avancés** dont **Qsys**, le nouvel outil de génération « Network On Chip ».
- Les modes avancés des différents bus, la compréhension des protocoles hardware et timings, l'optimisation de la bande passante, la création de composants custom complexes et leur intégration automatique dans la librairie de composants, l'accélération matérielle.
- Les nouvelles interfaces **AXI 3 & 4**. Les *Bus Functional Models (BFMs)* Avalon & AXI.
- La création de drivers au formalisme de la Hardware Abstraction Layer

Pré-requis

- Avoir besoin d'implémenter un Système on Programmable Chip sur FPGA Intel (avec ou sans processeur embarqué).
- Connaissances de base en Électronique Numérique et en Conception de Logique Programmable.
- Connaissances de base (rudiments) en Langage C.
- Pour la partie II « Expert », il est obligatoire d'avoir suivi la Partie I (3 premiers jours) de la formation.

N.B : la formation « Conception avec Quartus Prime » n'est pas un pré-requis et elle peut être suivie séparément.

Remarque: les Exercices pratiques de la formation utilisent un Kit Intel FPGA facilement disponible, ce qui permet de pouvoir les reproduire ultérieurement après la formation.

Formation « Essential » System on Programmable Chip Intel-FPGA Qsys Programme*

SOPC – Qsys – Nios II – Partie Hardware : Construire la Plateforme Matérielle

Journée 1

- **Les bases de la Conception d'un « System on Programmable Chip » (SoPC)**
Avantages et raison d'être des architectures SoPC FPGA. Introduction aux processeurs NIOS II. Configuration et choix du processeur, les Périphériques Utilisateur usuels (SystemID, Jtag UART, On-chip memory, PIO, Timer, Nios, JTag debug), Introduction à **Qsys**. Le Kit BeMicro Max 10.
Exercice : Création d'un système SoPC complet en utilisant Qsys et test sur le Kit.
- **Introduction à Qsys et au concept Network On Chip.**
Concepts, avantages, inconvénients, limitations, optimisation, types d'interfaces principaux.
- **Validation de la plateforme matérielle par (co)simulation HDL.**
Création des modèles de simulation, modélisation des composants périphériques et UART, utilisation du Simulateur ModelSim.
Exercice : Simulation RTL par ModelSim AE du système SoPC de l'exercice précédent.
- **Génération par Qsys de la Matrice d'Interconnexion**
Les formalismes d'interconnexion Avalon MM et Avalon ST, les différents types de ports et les modes de transfert et d'Adressage. Ajout de Périphériques Custom.

* NB : Contenu et répartition sont fournis à titre indicatif et sont susceptibles d'être modifiés ou adaptés par l'instructeur.

Journée 2

- **Introduction aux Périphériques « Custom »**
On-chip et Off-chip, utilisation pas-à-pas du Component Editor, scriptage Tcl, le fichier _hw.tcl, publication du composant.
Exercice Pratique : Conception VHDL et mise en œuvre d'un Périphérique utilisateur custom (PWM)
- **L'Optimisation matérielle des Performances**
Engorgement des architectures traditionnelles et Présentation du concept de l'accélération matérielle. Les systèmes Multi-maîtres et la matrice NOC.
Custom Masters : les processeurs DMA, Construction de blocs d'accélération et de traitement.
Custom Instructions : principe, avantages, facilité de conception et d'intégration. Exemple CRC.
Custom Peripherals : blocs de traitement personnalisés.
Exercice : Accélération d'algorithme de calcul par 3 méthodes et comparaison des performances.

SOPC – Software : Développement Logiciel sur la plateforme SOPC

- **Le processeur NIOS II - Flot de Conception du Logiciel.**
Concepts fondamentaux, aspects logiciels, l'environnement NIOS II EDS / Eclipse SBT, Création d'un projet logiciel, présentation des outils de base pour compilation et mise au point, Gestion des projets, Application C/C++ et Board Support Package (BSP), Présentation de l'environnement de debug hardware.
Exercice : Création d'un premier projet logiciel complet, compilation et test + debug sur le Kit.

Journée 3

- **Présentation du Flot de conception en ligne de commandes.**
Motivations, compatibilité avec le flot GUI, utilitaires d'automatisation et de scripting.
Download et lancement d'application par script.
Exercice : Utilisation du flot en ligne de commande pour automatiser complètement un projet.
- **Développer des Programmes pour Nios II : la Couche d'Abstraction Matérielle HAL.**
Introduction à la couche d'abstraction matérielle (Hardware Abstraction Layer) et aux périphériques standard : horloge système, alarmes, timers haute résolution. Accès aux périphériques, interaction avec la mémoire cache, le cas des périphériques sur mesure (Custom).
Exercice : Gestion des périphériques PIOs & Timers par l'utilisation de l'API HAL, .
- **Introduction aux Architectures Logicielles des Systèmes Embarqués**
System.h, mise à jour de la configuration matérielle, le mapping mémoire, la pile et le tas, le linker, connaître la séquence de boot (alt_sys_init et alt_main), approche *Hosted* contre *FreeStanding*. La séquence de boot et le *Boot Copier*. Notions de base sur l'optimisation de la taille du code généré.
- **Les Interruptions.**
Les avantages et les risques à utiliser des traitements sous interruption. Présentation des deux modes disponibles. Avantages et implications du nouveau contrôleur cascadable d'Interruptions vectorisées. Mise en œuvre du contrôleur intégré via l'API logicielle.
Exercice : Modification du projet PWM pour traitement par interruptions du PIO edge-triggered.
- **Aspects Logiciels de l'Accélération Matérielle**
Le flot d'optimisation. Identification des goulots d'étranglement des performances. Mise en œuvre des Custom Instructions, des Périphériques dédiés, des DMAs à travers l'API standard, solution alternative. Les problèmes liés aux caches.
Exercice : Mise en œuvre de différentes méthodes d'accélération et comparaison des gains.

Deuxième Partie - Concepts Avancés Qsys

Réservée aux Ingénieurs ayant suivi la Première partie de 3 jours.

Formation « Expert » System on Programmable Chip Intel-FPGA Qsys Programme*

Journée 1 (ou 4 sur 5)

- **Qsys : la nouvelle matrice d'Interconnexion « Network On Chip »**
Limitations des systèmes d'interconnexion traditionnels. Bottlenecks, limitations de bande passante, latence, gestion des périphériques à latence variable, arbitrage, mode Burst, outils de mise au point. Les avantages de la technologie NOC, comparaisons avec SOPC Builder, les Interfaces supportés.
- **QSys : Architecture du NOC, Protocoles supportés (Avalon & AXI), optimisation des performances**
Architecture interne, pipeline, optimisation des paramètres, trouver les bons compromis pour le meilleur résultat en Fmax et en surface. Utilisation de l'outil Qsys, interface **AXI 3 & 4**.
Exercice : Évolution d'un système multi-maîtres en Qsys et comparaison des performances.
- **Construire des composants Custom Avalon-MM**
Esclaves MM et Maîtres MM : transferts, gestion de la latence, signaux d'interface, Adressage. Pipelining, Latence variable, mode Burst (avantages et inconvénients). Concepts avancés (buffers élastiques, gestion de flux par pression. « Avalonisation » d'une IP.
Exercice : Codage VHDL et test sur carte d'un périphérique MM Master.
- **Component Authoring**
Search path & QIP, fichier « .ipx », « *_hw.tcl », versionnage des composants.
Les fonctions de callback, application à la validation et à l'élaboration. Paramétrage d'Instance.
- **Conception de Composants avancés**
Génération d'une horloge, d'interruptions. Présentation du système d'interruption Nios Vectorisé (VIC) : calcul d'adresse, de priorité, latence, API de mise en œuvre, support BSP, support RTOS.
Exercice : Ajout d'une interruption à un périphérique, ajout du VIC, mise en œuvre de l'API VIC.

Journée 2 (ou 5 sur 5)

- **Avalon-ST : les flux Streaming.**
Le formalisme Avalon-ST, concepts et exemples d'utilisation. Périphériques ST standard. Adaptateurs et MUXes. Contrôle de flux par Pression. Interruptions. Intérêt de la mixité MM+ST, « Avalonisation ».
Exercice : Conception d'un composant Avalon ST custom.
- **La Suite de Vérification Avalon + AXI & Intégration système avec Qsys**
Les BFM's Avalon-MM, Avalon-ST & AXI BFM's. Les Moniteurs, les bancs de test complexes. API, la *Verification Suite*, et la méthodologie associée. Flot de simulation
Exercice : Mise en œuvre des BFM's Avalon & AXI avec simulation par Modelsim-AE.
- **Mise au point en utilisant la System-Console**
SystemConsole et API Tcl associée. Introduction au Jtag_Master & USB Debug Master. Services, API, Flot, Monitoring. Exemples complets. Le dashboard et ses widgets graphiques.
Exercice : Debug sur carte de la fabric à travers le JTAG par SystemConsole.
- **Méthodes Avancées de Mise au Point pour plateforme Qsys & Nios II**
Hub JTag, mise au point distante (Remote jtag server), outils optionnels, debug multiprocesseurs/ multi-threads, Mesure de performance (Timer HiRes, Perf Counters & Profiler), interprétation.
Exercice : Utilisation du Profiler et des Performance Counters
- **Introduction à l'écriture de Pilotes (Device Drivers) Nios II (optionnel)**
L'interfaceHAL API, drivers et intégration, initialisation et API spécifiques, écriture d'un wrapper *_sw.tcl
Exercice : Écrire un pilote (driver) caractère puis utiliser le périphérique en tant que stdout.