Développement d'un noyau d'exécutif SynDEx pour DSP TI C6x appliqué aux architectures multiprocesseurs temps-réel et embarquées

Mickaël RAULET,
MITSUBISHI ELECTRIC ITE
Laboratooire de Télécommunications
80, av des Buttes de Coësmes, 35700 RENNES, France
Tél: +33/0 299 841 129 Fax: +33/0 299 842 115
Mél : raulet@tcl.ite.mee.com

Jean-François NEZAN, Olivier DEFORGES
Laboratoire IETR UMR 6164
INSA Rennes
20, av des Buttes de Coësmes, CS 14315 35043 RENNES Cedex, France
Tél: +33/0 223 238 459 Fax: +33/0 223 238 262
Mél : jnezan@insa-rennes.fr

Résumé

Les applications temps-réel de traitement du signal et d'images ont des contraintes de temps très importantes, nécessitant l'utilisation de plusieurs unités de calcul puissantes. Le but de nos travaux est de développer un processus de prototypage rapide dédié aux architectures parallèles constituées de plusieurs processeurs de traitement numérique du signal de dernière génération. Nous utilisons pour cela SynDEx, l'outil basé sur la méthodologie AAA[1], pour améliorer l'implantation de l'algorithme sur une architecture multiprocesseurs. Nous avons donc développé un noyau d'exécutif SynDEx pour la famille de DSP C6x, afin de générer automatiquement un exécutif statique, distribué et optimisé de l'algorithme spécifié sur ces processeurs.

1 Introduction

De plus en plus de champs d'applications (téléphone, radar, audio, multimédia ou encore radio logicielle) utilisent les technologies dédiées au traitement numérique du signal. Elles sont de plus en plus complexes, améliorant de ce fait les interactions avec l'utilisateur ou manipulant plusieurs objets multimédia.

Les performances requises doivent souvent être améliorées pour une utilisation temps-réel, à cause de la haute complexité des algorithmes développés. Les DSP[2] sont des processeurs programmables plus particulièrement dédiés au traitement numérique du signal. De tels processeurs traitent généralement des données en temps-réel et sont architecturalement optimisé pour accélérer des calculs numériques intensifs et répétitifs à consommation réduite.

Mais dans de nombreux cas, les solutions logicielles basées sur des architectures monoprocesseur ne sont pas assez efficaces. Des composants dédiés (ASIC, FPGA) permettent d'accélérer des calculs spécifiques, comme l'estimation de mouvement dans le codage d'image ou les turbo codes dans les communications numériques. Ces composants couplés à des processeurs programmables aboutissent à des plates-formes hétérogènes, ce qui requiert souvent de nombreux spécialistes pour leur mise au point. Généralement de telles plates-formes sont conçues pour une application unique.

Une autre solution est l'utilisation de plusieurs processeurs pour une application [**?**, **?**]. De meilleurs temps d'exécution peuvent être atteints, tout en gardant la flexibilité de la plate-forme. Mais là encore ce n'est pas simple : beaucoup d'opérations sont à faire à la main. Notre but est d'effectuer une implantation automatique de traitement du signal ou d'image sur une architecture multi-C6x. Nous introduisons pour cela son noyau d'exécutif SynDEx.

Ce papier est organisé de la façon suivante : dans la section 2, nous décrivons le logiciel SynDEx. Dans la section 3, les spécificités du C6x sont étudiées. Le processus de développement du noyau d'exécutif est expliqué dans la section 4. Finalement, nous donnerons nos perspectives et conclusions dans la section 5.

2 SynDEx

SynDEx[**?**, **?**] est un outil d'accès libre et universitaire, dont la signification est "Synchronized Distributed Execution". Le logiciel SynDEx s'inscrit dans le cadre des recherches menées dans le projet SOSSO de l'INRIA Rocquencourt, plus précisément sur la méthodologie AAA[**?**] pour les systèmes informatiques temps-réel distribués embarqués hétérogènes.

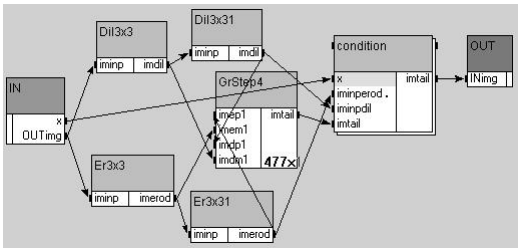

FIG. 1 – Graphe d'algorithme de SynDEx V6.

SynDEx nécessite la description de deux graphes. D'une part, un graphe d'algorithme (fig.**??**) permet de spécifier l'application à prototyper, en faisant ressortir son parallélisme potentiel. D'autre part, un graphe

[1] Adéquation Algorithme Architecture
[2] Digital Signal Processor

d'architecture qui implique les organes de calcul et les média de communications les reliant (fig.**??**), définit le parallélisme disponible de l'architecture sur laquelle on veut implanter l'application. Il faut ensuite définir les temps d'exécution des opérations de l'application pour chaque élément de l'architecture susceptible de l'exécuter, ainsi que la durée des communications inter-processeur.

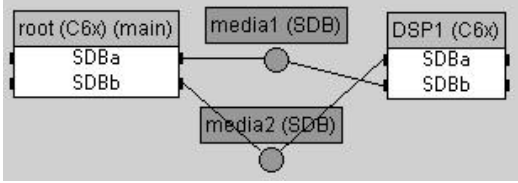

FIG. 2 – Graphe d'architecture de SynDEx V6.

L'implantation optimisée, appelée adéquation, forme un graphe obtenu par transformation des deux graphes précédents. Parmi toutes les transformations possibles, l'heuristique d'optimisation fondée sur la prédiction de performances, conserve celle qui minimise la durée d'exécution de l'algorithme (latence).

La visualisation du résultat s'effectue grâce au graphe temporel fourni par SynDex : le "time schedule" (fig.**??**). Cette prédiction permet de visualiser le parallélisme obtenu et, si nécessaire, d'optimiser l'implantation. Pour cela, l'utilisateur peut interagir avec l'heuristique pour l'aider à trouver de meilleurs résultats, en adaptant la taille des grains de l'algorithme, et en modifiant les ressources de l'architecture.

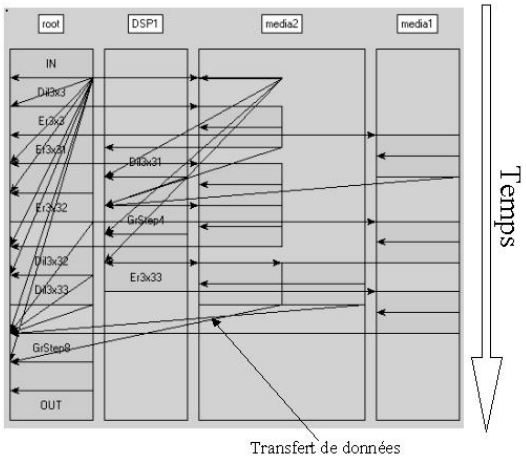

FIG. 3 – Adéquation SynDEx V6.

Après la construction du graphe d'implantation, SynDEx génère un macro code indépendant des langages utilisés par les processeurs (fig.**??**). Le processus qui conduit à l'objectif final de la méthodologie AAA (l'exécution de l'algorithme sur l'architecture réelle) inclut tout d'abord la transformation de chaque macro code, constituant les programmes de chaque processeur, en un exécutif compilable. Il est nécessaire pour cela de disposer de bibliothèques de traduction fourni avec SynDEx.

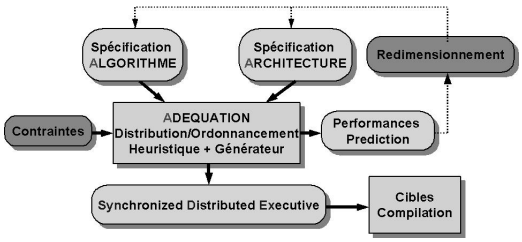

FIG. 4 – Utilisation de SynDEx

Le principal avantage de SynDEx est d'éviter les OS[3] comme 3L (produit de "Diamond") afin d'implementer un exécutif statique, taillé sur mesure pour une application donnée.

SynDEx est un outil en constante évolution. SynDEx version 6 offre de nouvelles fonctionnalités comme la possibilité de définir de la hiérarchie, du conditionnement et des répétitions dans le graphe d'algorithme.

3 DSP "Texas Instrument" TMS320C6x

3.1 Propriétés du C6x

Les C6x (fréquence de 150 MHz à 600 MHz) sont dotés du VelociTI, une architecture VLIW [4] hautement performante et très avancée, capable d'exécuter huit instructions de 32 bits sur huit unités de calcul par cycle d'horloge.

Plusieurs périphériques, très intéressants pour le temps-réel et les systèmes embarqués sont disponibles sur le C6x : DMA, plusieurs types de mémoire (interne, externe), des ports séries ou des timers.

Code composer Studio[**?**] est un outil de développement incluant un compilateur du langage C vers l'assembleur très performant, ce qui est de première importance pour les architectures VLIW. Il semble suffisamment optimisé pour rendre toute tentative de programmation directe en assembleur inutile (du point de vue de la performance) et coûteuse (du point de vue du temps de développement).

Les C4x, ancienne génération de DSP "Texas Instrument", intègrent des ports de communications (CP), fournissant des communications inter-C4x. Cette interface spécifique aux C4x permet de construire des architectures parallèles rapidement. A l'inverse, les DSP C6x ne possèdent pas de CP, c'est pourquoi les fabricants doivent ajouter des ressources matérielles entre les C6x afin de rendre possible les communications (fig.**??**), devenant de ce fait dépendantes de l'architecture. De part leurs architectures internes très différentes, l'assembleur spécifique au noyau d'exécutif C4x n'est plus utilisable sur le C6x.

[3] operating system
[4] very-long-instruction-word

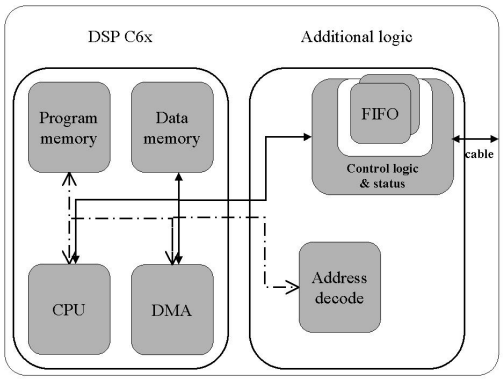

FIG. 5 – C6x Logique additionnelle pour les communications

La logique additionnelle généralement adoptée par les fabricants est constiutée d'une FIFO[5] qui communique via une autre FIFO connectée à un DSP. La taille, les informations de contrôle et d'état de la FIFO peuvent changer suivant la plate-forme. Une FIFO peut interrompre ou non une séquence de calcul sur le DSP par exemple à la fin d'un transfert. Le nombre et la manière de générer ces interruptions peuvent changer et ces spécifications sont effectuées par les fabricants.

3.2 Développement de la méthode

L'exécutif généré par SynDEx est divisé en plusieurs fichiers sources, chacun d'eux contenant un code intermédiaire composé d'une liste d'appels de macros du noyau générique. Ces appels de macros sont traduits par le macro processeur M4[**?**] en un fichier source compilable pour le processeur cible. Des librairies M4 ont donc été créées formant le noyau C6x pour SynDEx. Ce travail n'est habituellement pas à faire par un utilisateur de SynDEx. Notre rôle consiste justement ici à enrichir les bibliothèques de SynDEx qui seront fournies aux utilisateurs futurs.

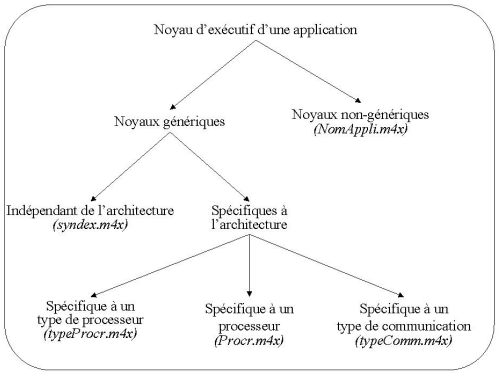

FIG. 6 – Organisation du noyau d'exécutif

Notre noyau a été développé en langage C afin d'être partiellement et facilement réutisable pour d'autres processeurs programmables en C. Le débogage est simplifié, sans perte de temps sur une application complète.

Le noyau d'exécutif C6x a été divisé en plusieurs librairies (fig.**??**), facilitant son adaptation à une autre architecture. La librairie non-générique spécifie les macros M4 pour l'application, tel que les fonctions d'appel et les entrées sorties de notre système. La bibliothèque indépendante de l'architecture contient les définitions qui permettent de faire l'interface entre les macros génériques et les macros des noyaux spécifiques à l'architecture et à l'application. Les autres librairies sont dépendantes de l'architecture : type de processeur, de média de communication, langage du processeur utilisé. Cette organisation permet de n'avoir que quelques macros à redéfinir pour de nouvelles plates-formes.

4 Noyau développé

4.1 Communications inter-C6x développées

Le macro code SynDEx de chaque processeur crée plusieurs fonctions entrelacées permettant du parallélisme entre la séquence de calcul ("main") et les séquences de communications (ensemble de fonctions de transfert appelé "thread"). L'ordonnancement de SynDEx est statique et géré par des sémaphores de contrôle. L'optimisation du parallélisme et du temps se fait grâce à l'usage du DMA multi-canal du C6x, chaque canal étant relié à un média de communication. Le C6x peut donc avoir sur le graphe d'architecture quatre média de communication donc quatre connexions.

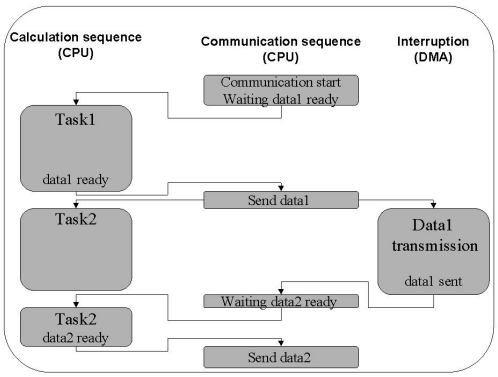

FIG. 7 – parallélisme entre calcul et communication

Les fonctions de transferts de chaque média (envoi ou réception sur la FIFO) sont structurées de la façon suivante :

- Calcul du nombre de paquets à transmettre (partitionnement de la donnée en paquet à cause de la taille de la FIFO)
- Sélection de la FIFO
- Configuration de la FIFO
- Autorisation des interruptions
- Démarrage du DMA

[5] first in first out

4.2 Sundance

Le noyau C6x développé a été tout d'abord testé sur une carte mère "Sundance" SMT320 constituée de deux modules SMT335. Chaque module SMT335 contient un DSP TMS320C6201, dont la fréquence d'horloge est de 200 MHz, et un FPGA[6]. Ce FPGA s'occupe de gérer les communications (FIFO bidirectionnelles) : six ports de communication de 20 Moctets/sec (CP : type de communication standard du C4x) et deux bus propriétaires de 200 Moctets/sec (SDB : Sundance digital bus).

Pour développer notre noyau sur cette plate-forme, nous nous sommes inspirés de librairies existantes puis nous avons créé quatre bibliothèques M4 : une dédiée aux macros spécifiques aux C6x, une pour chaque média de communications (CP.m4x et SDB.m4x) et une autre pour la plate-forme (SMT335.m4x). La dernière est indispensable pour l'initialisation des paramètres de notre carte. Notons que les librairies spécifiques aux média sont très proches les une des autres et de très bons exemples pour de futurs développements.

4.3 Pentek

La plate-forme "Pentek"[**?**] dispose de quatre DSP TMS320C6201. Chaque C6x possède trois liens de communications : deux FIFO bidirectionnelles entre les DSP et une FIFO d'entrée sortie connectée à des convertisseurs analogiques-numériques(CAN, CNA).

Cette plate-forme confirme les différences entre les architectures, c'est pourquoi nous avons donc développé deux autres librairies dépendantes du nouveau média de communication : BIFO.m4x (sans DMA) et BIFODMA.m4x (avec DMA). La version sans DMA est une version utile facilitant le débogage. Similairement à SMT335.m4x, nous avons réalisé Pentek.m4x pour la plate-forme. L'adaptation a cette plate-forme s'est avéré très rapide, validant nos choix suffisamment génériques.

4.4 Applications

La génération d'exécutif a été testée sur diverses applications : deux en traitements d'images (sur la plate-forme"Sundance") et une en traitement numérique du signal ("Pentek"). Le temps d'implantation d'une application est négligeable par rapport à sa description en graphe flot de données et sa vérification fonctionnelle. Le placement, le partitionnement et la génération d'exécutif sont effectués automatiquement.

5 Conclusions et perspectives

Nous avons développé un générateur d'exécutif distribué et automatique pour des architectures multi-C6x pour SynDEx. Plus particulièrement, nous l'avons testé sur des C6201, compatible avec les autres DSP de la famille C6x. De plus l'utilisation du langage C peut servir de base à la génération d'autres noyaux. Les exécutifs générés sont taillés sur mesure évitant d'incorporer des OS et sauvegardant l'intégrité de nos ressources matérielles.

Au cours de nos travaux, nous avons pu implanter un certain nombre d'applications complexes telles q'un décodeur Mpeg-4[**?**] et un récepteur FM numérique[**?**].

Nous collaborons étroitement avec l'INRIA sur SynDEx version 6 pour la réalisation automatique de code des nouvelles fonctionnalités : mémoire partagée, description hiérarchique et conditionnelle du graphe d'algorithme. Ces fonctionnalités doivent permettre la description et l'implantation d'application avec une reconfiguration totale ou partielle. Des points restent à éclaircir dans l'heuristique de SynDEx comme par exemple : l'optimisation de l'implantation du conditionnement, la détermination de l'architecture optimale conforme à l'algorithme, l'augmentation de la fréquence de calcul d'un algorithme (Pipelinisation).

Notre travail de recherche s'oriente également vers une architecture hétérogène incluant en plus FGPA et processeur de type GPP[7], pour lesquels des bibliothèques spécifiques, doivent être développées.

Références

[1] C. Lavarenne, Y. Sorel, *"Specification, performance optimization and executive generation for real-time embedded multiprocessor application with Syndex"*, Proc. of Real Time Embedded Processing for Space Applications, CNES Inter-national Symposium.

[2] T.Grandpierre, C.Lavarenne, Y.Sorel, *"Optimized rapid prototyping for real time embedded heterogeneous multiprocessors"*, In 7th International workshop on Hardware/Software Co-Design, pages 74-78, Rome, Italy, May 3-5 1999. IEEE Computer Society, ACM SIGSOFT, IFIP.

[3] M.Cosnard, A.Ferreira, *"The real power of loosely coupled parallel architectures"*, In parallel processing letters, pages 103-112, 1991.

[4] S.Srikanteswara, J.H.Reed, P.Athanas, R.Boyle, *"A soft radio architecture for reconfigurable platforms"*, In IEEE Communications Magazine, Feb 00.

[5] M.Loukides, A.ORAM, *"Programming with GNU software"*, O'Reilly, 1996.

[6] Texas Instruments technical documents, *"TMS320C6000 Code Composer Studio - tutorial"*, Ref spru301c. Available at http ://www.dspvillage.ti.com. Feb 00.

[7] J.F.Nezan, V.Fresse, O.Deforges *"Fast prototyping of paral-lel architectures : an Mpeg-2 coding application"*, In CISST'2001 proceedings, Las Vegas, USA, June 01.

[8] A.Kountouris, C.Moy, L.Rambaud, P.Lecorre *"Reconfigurable radio case study : A software based multistandard transceiver for UMTS, GSM, EDGE and Bluetooth"*, In IEEE VTC (Vehicular Technology Conference) Fall'2001, Atlantic City, USA, Oct 01.

[6] field programmable gate array

[7] General Purpose Processor