

Utilisation d'IP pour la conception de systèmes de télécommunications sur une plateforme de prototypage Excalibur-NIOS

S. Crand^{* ✉}, S. Gauthier^{**}, F. Nouvel-Uzel^{***} et S. Le Nours^{****}
(Université de Rennes 1, INSA de Rennes)

Mis en ligne le 9 septembre 2003.

Résumé

Cet article présente un projet destiné aux étudiants d'école d'ingénieurs, de DESS, d'IUT électronique ou de section de techniciens supérieurs (STS) en électronique. Suivant le profil et le niveau des étudiants ce projet pourra être présenté de différentes manières. Il a pour but de familiariser les étudiants à la conception de circuits numériques complexes de type SoC via l'utilisation d'un flot de conception adapté. L'objectif est de développer un bloc de traitement dédié à un système de télécommunications à partir d'IP (Intellectual Property). Le projet est basé sur l'utilisation de la carte de prototypage Excalibur-NIOS d'ALTERA.

Mots-clés : SoC, conception de systèmes numériques, IP, co-design.

© EDP Sciences, 2003.

Niveau de connaissances requis. Circuits logiques programmables, VHDL, communications numériques (filtrage numérique, modulation, transposition de fréquences).

Niveau des étudiants. École d'ingénieurs, DESS.

* **Samuel Crand**¹ a obtenu un diplôme d'ingénieur de l'INSA de Rennes en électronique et radiocommunications et un DEA d'électronique en 1994. En 1997, il a obtenu le doctorat d'électronique de l'Université de Nantes. Depuis 1998, il est maître de conférences à l'Université de Rennes 1 au sein de l'UFR Sciences et Propriétés de la Matière et effectue son activité de recherche au sein de l'Institut d'Électronique et de Télécommunications de Rennes (IETR).

Ses activités pédagogiques portent essentiellement sur la conception de systèmes analogiques et numériques. Il enseigne principalement en DESS Composants Électroniques, Maîtrise EEA et DIIC (Diplôme d'Ingénieur en Informatique et Communication).

Ses travaux de recherche au sein de l'IETR UMR CNRS 6164 portent principalement sur le développement de nouvelles méthodologies de conception et de prototypage rapide de systèmes de télécommunications mixtes (analogiques/numériques) et sur le développement d'une plateforme matérielle et logicielle pour les systèmes de télécommunications reconfigurables (radio logicielle).

✉ e-mail : Samuel.Crand@univ-rennes1.fr (auteur de correspondance)

** **Sébastien Gauthier**^{1,2} a été reçu à l'agrégation de génie électrique en 2001 et a obtenu un DEA d'électronique de l'Université de Rennes 1 en 2002. Il enseigne actuellement au Lycée Carnot à Bruay-la-Buissière (62).

*** **Fabienne Nouvel-Uzel**³, ancienne élève de l'INSA de Rennes, a obtenu son doctorat d'électronique de l'INSA de Rennes en 1994. Depuis octobre 1995, elle enseigne dans les domaines de l'électronique numérique, du traitement du signal, des composants intégrés, de l'architecture et des réseaux au niveau école d'ingénieurs à l'INSA de Rennes et DEA d'électronique à Rennes.

Ses thèmes de recherche sont centrés sur l'étude des systèmes électroniques pour des applications de communications embarquées ou non à haut débit. Ces activités s'insèrent dans le cadre de la radio logicielle et des systèmes de transport. Ses préoccupations portent sur l'étude des architectures adaptées aux transmissions de type étalement de spectre, multi-porteuses et qui soient reconfigurables. Ce dernier point est un élément important dans les systèmes embarqués (automobile, mobile) mais aussi fixes (supports de plusieurs normes). Elle occupe actuellement un poste de maître de conférences à l'INSA de Rennes dans le département ESC (Électronique et Systèmes de Communications) et est responsable de la quatrième année ESC.

**** **Sébastien Le Nours**³ prépare une thèse de doctorat en électronique à l'IETR (INSA Rennes). Ses thèmes de recherche portent sur le développement de nouveaux systèmes de radiocommunication pour les futures générations de mobiles. Il a notamment étudié les problèmes liés à l'adéquation entre les algorithmes développés et des architectures hétérogènes de prototypage.

¹ Institut d'Électronique et de Télécommunications de Rennes, Université de Rennes 1, Campus de Beaulieu, F-35042 Rennes, France.

² Lycée Carnot, 148 rue Alfred Leroy, BP 50, F-62701 Bruay La Buisnière, France.

³ Institut National des Sciences Appliquées (INSA), 20 av. des Buttes de Coësmes, F-35043 Rennes, France.

1. Introduction

L'utilisation de composants virtuels (IP) pour la conception de systèmes complexes permet de démocratiser ce domaine. En effet, l'association d'IP offre désormais la possibilité de concevoir des systèmes complexes sans pour autant nécessiter un temps de développement prohibitif.

Ce projet est basé sur l'utilisation de la carte de prototypage Excalibur d'ALTERA [1]. Cette carte de développement dispose d'un composant programmable APEX dans lequel il est possible d'implanter des SoCs de densité modeste. On parle alors d'un système sur une puce programmable (SOPC). Le principe de la conception SoC est de développer sur une même puce un système complet pouvant inclure des processeurs, des modules de traitements spécifiques, des liens de communication.

Dans un premier temps, les étudiants devront mettre en oeuvre le coeur de processeur NIOS. Ce processeur se présente sous la forme d'un IP soft. La configuration du processeur sera alors optimisée en fonction des exigences de l'application cible. Ensuite, les étudiants utiliseront des IPs d'ALTERA, des modules paramétrables et des blocs propriétaires décrits en VHDL pour constituer les unités de traitement du système.

Les blocs IP sont des fonctions réutilisables et personnalisables. L'utilisation d'IP permet au concepteur de se concentrer sur la valeur propriétaire ajoutée à sa conception. Ainsi, la rentabilité des processus de conception est accrue.

Pour ce projet, les IPs utilisés sont de type *OpenCore Plus* [2]. L'intérêt d'une telle utilisation est la gratuité temporaire de ces IP et la possibilité de les simuler fonctionnellement et de les implanter physiquement dans le composant programmable. Une fois implanté dans le composant, la durée de fonctionnement de l'IP est limitée. Cette durée est suffisamment grande pour permettre aux étudiants d'évaluer les performances de l'application.

Ce type de projet est destiné à deux types de population, la première étant des étudiants d'IUT ou de STS et la seconde des étudiants en fin d'études au niveau bac+5. Dans la suite de cet article, seul l'enseignement en DESS et en école d'ingénieurs sera présenté. En effet, suivant le parcours des étudiants le contenu pédagogique et les objectifs d'un tel projet sont clairement différents.

2. La carte de prototypage Excalibur

La carte de prototypage (Fig. 1) fait partie d'un kit de développement comprenant tous les outils logiciels et matériels permettant de concevoir un SoC programmable dans le composant APEX d'ALTERA. Cette carte permettra aux étudiants d'évaluer le champ d'application et les performances du coeur microprocesseur NIOS.

Le kit de développement contient entre autre l'IP du coeur microprocesseur NIOS et ses périphériques, la chaîne de développement logicielle complète (*GNUPro*, *QuartusII*, *Leonardo Spectrum*) et la carte de prototypage (FPGA de type APEX EP20K200E, mémoire, ports de communication, afficheur LCD...).

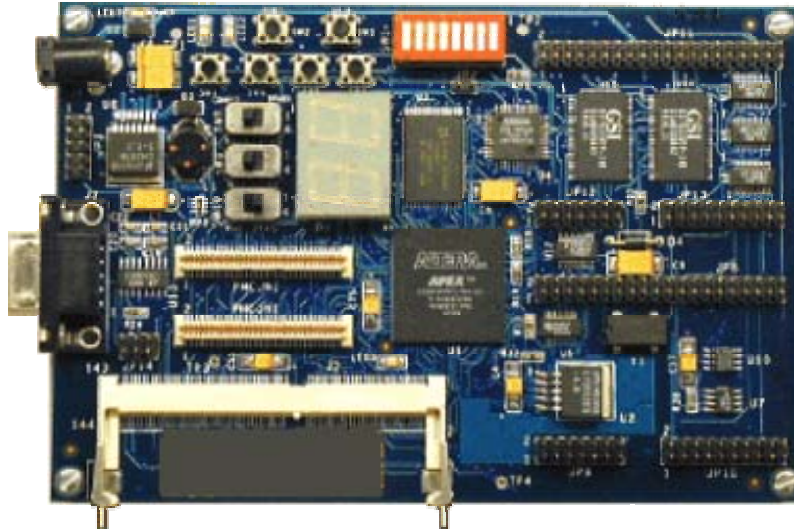


Fig. 1. La carte de prototypage Excalibur
(plateforme disponible via le CNFM - Comité National de Formation en Microélectronique -, <http://web.cnfm.fr/ALTERA>).

3. Le système à implanter

Le système choisi comme support du projet permettant d'illustrer les possibilités d'une conception orientée SoC est un modulateur numérique des voies I et Q pour un modem OFDM [3]. Le principe de ce module est de traiter numériquement la translation en fréquence des deux voies I et Q et d'utiliser qu'un seul convertisseur analogique.

Le modulateur numérique (Fig. 2) permet de pallier les inconvénients du modulateur analogique à savoir un défaut de quadrature en FI et un défaut d'équilibre (phase et gain) du parallélisme des voies I et Q [3]. Ceci est rendu possible grâce au passage à une seule voie analogique.

On ne dispose plus alors que d'un convertisseur numérique-analogique fonctionnant à la fréquence $2Fe$. Un multiplexeur prélève les échantillons de la voie I aux instants nTe et les échantillons de la voie Q aux instants $nTe + Te/2$. Les échantillons sur la voie Q n'étant présents qu'aux instants nTe , il faut créer les échantillons aux instants $nTe + Te/2$ en interpolant les échantillons connus aux instants nTe . Cette opération est effectuée par filtrage. La latence introduite par le filtre est compensée au niveau de la voie I.

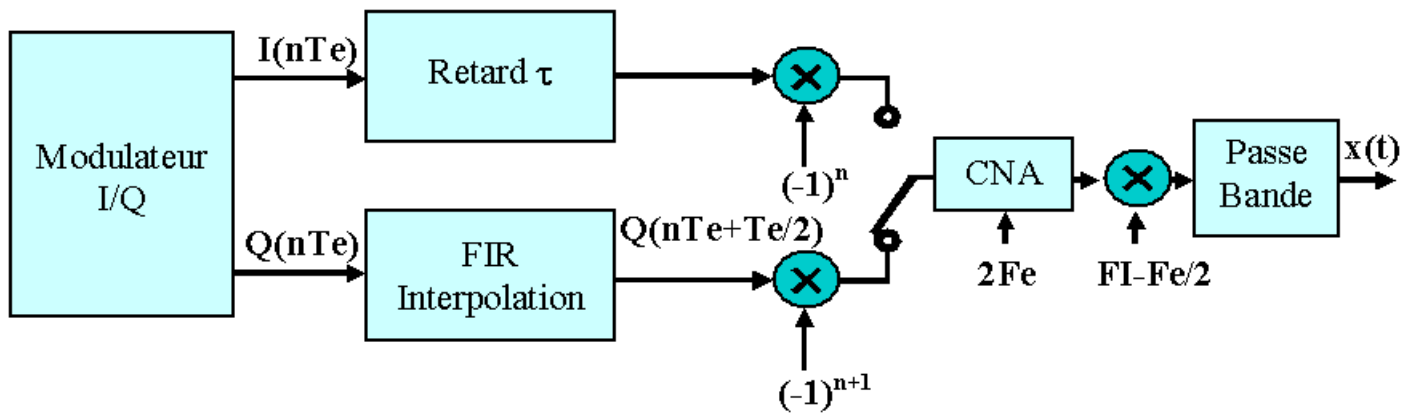


Fig. 2. Le schéma du modulateur numérique.

Dans un premier temps, les étudiants auront pour tâche d'étudier et de mettre en oeuvre le filtre d'interpolation avec l'IP *FIR Compiler* [4].

3.1. Organisation de l'enseignement

Vu l'aspect pluridisciplinaire d'un tel projet, celui-ci nécessite pour les étudiants de nombreux acquis. L'application visée étant un module s'intégrant dans un système de télécommunications, les aspects filtrage numérique, modulation et transposition de fréquences devront être assimilés. L'objectif principal de ce projet est d'illustrer les enseignements portant sur la conception de systèmes matériels/logiciels. De ce fait, les étudiants auront suivi un enseignement sur les circuits logiques programmables, le VHDL, l'informatique embarquée. L'utilisation des outils dans ce projet ne nécessite pas obligatoirement de préacquis étant donné que la prise en main du flot de conception est aussi un des objectifs de ce projet.

Comme précisé précédemment, ce projet a pour but de sensibiliser les étudiants à la conception d'un système complet comprenant des modules matériels et logiciels. Un tel projet permet aussi de mettre à profit le travail en équipe. En effet, il peut être judicieux de décomposer ce système en sous-modules effectués par binôme. Cette approche nécessite donc un travail de présentation générale du projet et des objectifs, un partitionnement au niveau système, une phase de conception distincte pour chaque module puis une mise en commun pour permettre la validation du système conçu.

Ce projet d'une durée de douze heures peut être effectué de la manière suivante :

Durée	Travail	Résultats attendus
2 heures	Présentation générale	- Compréhension du système à concevoir - Partitionnement système
7 heures	Travail par binôme (un projet : 3 binômes) - FIR - NIOS - Protocole de communications	- Conception et validation de chaque module - Exploration architecturale (étude des différentes structures du FIR par exemple)
1 heure	Présentation de 15 min par binôme + 15 min de cadrage par l'enseignant	- Mise en commun du travail
2 heures	Implantation de l'ensemble du système	- Validation de l'ensemble

Tab. 1. Exemple d'organisation du projet.

3.2. Mise en oeuvre du filtre d'interpolation avec *FIR Compiler*

Le *MegaCore FIR Compiler* permet de concevoir un large éventail de filtres de type FIR. Son emploi simple et rapide réduit considérablement le temps de conception de filtre.

Le filtre d'interpolation à concevoir devra vérifier les caractéristiques suivantes :

- $F_e = 20 \text{ MHz}$;
- $F_c = 10 \text{ MHz}$ (filtre demi-bande).

Le *FIR Compiler* se présente sous la forme d'une interface graphique appelée *MegaWizard* qui guide pas à pas l'utilisateur dans les choix de la configuration du filtre. Trois types de paramètres peuvent être définis : les paramètres d'utilisation (chargement des coefficients), de spécification (nombre de coefficients...) et d'implantation (type d'architecture).

Une étape de ce travail consistera donc à évaluer différentes configurations en termes de vitesse, de ressources matérielles nécessaires ou d'efficacité fonctionnelle dans le but de sélectionner la configuration la mieux adaptée à l'application envisagée. Par exemple, l'implantation d'un filtre de type parallèle constitué de huit coefficients codés sur 10 bits représente environ 11 % de la surface du circuit APEX 20K200E.

3.3. Mise en oeuvre du SoC

La conception du modulateur numérique est réalisée en fonction des contraintes imposées par le système global. Pour cette raison, il est nécessaire d'ajouter au processeur NIOS et au modulateur numérique des blocs supplémentaires.

Le processeur NIOS gère l'interface entre le PC et le modulateur numérique. En effet il permet d'appliquer les vecteurs de test au modulateur et de transmettre les données traitées au PC pour qu'elles puissent être visualisées et comparées aux données issues de la simulation. Le processeur NIOS peut aussi avoir d'autres fonctions comme la remise à jour des coefficients du filtre d'interpolation, calculés par exemple en fonction des signaux d'entrée.

Le schéma du système complet est présenté ci-dessous (Fig. 3) :

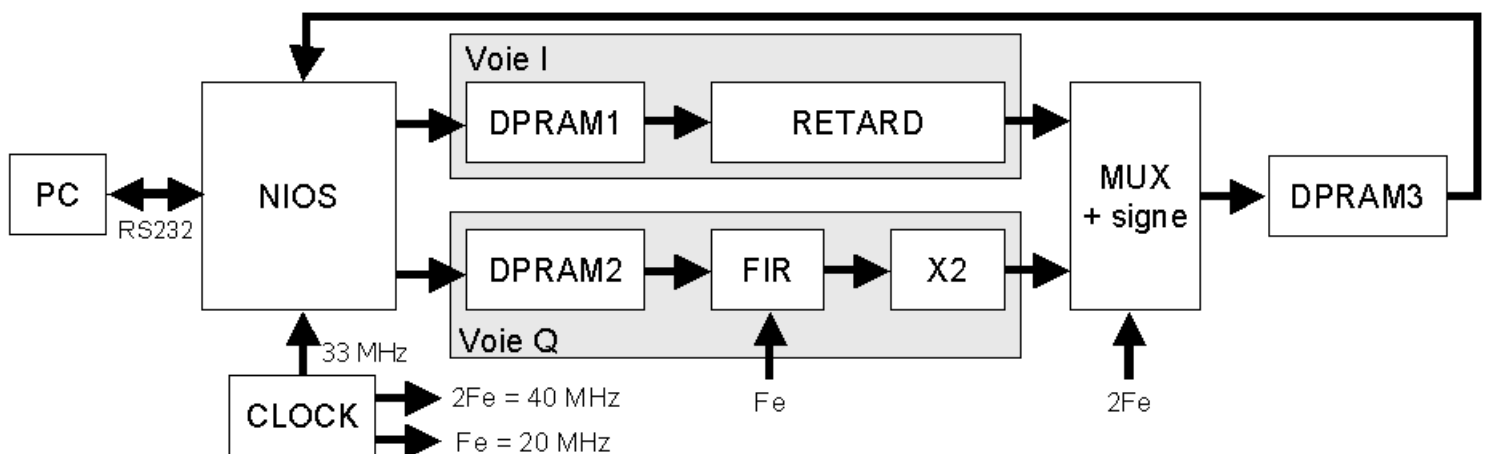


Fig. 3. Le schéma structurel du système.

Les fonctionnalités de chaque bloc peuvent être résumées de la manière suivante :

- Le bloc *CLOCK* permet de générer les deux horloges permettant ainsi de cadencer les différents éléments du système.

- Les mémoires *DPRAM1* et *DPRAM2* permettent de faire cohabiter dans le système final le modulateur et le processeur NIOS. En effet, le processeur NIOS fonctionne à une fréquence de 33,33 MHz (fréquence fixée sur la carte de prototypage Excalibur). Le filtre d'interpolation fonctionne à la fréquence de 20 MHz. Le processeur NIOS ne peut donc pas gérer directement les signaux à fournir et à recevoir du filtre d'interpolation. Le même problème se pose pour lire les données en sortie du multiplexeur qui fonctionne à 40 MHz. Pour y remédier, une solution est d'intercaler des mémoires tampons entre le processeur NIOS et le système. Les DPRAMs sont des RAMs avec double port d'entrées : un port pour la lecture et un port pour l'écriture. L'intérêt de la DPRAM est de pouvoir lire et écrire à des cadences différentes. La taille de chaque mémoire est de 1024 mots de 10 bits.
- Le bloc *RETARD* permet de compenser le retard introduit par le temps de latence. Ce bloc sera réalisé par les étudiants en langage de description VHDL.
- Le *FIR* utilisé comme filtre d'interpolation permet de générer les échantillons $Q(nT_e + T_e/2)$.
- Le multiplieur *X2* permet d'équilibrer la dynamique sur les deux voies. En effet, la dynamique des signaux en sortie du filtre d'interpolation est deux fois plus faible que celle des signaux d'entrée. Ce bloc de faible complexité sera également réalisé par les étudiants en VHDL.
- Le bloc *MUX* permet de multiplexer les signaux $I(nT_e - \tau)$ et $Q(nT_e + T_e/2 - \tau)$ à la fréquence de 40 MHz. La sortie du multiplexeur prend alternativement la valeur de I et de Q au rythme de $2F_e$. Les blocs de multiplication $(-1)^n$ et $(-1)^{n+1}$ de remise en forme des signaux sont intégrés dans ce module qui sera spécifié en VHDL.
- Le bloc *DPRAM3* permet de récupérer les données de sortie du modulateur. Ce bloc fait l'interface entre le NIOS et le modulateur. Les données de sortie du modulateur sont écrites dans la *DPRAM3* à la vitesse de $2F_e$.

4. Le flot d'utilisation

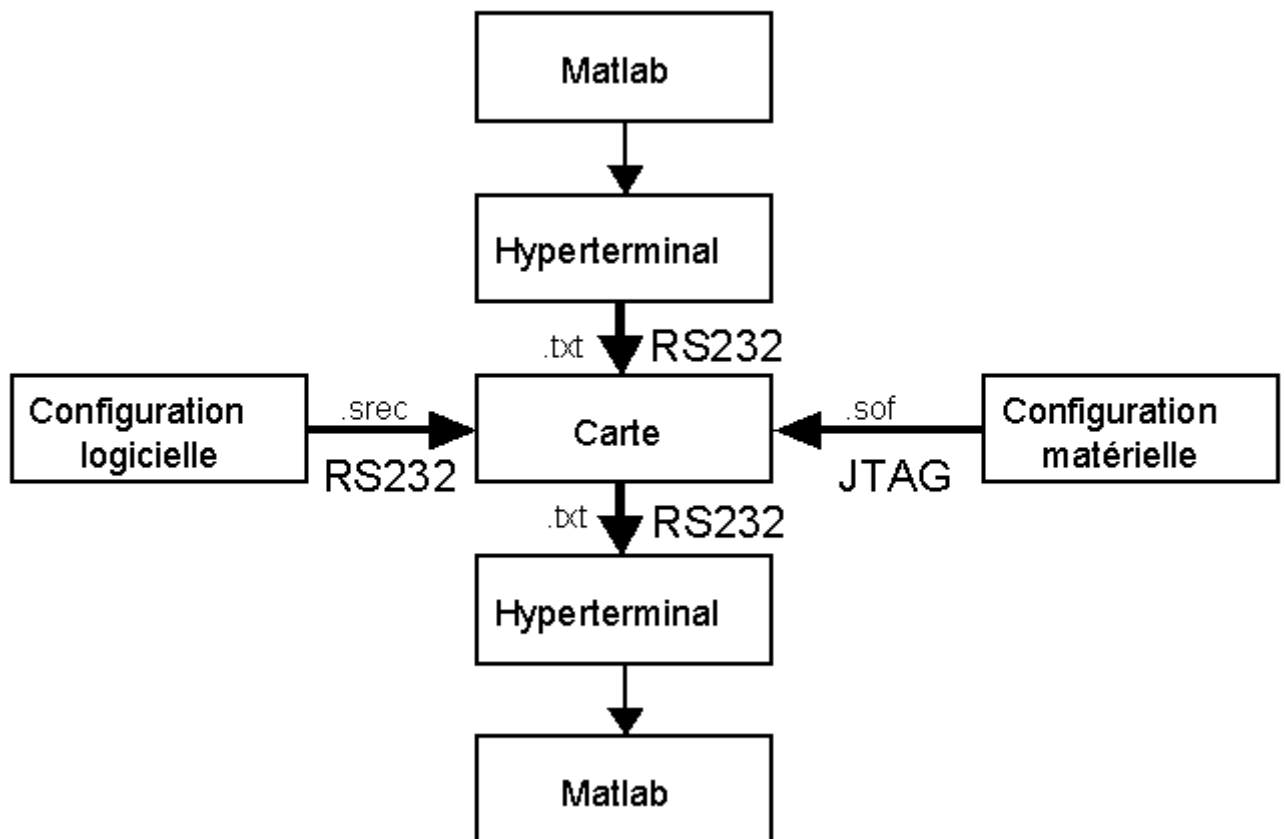


Fig. 4. Le flot d'utilisation de la carte de prototypage.

Le flot d'utilisation (Fig. 4) de la carte de prototypage suit les étapes suivantes :

- première étape : le SoC créé avec le logiciel *QuartusII* est implanté dans le composant APEX via le port JTAG ;
- deuxième étape : programmation du processeur NIOS du SoC via la liaison série RS232 ;
- troisième étape : création avec le logiciel *Matlab* des signaux d'entrée destinés au SoC, transmis à la carte via le logiciel *Hyperterminal* de *Windows* (gestion de la liaison série RS232) ;
- quatrième étape : utilisation de *Hyperterminal* pour récupérer les données traitées par le SoC. Ces résultats seront par la suite traités, visualisés et analysés avec *Matlab*.

Le choix d'un logiciel tel que *Matlab* se justifie principalement par le fait que cet outil est utilisé dans d'autres modules d'enseignement.

5. Le flot de conception du SoC

La figure 5 présente le flot de conception du SoC utilisé dans le cadre de ce projet. L'objectif final est d'obtenir une description de type *.sof* pour l'implantation matérielle (NIOS et blocs logiques) et un fichier de type *.srec* pour l'implantation logicielle (exécutif pour le processeur NIOS).

Ce flot est constitué de différents outils tels que *QuartusII* (spécifications schématiques, placement/routage, simulation logique...), *ModelSim* (simulation VHDL), *GNUPro* (compilation C) et *Leonardo Spectrum* (synthèse logique).

Ce projet pouvant être divisé en trois modules, chaque binôme n'utilisera pas obligatoirement l'ensemble de ces outils. Par exemple, pour le développement du module logiciel (microprocesseur NIOS) seuls *Quartus* et *GNUPro* seront utilisés. Par contre dans le cadre de la mise en commun, il est nécessaire que chaque étudiant puisse être sensibilisé d'une part aux interactions entre les modules matériels/logiciels et d'autre part aux liens entre les outils de CAO.

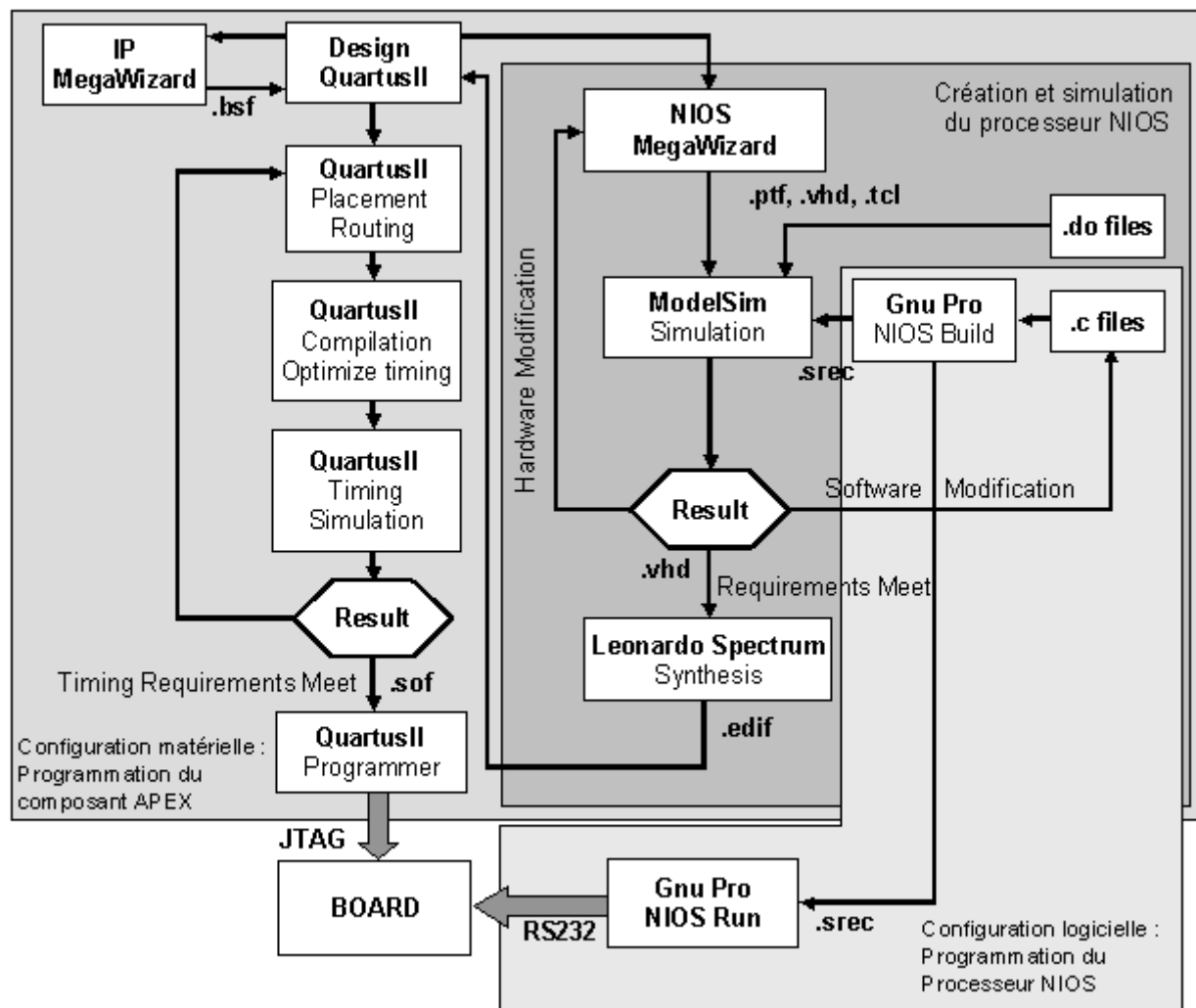


Fig. 5. Le flot de conception d'un SoC.

6. Conclusion

Cet article présente un projet qui s'inscrit dans un module d'enseignement sur la conception de systèmes numériques. L'objectif de ce projet est de proposer aux étudiants de concevoir un système complet en utilisant des blocs pré-compilés issus des bibliothèques ALTERA, des blocs spécifiés par les étudiants et des composants virtuels dits IP. Le principe d'utilisation des *IP OpenCore Plus* permet de les utiliser gratuitement d'un point de vue fonctionnel mais aussi de les implanter physiquement dans le circuit. De plus, l'utilisation de ce type de flot de conception leur permet d'utiliser divers outils et d'appréhender les problèmes liés à la conception mixte logicielle/matérielle. Le système proposé est implanté sur une plateforme de type système sur puce programmable.

De plus ce projet pouvant être effectué par trois binômes en parallèle (Tab. 1), une part non négligeable de leur travail consiste à communiquer entre eux pour que la mise en commun puisse être réalisée le plus efficacement possible. En effet, chaque binôme réalise une partie distincte de manière quasi indépendante des deux autres puis une enveloppe englobant ces trois parties est spécifiée et conçue pour l'obtention et la validation du système complet.

Ce projet permet aux étudiants d'utiliser différents outils composant un environnement de conception de systèmes mixtes, de suivre une méthodologie descendante précise, d'implanter, de caractériser et de valider un système numérique complet sur une plateforme de prototype.

Références bibliographiques

- [1] S. Gauthier, *Intégration de fonctions de communications numériques sur une plateforme de prototypage Excalibur-Nios*, mémoire de DEA, IETR, Rennes (2002).
- [2] Altera, *OpenCore Plus hardware evaluation of MegaCore functions, version 1.1*, Application Note 176 (avril 2002), <http://www.altera.com/literature/an/an176.pdf> (consulté le 4 septembre 2003).
- [3] J.F. Helard et D. Castelain, *Digital I and Q for OFDM*, Electron. Lett. **35** (1995) 2081.
- [4] <http://www.altera.com> (consulté le 4 septembre 2003).