

## Mise en oeuvre de l'outil SynDEx pour la conception et implantation de systèmes sur plateforme hétérogène

F. Nouvel-Uzel <sup>\* ✕</sup>, S. Le Nours <sup>\*\*</sup> et I. Herman <sup>\*\*\*</sup> (IETR/INSA de Rennes ; ENSTA)  
Mis en ligne le 30/04/2004.

### Résumé

Cet article présente l'outil de conception SynDEx, utilisé pour l'implantation d'applications de télécommunications sur des plateformes matérielles utilisant des processeurs de signaux (Digital Signal Processor) et processeurs matériels (Field Programmable Gate Array). L'outil SynDEx repose sur la méthodologie Adéquation Algorithme Architecture, AAA ou A<sup>3</sup> développée à l'INRIA. Elle permet, à partir d'un graphe décrivant les algorithmes et d'un graphe représentant l'architecture cible, d'exprimer tout le parallélisme potentiel dans l'exécution des différentes fonctions et de trouver une implantation optimale des algorithmes sur une architecture hétérogène. L'article a pour objectif de familiariser les étudiants avec les notions de conception dites de Co-Design et de présenter les potentialités de cet outil. Le premier paragraphe de l'article situe le contexte de l'étude. La méthode A<sup>3</sup> est ensuite présentée. L'outil SynDEx est abordé au travers du projet étudiant.

**Mots-clés** : méthodologie de conception, conception numérique, architecture hétérogène.

© EDP Sciences, 2004.

---

**Niveau de connaissances requis.** Notions sur les circuits logiques programmables, langage VHDL, communications numériques, traitement de signal (fonction FFT/IFFT).

**Niveau des étudiants.** École d'ingénieurs dernière année d'étude, DESS.

---

\* **Fabienne Nouvel-Uzel** <sup>1,2</sup>, 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 ASIC/FPGA, des techniques de communication et 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 standards). 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.

✉ e-mail : fabienne.nouvel@insa-rennes.fr

\*\* **Sébastien Le Nours** <sup>1,2</sup> a soutenu sa thèse de doctorat en électronique à l'IETR (INSA Rennes) le 15 décembre 2003. Le titre de son mémoire est « *Étude, optimisation et implémentation de systèmes MC-CDMA sur architectures hétérogènes* ». Il travaille 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.

e-mail : sebastien.le-nours@insa-rennes.fr

\*\*\* **Ivan Herman**<sup>1,2,3</sup> prépare une thèse de doctorat en électronique à l'ENSTA de Paris, au laboratoire LEI. Ses thèmes de recherche portent sur le développement d'une plateforme de prototypage pour les System On Chip (SOC) dans le cadre des applications Radio Logicielle. Son responsable de thèse est Omar Hammani de l'ENSTA et le directeur de thèse est Dominique Houzet de l'IETR.

<sup>1</sup> Institut d'Électronique et de Télécommunications de Rennes (IETR), Université de Rennes I, Campus de Beaulieu, F-35042 Rennes, France.

<sup>2</sup> Institut National des Sciences Appliquées (INSA), 20 av. des Buttes de Coësmes, F-35043 Rennes, France.

<sup>3</sup> École Nationale Supérieure des Techniques Avancées (ENSTA), 33 bd Victor, F-75739 Paris, France.

---

## 1. Introduction

L'évolution des nouveaux standards de télécommunication, les besoins de hauts débits pour les applications multi-média, font apparaître un nouveau concept dit de Radio Logicielle ou Software Radio [1].

Pour satisfaire aux nombreuses contraintes des nouvelles normes (débit, reconfigurabilité, multi-modes,...), une architecture support hétérogène doit être considérée. Le terme hétérogène désigne une plateforme accueillant des processeurs logiciels et matériels (DSP (Digital Signal Processor), processeurs, FPGA (Field Programmable Gate Array), ASIC (Application Specific Integrated Circuit)). Le terme mixte désignera la cohabitation de composants numériques et analogiques sur une même plateforme. Ces architectures permettent en effet :

- de combiner les fonctionnalités offertes par les DSP telles que les instructions Multiply and Accumulate (MAC), le parallélisme spatial des unités de traitement ;
- avec celles de composants FPGA, qui offrent à la fois du parallélisme spatial (des fonctions indépendantes se déroulent en parallèle) et du parallélisme temporel (optimisation du temps d'exécution).

Par ailleurs, les composants virtuels IP (Intellectual Property) sont de plus en plus utilisés dans les systèmes complexes, réduisant ainsi les durées de conception, tant sur DSP que sur FPGA.

Lors du portage d'algorithmes sur une architecture hétérogène, l'utilisateur devra déterminer quelles sont la répartition et la distribution optimales de ceux-ci sur les différentes cibles. Aussi, l'utilisateur doit-il disposer d'un outil permettant de le guider dans cette Adéquation Algorithme-Architecture. L'utilisation de ces architectures hétérogènes induit de nouvelles considérations lors des spécifications du système, notamment :

- une description des algorithmes indépendante de la cible matérielle ;
- une description de l'architecture générique et modulaire ;
- une connaissance a priori des temps d'exécution selon la cible (issus de tests unitaires et d'expériences) ;
- la prise en compte des temps de transfert entre composants pour l'estimation des performances globales du système.

Les approches « Co-Design » [2] visent à orienter le concepteur vers une solution Algorithme-Architecture optimale. Différentes méthodes et outils existent (VCC, POLIS, COMES/MCSE, ...) et notamment la méthodologie Adéquation Architecture Algorithme, AAA ou A<sup>3</sup> et son outil associé SynDEx [3]. Utilisée dans les applications de traitement d'image, il semble intéressant d'évaluer les possibilités de A<sup>3</sup>/SynDEx dans d'autres domaines, et plus particulièrement celui des télécommunications.

Le projet proposé aux étudiants est basé sur l'utilisation de l'outil SynDEx pour la conception d'un système de communication utilisant la technique MC-CDMA [4]. L'objectif est d'illustrer l'enseignement des méthodologies de conception pour l'implantation de systèmes sur des plateformes conçues autour de DSP et FPGA. Ce projet est principalement destiné aux étudiants de niveau BAC+5 (DESS, DEA ou 5<sup>e</sup> année ingénieur). Il peut être adapté à d'autres étudiants n'ayant pas tous les pré-requis, notamment en communications numériques. Dans un premier temps, les étudiants disposent d'une description simplifiée du système à implanter. Ils modéliseront celui-ci dans l'environnement SynDEx, tant du point de vue algorithme que matériel. Puis, ils devront optimiser le système, selon les critères temporels. À l'issue de l'adéquation, les codes générés pour chaque fonction seront implantés sur la plateforme.

## 2. La méthodologie Adéquation Algorithme Architecture

La méthodologie AAA/ A<sup>3</sup> [3] vise le prototypage rapide et l'implantation optimisée d'applications distribuées temps réel comprenant du traitement du signal et du contrôle. Elle est basée sur un modèle unifié de graphes utilisés, tant pour spécifier l'Algorithme que l'Architecture multi-composants, que pour déduire les implantations possibles en termes de transformations de graphes. Cette approche globale vise à éviter toute rupture entre les différentes phases du cycle de développement et permet des vérifications formelles et des optimisations.

Ces deux graphes expriment le parallélisme potentiel des algorithmes et le parallélisme effectif de l'architecture. La méthodologie A<sup>3</sup> recherche un ordonnancement optimisé des fonctions ainsi qu'une distribution spatiale de celles-ci. L'Adéquation est un problème d'optimisation qui consiste à choisir une implantation des algorithmes sur l'architecture dont les performances respectent des contraintes de temps. Les performances sont déduites des caractéristiques des composants, des temps d'exécution des fonctions et de communication entre les différents composants. L'utilisateur peut intervenir sur le résultat de l'adéquation soit en modifiant les différents graphes, soit en ajustant les temps respectifs.

## 3. L'outil SynDEx

Le logiciel SynDEx, **S**ynchronised **D**istributed **E**xecutive, supporte la méthodologie A<sup>3</sup>, pour le prototypage rapide optimisé et le Co-Design d'applications temps réel. C'est un outil totalement gratuit, disponible à l'adresse suivante : <http://www-rocq.inria.fr/syndex/download.htm>. Il s'exécute dans l'environnement Windows ou Unix. Le concepteur spécifie ses graphes et les paramètres liés aux composantes de ceux-ci. À l'issue de l'adéquation, un macro-code générique, indépendant de la cible, est généré. C'est un exécutif distribué, statique (pas d'utilisation d'Operating System (OS)), optimisé pour les processeurs de traitement du signal et également pour les FPGA.

Appliqué aux DSP, le macro-code généré est traduit en langage C. L'allocation des ressources statiques, réduit les « sur-coûts » introduits par un OS temps réel et permet de prévoir le temps d'exécution d'une séquence d'opérations et donc les performances du système.

Appliqué aux FPGA, le macro-code généré est traduit en code VHDL. Comme pour le DSP, l'ordonnancement et le déroulement séquentiel est respecté lors de la traduction. Les communications entre le FPGA et les autres cibles DSP/FPGA sont par ailleurs incluses dans le code VHDL généré.

## 4. Description du système étudié et optimisé pour une implantation sur une plateforme hétérogène

Le système choisi pour le support de ce projet est la réalisation d'un modem MC-CDMA (Multi-Carrier, Code Division Multiple Access [4]). Ce système met en jeu des fonctions de contrôle, des algorithmes de communications numériques (codage/décodage, transformée de Fourier directe et inverse (FFT/IFFT)), estimation de canal, égalisation).

La modulation MC-CDMA associe les techniques d'accès multiple par répartition de codes (AMRC/CDMA) et modulation multi-porteuses (MC). L'AMRC associe à chaque utilisateur un code d'étalement, orthogonal aux autres codes. Le multiplex fréquentiel est constitué de canaux de fréquences orthogonales et de largeurs identiques. La bande totale de transmission est égale à  $N_c/T_d$ . Le symbole codé d'un utilisateur est transmis sur plusieurs porteuses en parallèle. Plusieurs utilisateurs peuvent alors transmettre en même temps, en utilisant des codes orthogonaux, sur l'ensembles des  $N_c$  porteuses.

L'expression du signal multi-porteuse  $x_j(t)$  de l'utilisateur  $j$ , pendant l'intervalle de temps  $[0, T_d[$ , dans une bande constituée de  $N_c$  sous-porteuses s'exprime selon l'équation (1) [5] :

$$x_j(t) = \Re \left\{ \sum_{k=0}^{N_c-1} d_j P(t) c_{k,j} e^{i2\pi f_k t} \right\}, \quad 0 \leq t \leq T_d \text{ avec } f_k = f_0 + \frac{k}{T_d}$$

$P(t)$  est la fonction de mise en forme du signal, généralement la fonction porte. Ce signal est échantillonné à la fréquence  $N_c/T_d$ , et conduit à exprimer l'échantillon  $n$  de l'enveloppe complexe  $X$  du signal  $x_j(t)$  modulé par :

$$X(nT_d/N_c) = \sum_{k=0}^{N_c-1} \frac{X_k}{\sqrt{N_c}} e^{2\pi i n(k-N_c/2)/N_c}$$

Cette expression de  $X$  correspond à la transformée de Fourier inverse du signal. Aussi, la modulation multi-porteuses est-elle réalisée pratiquement grâce à une transformée de Fourier inverse (IFFT). Dans le récepteur, on retrouve les fonctions duales de l'émetteur.

Le signal large bande va subir les effets du canal de propagation. Ceux-ci se matérialisent par des évanouissements sélectifs/non sélectifs en fréquence. Par ailleurs, le signal va subir plusieurs réflexions et réfractions dus aux obstacles, bâtiments, arbres, qui induisent des trajets multiples. Aussi, le récepteur va recevoir une combinaison de plusieurs répliques du signal, chaque réplique étant caractérisée par son atténuation et son retard. Il convient alors de corriger ces phénomènes par les fonctions d'estimation de canal (estimation des coefficients d'atténuation et de déphasage) et d'égalisation.

Dans le cadre de ce projet, les étudiants sont répartis en binômes ou trinômes. Les fonctions à assimiler sont différentes pour les groupes. Cependant, un intérêt plus particulier est porté aux opérations critiques: FFT/IFFT, égalisation. L'estimation du canal sera supposée parfaite.

Par ailleurs, les étudiants ont la possibilité d'évaluer le résultat de l'adéquation sur matériel. La carte de prototypage SUNDANCE [5] est constituée d'une carte mère, pouvant accueillir 4 modules, au format TIM. Dans ce projet, les élèves disposent de la configuration de carte suivante :

- deux DSP TMS320C67X, dont un dédié à la fonction canal de propagation ;
- un FPGA VIRTEX XCV400 ;
- des ports de communications pour les échanges DSP/FPGA, conformes au format d'échange TMS320C4X (C4X) à 20 Moctets/s ou SDB (Sundance Digital Bus) à 200 Moctets/s. Par la suite, seuls les liens C4X seront utilisés.

Les figures 1 et 2 représentent le système étudié et la carte cible avec les 2 DSP et le FPGA.

Fig. 1. Schéma du système MC-CDMA.

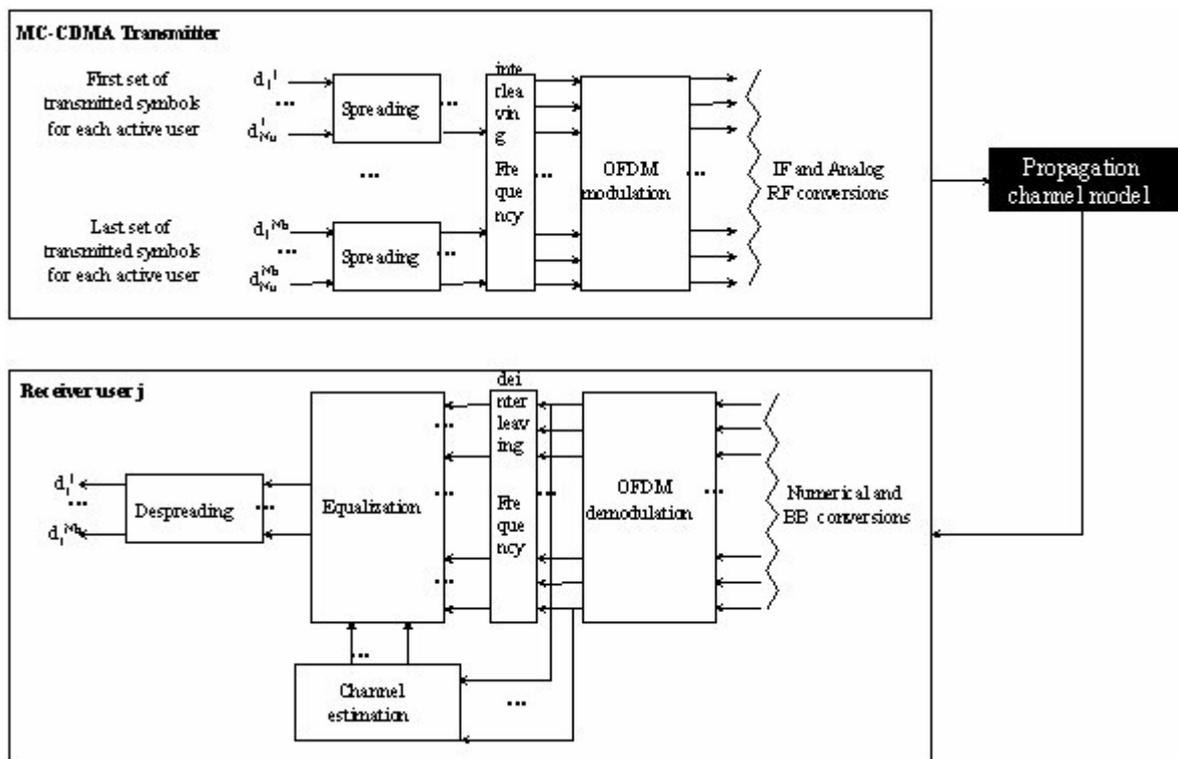
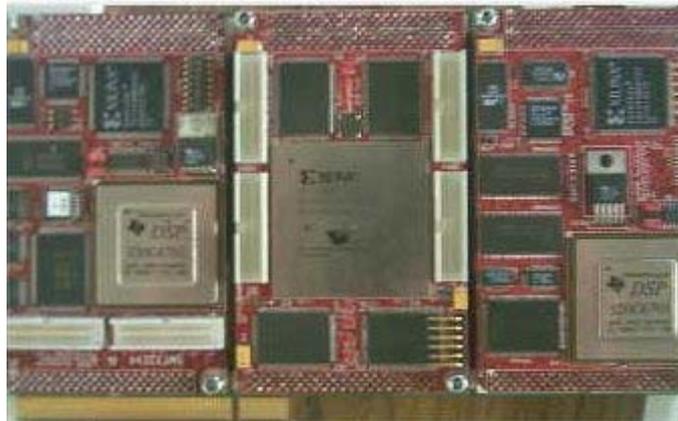


Fig. 2. La carte de prototypage Sundance.



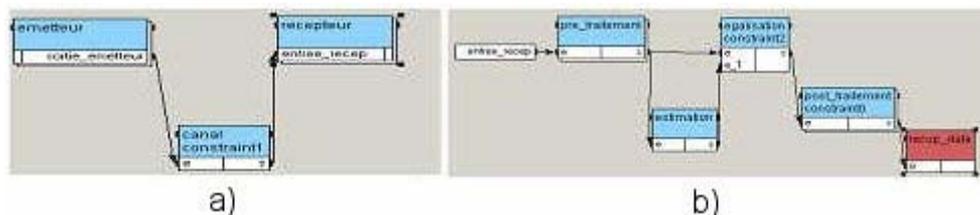
## 5. Déroulement du projet sous SynDEx

L'utilisateur interagit avec SynDEx par l'intermédiaire d'une interface graphique, de menus déroulants contextuels, simples et intuitifs. Trois menus spécifiques, Algorithm, Architecture et Adequation, permettent d'établir respectivement les graphes d'algorithmes, d'architecture, de les caractériser (durées des calculs et des communications), lancer l'adéquation, visualiser son résultat et enfin générer le macro-code de l'exécutif distribué. L'algorithme est traduit selon la cible et son déroulement est effectué tel que distribué et ordonné par l'heuristique.

### 5.1. Graphe d'algorithme

Après une présentation et étude du système, les étudiants modélisent le système dans l'environnement SynDEx. Pour créer le graphe, l'utilisateur représente chaque fonction par une opération, hiérarchique ou non. Chaque opération est caractérisée par : son nom, ses entrées et ses sorties, en précisant le type et la taille des données échangées. La cohérence du graphe est vérifiée à chacune des modifications et plus particulièrement les échanges de données entre opérations. Une entrée d'opération ne peut être connectée qu'à une seule sortie, portant le même type de donnée et ne créant pas de cycle de dépendance. Toute entrée doit être connectée. La figure 3 (a et b) illustre le graphe d'algorithme obtenu pour le système. On peut remarquer les niveaux hiérarchiques des opérations émetteur et récepteur. Il faut par ailleurs souligner que la description des opérations ne nécessitent pas la connaissance de l'algorithme (code source). On s'intéresse dans ce graphe uniquement à l'enchaînement des opérations, et non à l'exécution.

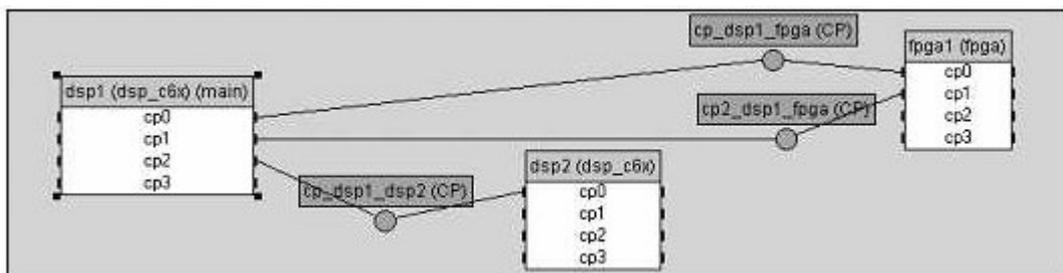
Fig. 3. a) Graphe du système. b) Graphe du module hiérarchique récepteur.



## 5.2. Graphe d'architecture

Comme précédemment, l'utilisateur modélise chaque processeur, DSP ou FPGA, par un opérateur et des médias de communication. Un opérateur peut aussi être composé de sous-graphes. Ces entités DSP ou FPGA n'exécutent qu'une seule tâche à un instant donné, ce qui correspond alors à une concurrence unitaire. Aussi, chaque opération du graphe d'algorithme placée sur un des composants sera exécutée séquentiellement selon l'ordonnancement obtenu suite à l'adéquation. La cohérence du graphe est vérifiée à chaque modification : deux opérateurs ne peuvent avoir le même nom, seuls les ports de types identiques peuvent être connectés par un média de communication de même type. Une vérification supplémentaire est effectuée avant le lancement de l'heuristique : le graphe d'architecture doit être connexe, il doit comporter un opérateur maître (*root*). La figure 4 présente le graphe d'architecture correspondant à la carte de prototypage.

Fig. 4. Graphe d'architecture de la plateforme cible.



## 5.3. Adéquation et visualisation du comportement séquentiel

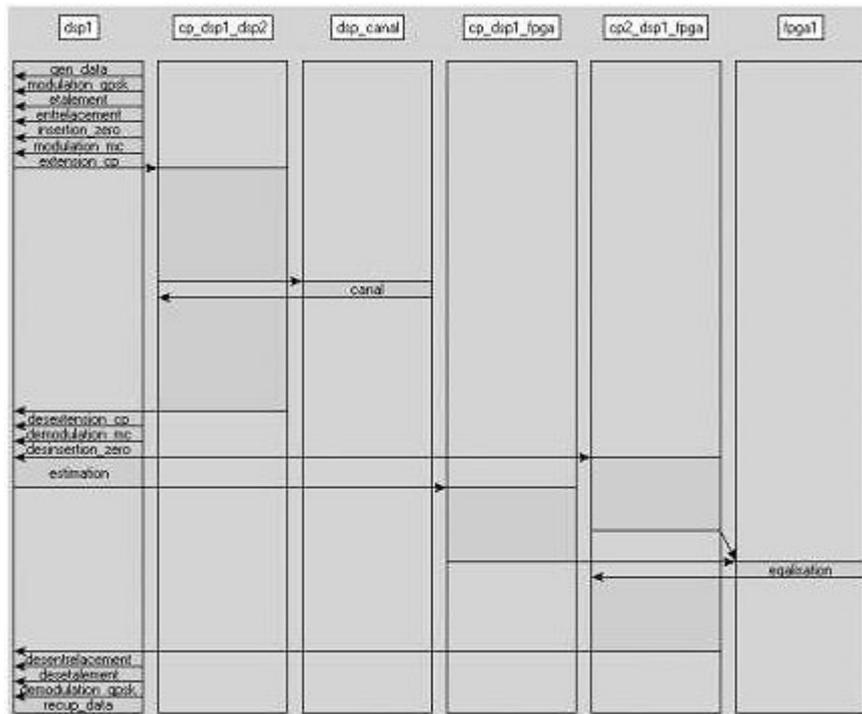
L'heuristique calcule les chemins critiques, selon les durées des opérations de calcul et de communication, définies pour les composants et les médias de communication. Elle va rechercher à minimiser la durée totale d'exécution du système. Il est possible de contraindre une fonction sur un opérateur, comme par exemple contraindre la fonction égalisation à être exécutée sur le FPGA.

À partir de ce résultat, il est possible de visualiser le diagramme temporel décrivant la prédiction du comportement des opérations réparties sur les opérateurs. Cette visualisation est importante car elle permet d'évaluer les performances de l'application, sans l'exécuter réellement sur la cible matérielle. Elle renseigne le concepteur sur la distribution des opérations sur les composants ainsi que sur l'ordonnancement des opérations de calcul et de communication.

Dans la figure 5, le diagramme temporel correspond à l'implantation du modem MC-CDMA et du canal sur les 3 processeurs. L'heuristique a distribué les opérations correspondant à l'émetteur sur le DSP1. Le canal est contraint sur le DSP2. Pour le récepteur, la fonction égalisation est implantée sur le FPGA, les autres fonctions sur le DSP1, les deux ports C4X sont actifs. Le temps prédit correspond à la somme des temps d'exécution de chaque fonction sur les différentes cibles et des temps de communication entre les différents composants. Il suffit de lancer la génération du macro-code pour chaque opérateur.

Au cours de cette étape, les étudiants doivent tester différentes configurations, et retenir la solution optimale. L'objectif de ces trois étapes est de montrer aux étudiants que l'Adéquation Algorithme-Architecture n'est pas un problème simple. Avec la complexité croissante des systèmes, seule l'expérience du concepteur n'est pas suffisante pour résoudre le partitionnement optimal. Avec SynDEX, la prédiction du comportement temps réel de l'application sera d'autant plus exacte que les modèles d'algorithmes, d'architectures et de communications seront précis.

Fig. 5. Diagramme temporel de l'exécution des fonctions sur les différents processeurs DSP/FPGA.



#### 5.4. Macro-code exécutif et traduction

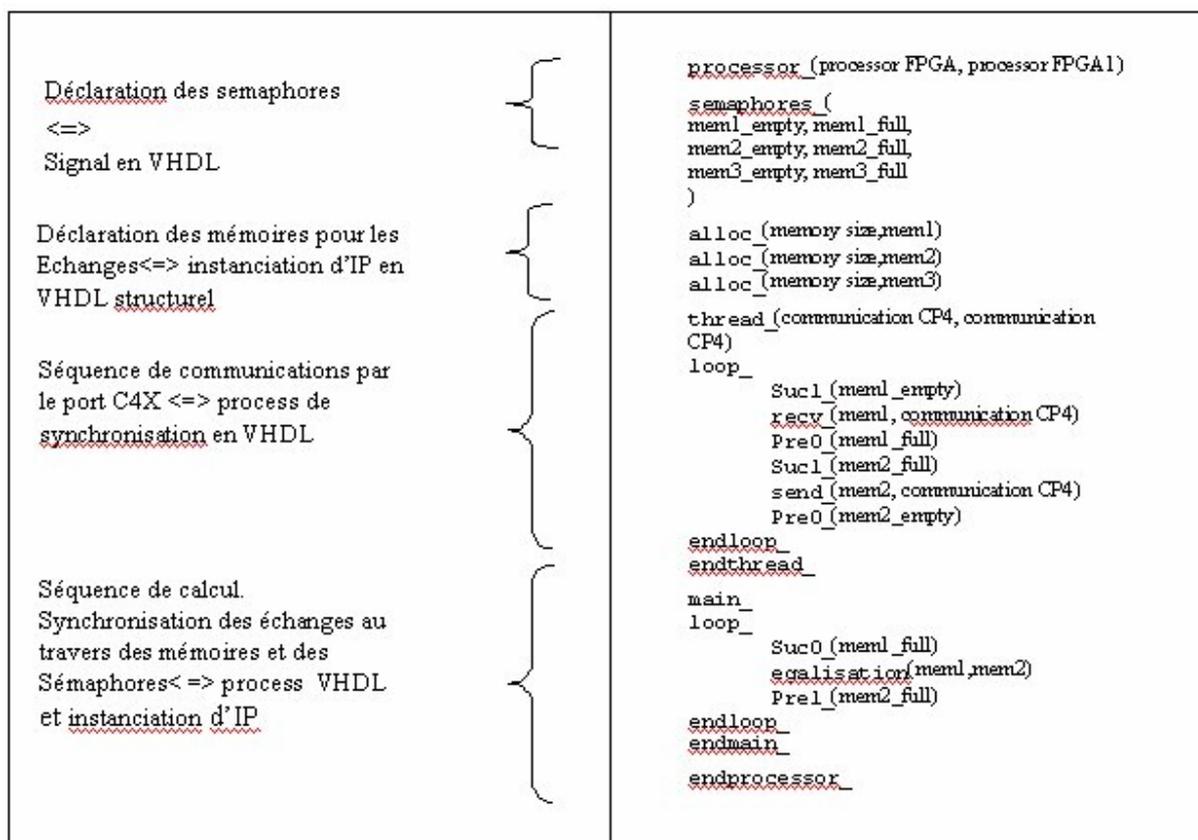
Le macro-code généré pour chaque type processeur possède la même structure, seuls diffèrent les appels des fonctions de calcul et de communication résultant de l'adéquation. L'exemple simplifié présenté figure 6 correspond au macro-code généré pour le FPGA, dans lequel la fonction égalisation est implantée. On peut remarquer trois parties importantes dans ce code :

- allocation des mémoires pour les échanges ;
- séquence de communication, qui gère les échanges entre composants ;
- séquence de calcul, dont l'activation dépend des mémoires d'entrée et de sortie.

Les deux séquences sont synchronisées à travers l'utilisation de sémaphores, assurant l'accès exclusif des mémoires partagées (macro Pre et Suc). Ce point est particulièrement important, car il met en avant l'aspect ordonnancement séquentiel et sans inter-blocage de l'exécutif. Puisque les exécutifs sont générés automatiquement, leurs codages et mises au point ne sont plus nécessaires. Cependant, le bon fonctionnement de chacune des fonctions de calcul et de communication, décrite dans le code cible, doit être testée unitairement avant intégration dans la bibliothèque.

La phase de traduction s'appuie sur un ensemble de bibliothèques, décrites pour chaque processeur (DSP ou FPGA), ainsi que pour chaque fonctions. Dans le cadre du FPGA, la bibliothèque de traduction macro-code /VHDL a été développée et validée au cours de stages DESS et DEA [6-7]. Les programmes VHDL ou IP sont fournis aux étudiants.

Fig. 6. Exemple de macro-code générique.



Le macro-code étant générique, l'utilisateur a la possibilité de porter une nouvelle application sur une autre cible, dès lors que la librairie de traduction est disponible. Cette approche offre ainsi aux concepteurs un niveau d'abstraction suffisant, lui permettant de considérer toute cible matérielle uniquement du point de vue de ses performances et moyens de communications. Aucune re-programmation de fonctions n'est alors nécessaire suite aux différentes implantations.

Vis-à-vis d'autres méthodes, l'approche A<sup>3</sup> ne nécessite aucun codage pendant toute la phase de recherche de solution optimale. L'utilisateur est alors libéré des tâches lourdes de programmation (C/VHDL) et de la mise au point des algorithmes. Ce flot de conception réduit dès lors considérablement le temps de conception-implantation, objectif primordial dans une optique de prototypage rapide et d'évaluation de systèmes.

## 6. Organisation du projet

Afin de mener au mieux ce projet, les étudiants ont besoin d'acquis variés : communications numériques, enseignement sur les processeurs et composants programmables, langage de programmation C et VHDL, systèmes temps réel et méthodologies de conception. L'utilisation de l'outil SynDEx ne nécessite pas de pré-requis. La prise en main et la compréhension des différentes phases sont aussi des objectifs de ce projet.

Ce projet favorise aussi le travail en équipe, chaque groupe pouvant confronter ses résultats. Un travail par binôme ou trinôme semble bien adapté à ce sujet.

Le projet d'une durée de 16 heures peut être décomposé comme suit :

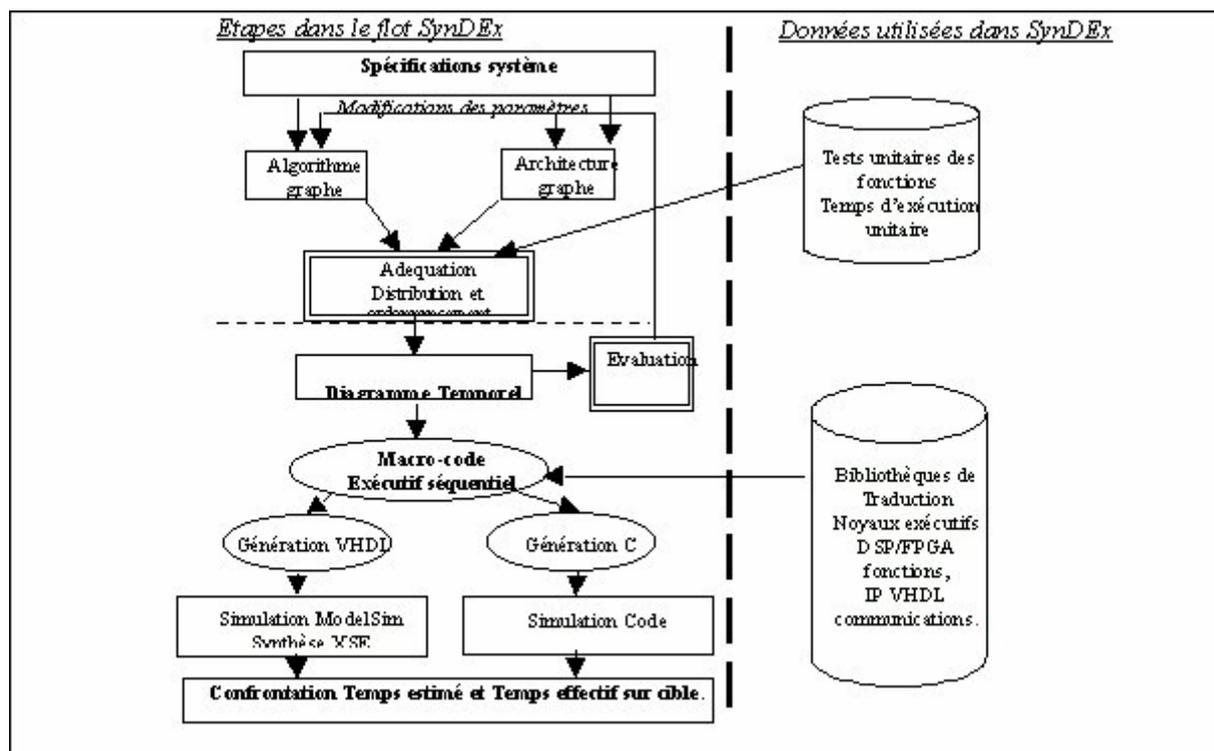
Travail	Résultats attendus	Durée
Présentation générale : système et outils	Compréhension du sujet Applications des méthodes de co-design	2
Prise en main de l'outil ( tutorial ) par chaque groupe	Acquisition du flot de conception	2
Développements de l'application. Étude d'une fonction particulière	Saisie des différents graphes. Adéquation. Analyse des solutions et optimisation. Génération du macro-code. Traduction. Analyse des codes générés	8
Simulation	Simulation des algorithmes / Outils Code Composer et ModelSIM	2
Synthèse *	Portage de la solution retenue sur la carte de prototypage DSP et FPGA	5 *
Présentation des groupes	Confrontation des résultats. Analyse	2
Total projet		16

\* En ce qui concerne l'étape de synthèse, celle-ci peut être proposée comme un complément au projet. L'objectif est de montrer que le code obtenu fonctionne correctement, et que tout nouveau partitionnement n'engendre aucun test supplémentaire. La simulation a pour but de vérifier les mécanismes d'échanges.

## 7. Flot de conception avec SynDEx

La figure 7 reprend le flot de conception d'un système avec SynDEx, complété par les outils de simulation et synthèse. Le choix des outils est justifié par le fait que ceux-ci sont liés aux composants cibles.

Fig. 7. Flot de conception de co-design complet avec SynDEx.



## 8. Raffinement de la modélisation des FPGA dans le graphe d'architecture. Extension possible du projet

Dans le graphe d'architecture, le FPGA est considéré par SynDEx comme un processeur quelconque. L'utilisateur renseigne le modèle en apportant seulement un critère de vitesse de fonctionnement relatif aux autres processeurs. Le FPGA apparaît alors comme un co-processeur, avec un degré de parallélisme unitaire. Il permet d'améliorer les temps d'exécution mais toute sa potentialité n'est pas utilisée, notamment :

- le parallélisme temporel : SynDEx n'exécute qu'une seule fonction à un instant donné dans un FPGA alors que plusieurs fonctions peuvent être exécutées simultanément et implantées selon les ressources disponibles ;
- le parallélisme spatial : une fonction peut être décomposée en opérations de granularité plus fine ; chaque opération peut être implantée plusieurs fois dans le composant et exécutée indépendamment des autres instances.

Afin de lever ces restrictions, une prolongation au projet peut être proposée dans le cadre d'un stage. Le but est de modéliser le FPGA en faisant apparaître son parallélisme effectif. Parmi les solutions, une d'elle consiste à considérer le FPGA comme un ensemble de *sous-fpga*. Chaque *sous-fpga* est introduit dans le graphe d'architecture. Cet ensemble ne correspond en fait qu'à un seul composant FPGA sur la plateforme cible. Dans chacun des *sous-fpga*, une seule fonction pourra être exécutée à un instant donné mais plusieurs *sous-fpga* peuvent exécuter des fonctions en parallèle. Ils peuvent être connectés entre eux et/ou à un processeur logiciel. Ils vont constituer les composants matériels de l'architecture hétérogène, pour lesquels deux types de communications seront nécessaires :

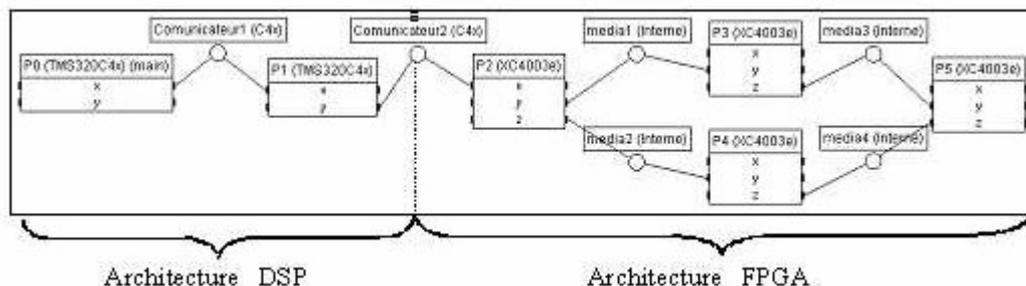
- des communications entre processeurs logiciels et processeurs matériels. Celles-ci correspondent aux ports de communication de type C4X ou SDB dans le cas de notre système ;
- des communications entre processeurs matériels, c'est à dire entre *sous-fpga* ; elles sont internes au composant.

Cette notion de *sous-fpga* est alors utilisée lorsque l'on va rechercher à implanter plusieurs fonctions dans un seul FPGA. La figure 8 illustre cette extension de modèle. La hiérarchie fait alors apparaître de nouveaux modules, qui seront appelés lors de la génération du VHDL à partir du macro-code :

- un module dit « routeur » (P2), réalisant le lien entre les processeurs logiciels et les processeurs matériels (*sous-fpga* P3 à P5). Ce composant comporte notamment l'IP de communication C4X et le lien vers une IP de communication intra-FPGA ;
- des modules (média 1 à 4) pour les communications entre les *sous-fpga*, et donc les fonctions portées sur chaque *sous-fpga*.

Complexe, ce problème, traité dans [7], peut être évoqué afin de mettre en avant les optimisations à apporter à SynDEx.

Fig. 8. Extension du modèle FPGA sous SynDEx.



## 9. Conclusion

Dans cet article, l'utilisation de l'outil SynDEx pour la conception de systèmes portés sur des architectures hétérogènes est présentée. Le déroulement du flot de conception, répondant à la méthodologie A<sup>3</sup>, permet aux étudiants de mettre en pratique les notions de Co-Design, et d'appréhender les problèmes liés à la conception mixte logicielle/matérielle et hétérogène DSP/FPGA. Le sujet étudié est celui d'un système de communication numérique simplifié. Partant de ce système, les binômes peuvent confronter leurs résultats, qui sont validés par l'étape finale de simulation et synthèse du système.

Ce projet peut être étendu à un sujet plus vaste de stage, dont le but est d'enrichir les modèles d'architectures et les bibliothèques associées.

## Références bibliographiques

- [1] E. Buracchini, *The Software Radio Concept*, IEEE Communications Magazine, **38(9)** (2000) 138-143.
- [2] B. Seljak, *Hardware-software co-design for a real time executive*, Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE'99, Bled, Slovenia, 1999, Vol. 1, 55-58.
- [3] T. Grandpierre et Y. Sorel, *Optimised rapid prototyping for real time embedded heterogeneous multiprocessors*, actes du congrès 7<sup>th</sup> International Workshop on Hardware/software Co-Design, Rome, Italy, 1999, 74-78.
- [4] Site internet *Sundance* : [www.sundance.fr](http://www.sundance.fr) (consulté le 5 avril 2003).
- [5] S. Le Nours, F. Nouvel et J.F. Héland, *Efficient Implementation of a MC-CDMA Transmission System for the Downlink*, IEEE 54<sup>th</sup> Vehicular Technology Conference, VTC2001, Atlantic City, USA, octobre 2001.
- [6] Y. Le Mener, M. Raulet, J.F. Nezan, A. Kountouris et C. Moy, *SynDEx executive kernel development for DSPs TI C6x applied to real-time and embedded multi-processors architectures*, European signal processing conference, EUSIPCO 2002, Toulouse, France, septembre 2002.
- [7] I. Hermann, *Utilisation de SynDEx pour les architectures logicielles et matérielles*, rapport de stage DESS Composants Electroniques, Université de Rennes 1 (2002).