Placement de données en mémoire sans conflit pour l'optimisation du réseau d'interconnexion et du contrôleur des entrelaceurs parallèles

BRIKI AROUA, CHAVET CYRILLE, COUSSY PHILIPPE, MARTIN ERIC

Lab-STICC / Université de Bretagne Sud, CNRS UMR 6285.
Centre de Recherche Christiaan Huygens, Rue St Maude 56100 Lorient

{prénom.nom@univ-ubs.fr}

Thème – Adéquation algorithmes et architectures (T5.1)

Problème traité – Génération automatique d'architectures d'entrelacement et placement des données en mémoire garantissant des accès mémoires parallèles sans conflits avec une forte réduction du coût de l'architecture (contrôleur, réseau d'interconnexion…).

Originalité – Nous proposons (1) un modèle formel matricielles permettant de représenter les séquences d'accès aux données, (2) un modèle formel basé sur des graphes de conflits pour représenter les relations entre les accès aux données et (3) des algorithmes d'exploration de la matrice et du graphe pour réaliser l'assignation des données sur des structures de mémorisation en optimisant le coût architectural du système résultant. L'approche proposée permet une exploration architecturale rapide et performante de l'espace de conception.

Résultats – A partir d'une description de la règle d'entrelacement (issue de standards de communication) et de contraintes de débit et de parallélisme, une architecture optimisée (décrite en VHDL RTL synthétisable) et un placement des données en mémoire sont générés. Les synthèses réalisées montrent de fortes réductions de la surface des architectures.

1 Problématique

Les applications du traitement du signal (TDSI) sont maintenant largement utilisées dans des domaines variés allant de l'automobile aux communications sans fils, en passant par les applications multimédias et les télécommunications. La complexité croissante des algorithmes implémentés et l'augmentation continue des volumes de données et des débits applicatifs requièrent souvent la conception de circuits intégrés dédiés (ASIC). Typiquement l'architecture d'un composant complexe du TDSI utilise (1) des éléments de calculs de plus en plus complexes, (2) des mémoires et des modules de brassage de données (entrelaceur/désentrelaceur pour les TurboCodes, blocs de redondance spatio-temporelle dans les systèmes OFDM [1]/MIMO, …). Aujourd'hui, la complexité et le coût de ces systèmes sont très élevés; les concepteurs doivent pourtant parvenir à minimiser la consommation et la surface total du circuit, tout en garantissant les performances temporelles requises. Sur cette problématique globale, nous nous intéressons à l'optimisation des architectures des modules de brassage de données (réseau d'interconnexion, contrôleur…) devant réaliser une règle d'entrelacement définie par l'application et ayant pour objectif d'utiliser de réseau d'interconnexion défini par le concepteur.

Les solutions existantes dans l'état de l'art ne répondent que partiellement à ces objectifs. Ces approches se répartissent en trois grandes familles :les approches telles que proposées dans [GNA04] permettent de résoudre les conflits d'accès mémoire à un coût architectural limité, mais impose au concepteur de définir sa propre règle d'entrelacement; d'autres travaux tels que [WHE04][MUL06], proposent de résoudre les conflits mémoire à l'exécution du système, mais au prix d'un surcoût architectural important et d'un dégradation des performances temporelles de l'application ; enfin des travaux visent à résoudre les conflits mémoires hors ligne, mais ces solutions ne peuvent pas cibler un réseau d'interconnexion défini par le concepteur [TAR04][SAN11] ni même optimiser le coût architectural du système [TAR04][SAN11][CHA10]. L'approche que nous proposons, se classe dans cette dernière famille (génération hors ligne du placement mémoire).

2 Approche proposée

Nous proposons une méthodologie d'exploration et de conception permettant de générer automatiquement une architecture d'entrelacement optimisée réalisant une règle de brassage de données, ou entrelacement, tel que définie par exemple dans un standard de communication. Notre flot de conception prend en entrée (1) des diagrammes temporels (générés à partir de la règle d'entrelacement et de contraintes spécifiant les séquences d'accès parallèles

[1]OFDM : technique de modulation se basant sur le multiplexage fréquentiel de signaux.

aux données) et (2) une contrainte utilisateur sur le réseau d'interconnexion que doit utiliser l'architecture. Ce flot formalise ensuite ces contraintes de brassage des données sous la forme (1) d'un modèle matriciel des séquences de données qui devront être traitées par chaque processeur (cf. Figure 1) et (2) d'un Graphe de Conflit d'Adressage (GCA), dont les propriétés permettent une exploration efficace de l'espace des solutions architecturales. L'objectif est ensuite de générer une architecture cible, en garantissant un fonctionnement sans conflit d'accès mémoire (lorsque plusieurs processeurs veulent accéder en même temps à un même banc mémoire mais pour traiter des données différentes), en respectant la contrainte de réseau et en optimisant l'architecture obtenue (notamment concernant l'architecture de son contrôleur).

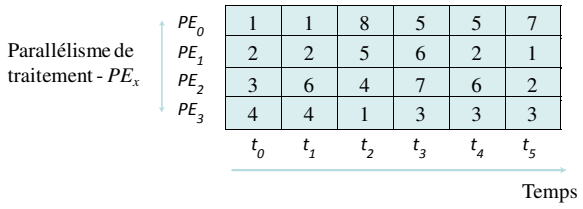

Figure 1 : Ordre d'accès aux données

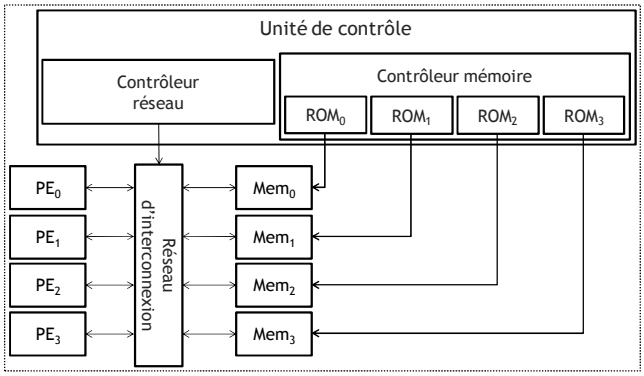

Figure 2 : Architecture cible

L'architecture que nous ciblons (cf. Figure 2) se compose d'éléments de calculs (PE_0,…PE_n), de mémoires de données utilisées pour stocker les opérandes et les résultats produits par les éléments de calculs (Mem_0,…Mem_m), d'un réseau d'interconnexion reliant les éléments de calculs aux mémoires et d'une unité de contrôle. Le réseau d'interconnexion est défini par l'utilisateur et peut être basé sur différent modèles : cross-bar, réseaux de Benes [BEN65], réseau de Bruinj [BRU46], barrière de multiplexeurs, barrel-shifters (barillets), papillons... L'unité de contrôle est composé de deux parties : un contrôleur de réseau et un contrôleur de mémoires. Ces contrôleurs sont basés sur un ensemble de mémoires de contrôle (une ROM de contrôle par banc mémoire *Mem* dans l'architecture cible) contenant les mots de commande relatifs au fonctionnement du système. Plus le système devra mettre en œuvre de règles d'entrelacement (e.g. différentes longueurs de trames pour un même standard, différents standards…), plus le coût de ce contrôleur sera élevé car il faudra un ensemble de ROMs distinct pour chacun des modes de fonctionnement (cf. [SAN12]). L'approche que nous proposons est à même d'optimiser cette partie de contrôle de l'architecture.

La Figure 3 présente notre flot de conception qui se compose de deux étapes :

- Génération d'un placement des données en mémoire garantissant un fonctionnement du système sans conflit d'accès à ces mémoires et en respectant la contrainte de réseau définie par le concepteur. Pour ce faire, les accès aux données (cf. Figure 1) sont formalisés via une matrice associant à chaque donnée, deux bancs mémoire (accès en lecture et en écriture à la donnée, cf. Figure 4). Une étape dite « relaxation de contraintes » permet d'assigner temporairement des données conflictuelles dans des registres additionnels, en lieu et place d'un banc mémoire (cf. Figure 4, solution à 4 bancs mémoire -*A,B,C,D*- et un registre *r*). En plus de ce placement des données en mémoire, cette première étape génère un graphe GCA modélisant les conflits d'adressage potentiels pour chacun des bancs mémoire de l'architecture. Dans un tel graphe, les nœuds représentent les différents accès aux données (lecture ou écriture) qui doivent être réalisés ; les arcs représentent les incompatibilités entre ces accès (i.e. si deux nœuds sont reliés, cela signifie qu'il faudra les stocker dans deux adresses mémoires distinctes dans le banc mémoire considéré). L'assignation des données aux bancs mémoires est réalisée via une heuristique dédiée qui tout en tenant compte des conflits mémoire et des contraintes utilisateur, vise à réduire le nombre de conflits d'adresse potentiels dans le GCA afin minimiser le coût du contrôleur d'adresse dans l'architecture. Cette première étape génère ainsi un placement en mémoire sans conflit des données et un graphe GCA.
- Génération de l'adressage des données dans chaque banc mémoire. Il s'agit d'affecter aux données leur adresse de stockage, au sein du banc mémoire qui leur a été assigné lors de la première étape. Cette affectation se fait via l'exploration des GCAs avec pour objectif de minimiser le coût du contrôleur mémoire, en favorisant au maximum les ressemblances entre les séquences de contrôle qui seront stockées dans chacune des ROMs de l'architecture (cf. Figure 2). Ainsi, si par exemple plusieurs ROMs contiennent les mêmes séquences de contrôle, une seule pourra être utilisée dans l'architecture finale. Pour ce faire, une heuristique dédiée à été définie pour affecter à une donnée l'adresse dans son banc mémoire qui respectera au mieux l'objectif d'optimisation du contrôleur mémoire global (i.e. en relation avec les GCAs des autres bancs mémoires et les données dont l'adressage dans leur banc a déjà été défini).

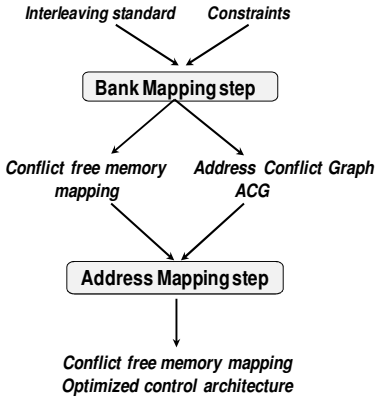

Figure 3 : Flot d'exploration proposé

1		1		8		5		5		7	
B	A	A	D	C	C	A	D	D	A	C	D
2		2		5		6		2		1	
C	C	C	A	A	A	C	B	A	D	B	B
3		6		4		7		6		2	
A	B	B	C	B	D	D	C	B	B	D	C
4		4		1		3		3		3	
D	D	D	B	D	B	B	r	r	r	r	A

Figure 4 : Matrice de placement mémoire

3 Résultats expérimentaux

Cette approche a été mise en œuvre au sein d'un d'outil et appliquée sur plusieurs cas d'étude. Le VHDL généré a été synthétisé via Xilinx ISE Design Suite 2012, sur plateforme Virtex6. Les cas d'études utilisé couvrent des standards de codes correcteurs d'erreurs de type Turbo-Codes et LDPC : High Speed Packet Access (HSPA), Ultra-WideBand (UWB) et une application Wimax. Ces expériences ont explorées différents parallélismes et radix. Nos résultats sont comparés à différentes approches comparables de l'état de l'art [TAR04] et [CHA10] (cette dernière approche ne peut pas s'appliquer dans le cas des LDPC). Puisque que l'approche que nous proposons peut cibler plusieurs type de réseau d'interconnexion, nous utiliserons deux cibles : un réseau de type papillons (HPSA, WiMAX) et un barrel-shifter (UWB). Les résultats sont présentés dans les tableaux suivant.

Tab 1 : Résulats comparatifs à [TAR04]

Test Case	Frame length	Parallelism	Area		Gain
			[TAR04]	SAGE	
HSPA 1024 Radix 2	*1024*	*4*	967	625	**35,37%**
HSPA 1024 Radix 2	*1024*	*8*	1309	1051	**19,71%**
HSPA 1024 Radix 4	*1024*	*8*	1320	1057	**19,92%**
HSPA 5120 Radix 2	*5120*	*8*	6804	3886	**42,89%**
HSPA 5120 Radix 2	*5120*	*16*	9735	7210	**25,94%**
HSPA 5120 Radix 4	*5120*	*16*	9668	9312	**3,68%**

Tab 2 : Résultats comparatifs à [CHA10]

Test Case	Frame length	Parallelism	Area		Gain
			[CHA10]	SAGE	
UWB	*1200*	*8*	2215	1544	**30,29%**
UWB	*1200*	*4*	1519	1038	**31,67%**
UWB	*600*	*8*	1346	981	**27,12%**
UWB	*600*	*4*	994	686	**30,99%**
UWB	*300*	*8*	949	762	**19,70%**
UWB	*300*	*4*	593	394	**33,56%**
WiMAX 2by3	*80*	*10*	553	532	**3,80%**

Les résultats de synthèse que nous présentons sont exprimés en « slices » et portent sur les coûts comparés du contrôleur du système (le reste de l'architecture est commun, excepté pour [TAR04] dont l'incapacité à respecter un réseau d'interconnexion ciblé amène un surcoût important mais les résultats présents en auraient de ce fait été faussés). En moyenne, notre approche permet de réduire de 24,97% à 42% la surface du contrôleur selon les applications. Ces résultats démontrent la pertinence de l'approche que nous proposons. Nos travaux actuels concernent désormais la génération d'architectures mutli-mode, c'est-à-dire pouvant respecter plusieurs standards de communications différents, ou bien plusieurs longueurs de trame différentes d'un même standard. De plus, des travaux sont menés afin de définir une approche capable, le cas échéant, de s'affranchir de la contrainte de réseau définie par l'utilisateur. L'idée est que le flot de conception soit à même d'analyser la règle d'entrelacement qui lui est fournie, pour ensuite proposer au concepteur un réseau d'interconnexion had-hoc en fonction de cette règle d'entrelacement.

4 Références

[GNA04] D. Gnaedig, E. Boutillon, M. Jezequel, V.C. Gaudet, P.G. Gulak, "On multiple slice turbo codes", in proc.3rd *Int. Symp. TurboCodes, pp. 343-346,* Brest, 2003.

[WHE04] N. When, "SOC-Network for Interleaving in wireless Communications", *MPSOC*, 2004.

[TAR04] A. Tarable, S. Benedetto, G. Montorsi, "Mapping interleaving laws to parallel turbo and LDPC decoder architectures", *IEEE Trans. Inf. Theory, vol.50, no.9, pp.2002-2009*, Paris, 2004.

[MUL06] O. Muller, A. Baghdadi, M. Jezequel, "ASIP-based multiprocessor SoC design for simple and double binary turbo decoding", *Design, Automation, and Test in Europe (DATE)*, 2006

[SAN11] A.H. Sani, P. Coussy, C. Chavet, E. Martin, "An Approach Based on Edge Coloring of Tripartite Graph for Designing Parallel LDPC Interleaver Architecture", *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, Rio de Janeiro, Brazil, 2011.

[CHA10] C. Chavet, P. Coussy, E. Martin, P. Urard, "Static Address Generation Easing: a Design Methodology for Parallel Interleaver Architectures", *proc of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1594-1597,* Dallas, 2010

[BRI12] A. Briki, C. Chavet, P. Coussy, E. Martin, "A Design Approach Dedicated to Network-Based and Conflict-Free Parallel Interleavers", *proc of Great Lake Symposium on VLSI (GLS-VLSI)*, Salt Lake City, 2012

[BEN65] V.E. Benes, "Mathematical Theory of connecting network and telephone trafic", New York, N.Y.: Academic, 1965.

[BRU46] N.G de Bruijn, « A Combinatorial Problem », Koninklijke Nederlandse Akademie v. Wetenschappen, vol. 49, 1946, p. 758–764

[SAN12] O. Sanchez, S. ur Rehman, A. Sani, C. Jego, C. Chavet, P. Coussy, M. Jezequel, "A dedicated approach to explore design space for hardware architecture of turbo decoders", I*n Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS),* Quebec, 2012