Optimisation de la consommation des unités de mémorisation lors de la synthèse d'architecture

Gwenolé CORRE, Nathalie JULIEN, Eric SENN, Eric MARTIN

LESTER[1], Université de Bretagne Sud
Centre de recherches, BP 92116
56321 Lorient Cedex – France
Tel : 02/97/87/45/29 Fax : 02/97/87/45/00
e-mail : prenom.nom@univ-ubs.fr

RÉSUMÉ : Les systèmes supportant des applications de traitement du signal et de l'image manipulent de plus en plus de données. La multiplication du nombre de données entraîne une utilisation intensive de la mémoire qui devient le point critique du système ; la mémoire limite les performances et représente une proportion importante de la consommation globale. Dans le contexte de la synthèse d'architecture, nous nous intéressons au développement d'une stratégie de conception et d'optimisation de la consommation des unités de mémorisation. Le travail est dans sa première étape ; il s'agit de définir et d'évaluer les différentes phases permettant la mise en place d'un flot de conception des unités de mémorisation, orienté vers la réduction de la consommation.

MOTS-CLÉS : Synthèse d'architecture, synthèse mémoire, optimisation de la consommation.

1. Introduction

La conception de systèmes électroniques soulève de nombreuses problématiques : respect des contraintes de fiabilité, de sûreté de fonctionnement, de temps de conception, de surface et de consommation, garantie des performances. Les applications deviennent de plus en plus complexes et manipulent un nombre de données toujours plus important, notamment en traitement du signal et de l'image. La consommation et les temps d'accès des unités de mémorisation dominent largement la consommation totale et les performances globales de telles applications. La mémoire apparaît alors comme le point critique ; elle représente de 50 à 80% de la consommation globale dont la moitié dissipée par le décodage d'adresse et les buffers d'adresses pour les mémoires modernes [1]. Pour concevoir un système répondant à des contraintes fortes, il faut donc mettre en œuvre des techniques d'optimisation permettant de réduire la consommation de la mémoire, tout en garantissant les contraintes temporelles et spatiales du système.

La problématique de l'optimisation de la consommation d'un système se pose à tous les niveaux de son flot de conception : niveau physique,

logique, RTL, algorithmique. Plus le facteur de consommation est pris en compte à un niveau d'abstraction élevé, et plus les transformations opérées à ce niveau auront un impact important sur la consommation. La prise en compte des problèmes d'optimisation de la consommation au niveau de la synthèse d'architecture permet donc de réduire de manière significative la consommation globale d'un système ; pour cela le concepteur pourra agir sur les paramètres algorithmiques, architecturaux et technologiques. Pour répondre au problème d'optimisation de la consommation, il faut identifier tous les paramètres mis en jeu et définir les phases du flot de conception durant lesquelles ils auront le plus d'influence. L'efficacité de la stratégie de conception dépend fortement de ce travail préliminaire.

Dans cet article, nous présentons tout d'abord le contexte de l'étude en section 2, avec la définition de consommation des mémoires et un état de l'art sur les techniques d'optimisation. Puis, dans la section 3, nous explicitons une première démarche de la méthodologie que nous allons mettre en œuvre pour obtenir une unité de mémorisation optimisée avant de conclure brièvement sur les travaux futurs.

---

[1] Laboratoire d'Electronique et des Systèmes TEmps Réel ; http//lester.univ-ubs.fr :8080

2. Contexte

a. Consommation des mémoires

Vu le nombre important de transistors intégrés dans une mémoire, la consommation statique n'est pas négligeable devant la consommation dynamique, contrairement aux autres ressources. La consommation statique est proportionnelle à la taille, la consommation dynamique correspond à la consommation moyenne par accès.

$$Pmem = Pstat + Pdyn \quad (1)$$

$$Pdyn = \alpha F C Vdd^2 \quad (2)$$

Dans l'expression (1), la puissance totale de la mémoire, Pmem, est la somme de la puissance statique Pstat et de la puissance dynamique Pdyn. La puissance statique est due principalement au courant de fuite des transistors. Le nombre de transistor augmentant avec la taille mémoire, Pstat peut être définie par une fonction dépendant de la taille mémoire. La puissance dynamique, Pdyn, dépend du taux d'activité du circuit α, de la fréquence d'accès F, de la capacité de charge effective C et de la tension d'alimentation Vdd.

Les paramètres ayant le plus d'influence sur la consommation sont les transferts de et vers la mémoire, le nombre de données "vivantes" à un instant et le nombre de transitions sur les bus d'adresses. Le nombre de transferts vers la mémoire influence directement le nombre de transitions entre l'unité de mémorisation et l'unité de traitement. Ce nombre détermine la bande passante et le nombre de ports. Il affecte grandement la surface et le nombre de charges capacitives à faire commuter. Une façon efficace de réduire les accès mémoire est d'améliorer les localités spatiales et temporelles. Le nombre de données "vivantes" devant être stockées simultanément et le nombre total de données déterminent le nombre et la taille des mémoires. Ils ont une influence directe sur la consommation puisque le nombre de transitions (charge capacitive) et la puissance statique augmentent avec la taille. La réduction de la consommation peut être obtenue par organisation optimisée des données en mémoire (distributions et placement des données en mémoire) et par la hiérarchie mémoire (la consommation peut être réduite en plaçant les données les plus souvent utilisées dans des mémoires plus petites). La simplification de l'adressage permet de réduire la consommation. Pour des adresses codées sur un grand nombre de bits, il est important de s'assurer qu'un nombre minimum de transitions est effectué entre deux adresses fournies à la mémoire.

b. Etat de l'art

Les approches d'optimisation des unités de mémorisation présentes dans la littérature s'appuient sur les trois paramètres cités dans le paragraphe précédent, et peuvent être regroupées en trois classes.
Une première approche s'intéresse à la conception de hiérarchie et d'architecture mémoire pour des applications dont les séquences d'accès sont connues à priori. Cette connaissance des accès mémoire est généralement obtenue par profiling. De nombreuses méthodologies ont été développées ; elles proposent des techniques permettant de sélectionner une architecture "low power" pour les mémoires caches [2] ou les mémoires SRAM [3]. Le choix des architectures mémoire est effectué par rapport aux estimations en vitesse, surface et consommation obtenues à l'aide de modèles analytiques. Une méthodologie permettant de réduire la surface occupée par l'unité de mémorisation, pour des applications de traitement du signal temps réel, a déjà été développée [4]. Certaines phases de la stratégie de synthèse mémoire peuvent être reprises dans notre étude, en s'orientant plus particulièrement vers la réduction de la consommation.
La seconde technique d'optimisation mémoire considère une hiérarchie et une architecture fixe ; la réduction de la consommation est réalisée par transformation du code. Ces transformations vont permettre de réduire les besoins mémoires, les séquences d'accès mémoire ou le nombre de transitions sur les adresses. Au niveau du code source, une sélection de structures de données adaptées à l'application, permettra de réduire les besoins de mémorisation [5]. De nombreuses techniques d'ordonnancement des accès mémoires [6] et d'assignation des données en mémoire et en registres [7] ont été développées en s'orientant vers l'optimisation de la consommation. Une méthode de gestion explicite de placement des données en mémoire, développé au laboratoire LESTER, permet d'optimiser la consommation de la partie mémoire [8]. D'autres études [9] montrent l'apport de la limitation du nombre de transitions sur les bus d'adresses.
La troisième approche est celle adoptée par l'IMEC [10] sous la dénomination de DTSE (Data Transfert and Storage Exploration). Elle apporte de bons résultats puisqu'elle permet l'optimisation conjointe de l'architecture mémoire et du code de l'application. Cependant le temps nécessaire à l'obtention d'une solution optimale est relativement important car de nombreuses étapes ne peuvent être automatisées et requièrent l'expertise d'un concepteur avisé.

Le cadre de notre étude se situe dans la philosophie de la dernière approche. Cependant l'objectif n'est pas ici de trouver la solution optimale mais de trouver, en se plaçant dans un flot de synthèse d'architecture, un compromis entre le temps de conception et une solution répondant aux contraintes de l'application.

3. Stratégie de conception

Nous nous intéressons ici aux applications de traitement du signal et de l'image ayant un comportement déterministe (les accès mémoires peuvent être connus à l'avance). La synthèse d'architecture de telles applications, caractérisées par des besoins de mémorisation importants, permet d'explorer de nombreuses alternatives d'implémentation de hiérarchie mémoire. Cependant, parmi toutes ces alternatives, seules les architectures conduisant aux meilleures performances et à de faibles consommations doivent être retenues. Aussi, il est nécessaire de contraindre le processus de synthèse et de mettre en œuvre des méthodes afin de restreindre l'espace de conception et aboutir rapidement aux meilleures solutions.

La problématique de la synthèse mémoire peut être énoncée de la façon suivante : Quelle est la solution mémoire optimale, en terme de consommation, permettant de stocker toutes les données tout en assurant leur cohérence? L'analyse de cette problématique conduit à la mise en place d'un flot de conception d'unité de mémorisation s'inscrivant dans la synthèse architecturale. Il faut identifier toutes les phases de synthèse nécessaires à l'obtention d'une architecture et d'une hiérarchie mémoire optimisées. Il apparaît alors important de définir correctement les objectifs des différentes phases identifiées.

Notre stratégie peut se décomposer en deux étapes, comme le montre la figure 1. Dans un premier temps, nous mettrons en place des techniques nous permettant de choisir une hiérarchie et de distribuer les données à travers cette hiérarchie. Les choix s'effectuent par rapport à des fonctions de coûts en vitesse, surface et consommation définies à l'aide de modèles. Une fois cette étape réalisée, nous définissons une stratégie de conception pour chaque niveau de hiérarchie. Nous cherchons à identifier les différentes phases amenant à une architecture mémoire faible consommation. Pour pouvoir mettre en œuvre une stratégie de synthèse, il faut définir de manière précise les objectifs de chaque phase ainsi que leurs dépendances. Les deux étapes de conception sont détaillées dans les paragraphes suivants.

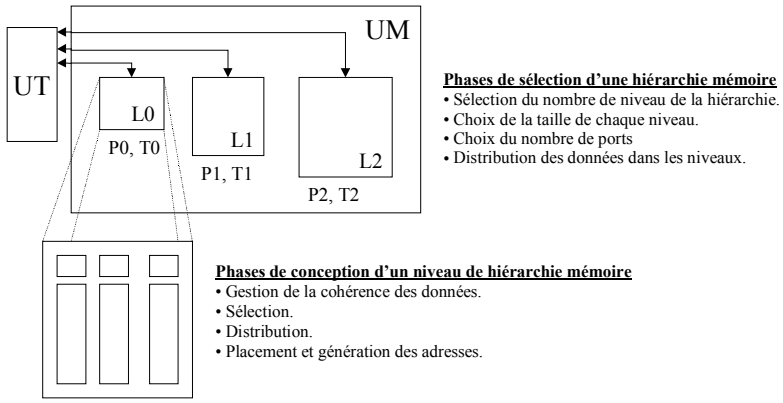

Figure 1 : Phases de conception\sélection d'une hiérarchie mémoire.

a. Choix d'une hiérarchie mémoire

Les niveaux bas de la hiérarchie sont composés de mémoires rapides, de petites tailles, proches de l'unité de traitement. Plus le niveau de hiérarchie augmente, plus les tailles et l'éloignement des mémoires par rapport à l'unité de traitement augmentent. Nous nous intéressons ici aux performances et à la consommation. Les valeurs Pn, et Tn représentent les coûts en consommation et en temps de chaque niveau de la hiérarchie ; Pn représente le coût d'un accès mémoire et le coût de transfert des données à travers la hiérarchie ; Tn représente les temps d'accès permettant d'accéder à un niveau n. En fonction des paramètres architecturaux et algorithmiques, un compromis peut être trouvé pour aboutir à une organisation hiérarchique de l'unité de mémorisation. Les paramètres architecturaux regroupent la taille mémoire et la taille des mots d'une mémoire ; les paramètres algorithmiques sont le nombre de lectures et d'écritures et pour les mémoires cache le nombre de défauts de cache. Les critères de sélection permettant de définir la hiérarchie mémoire sont la taille de chaque niveau de hiérarchie, les coûts en temps et en consommation. Une fois la hiérarchie et les tailles mémoires définies, nous déterminons la distribution des structures de données de l'application en se basant sur leurs durées de vie et leurs tailles. De cette façon, une structure de données ayant une densité d'accès importante sera placée dans un niveau hiérarchique bas si sa taille lui permet.

b. Conception de l'architecture mémoire pour chaque niveau de hiérarchie

Cette étape de conception a pour but de trouver, pour un niveau de hiérarchie donné, une architecture mémoire et une organisation des données de l'architecture. Cette étape est effectuée en plusieurs phases que nous allons détailler.

La gestion de cohérence permet d'éviter les conflits d'accès aux données en définissant un nombre de bancs mémoire suffisant. La gestion de la cohérence permet également de réduire la consommation de la mémoire en limitant les transferts de données.
A partir des besoins de l'application, la sélection de mémoires consiste à choisir, parmi un ensemble de composants, un sous-ensemble de mémoires adaptées aux stockages des données. Le choix de ces mémoires dépendra d'une fonction de coût intégrant des paramètres de vitesse, surface et consommation.
La distribution consiste à allouer une mémoire à chaque donnée. L'objectif principal est de limiter les connexions vers l'unité de traitement et les autres niveaux hiérarchiques.
Le placement dans les bancs mémoires revient à affecter une adresse précise à chaque donnée. Le choix des adresses doit permettre de limiter la complexité des générateurs d'adresses et les transitions sur les bus d'adresses. Les phases de placement de données et de définition de générateurs d'adresses sont fortement dépendantes.
Les différentes phases interagissent plus ou moins entre elles. Avant de définir complètement le flot de conception de l'unité de mémorisation, il faut analyser les inter-dépendances entre toutes les phases et les conséquences de chacune sur le coût global de la solution mémoire. Cette analyse permettra de définir une séquence d'enchaînement des phases lors de la synthèse de mémoire.

4. Perspectives

La stratégie développée n'a pas pour but l'obtention d'une architecture et d'une hiérarchie mémoire optimale. Comme l'espace d'exploration de solutions d'architecture mémoire est très vaste, une recherche exhaustive pour chaque étape amènerait à des temps d'exploration excessifs.

L'intérêt de la stratégie repose ici sur un compromis entre l'obtention d'une solution répondant aux contraintes imposées et le temps de recherche ayant amené à cette solution.

Les travaux futurs s'orienteront vers la modélisation de la consommation des mémoires afin de :
(i) Disposer d'une bibliothèque mémoire.
(ii) De définir correctement les fonctions de coût pour tous les types de mémoire.
(iii) D'identifier les paramètres à prendre en compte.
La stratégie devra être également affinée afin d'intégrer les phases transformations de code et d'ordonnancement des accès mémoire qui n'apparaissent pas encore dans le flot de conception.

5. Bibliographie

[1] F. Catthoor, F.Franssen, S. Wuytack, L. Nachtergaele, "Global communication and memory optimizing transformations for low-power signal processing systems," *In Proceedings of Workshop On VLSI Signal Processing*, pp 178-187, October 1994.

[2] M. Kamble, K Ghose, "Analytical energy dissipation models for low power caches," *In Proceedings of International Symposium. on Low Power Electronic and Design*, pp 143-148, August 1997.

[3] S. Coumeri, D. Thomas, "Memory Modeling for System Synthesis," *In Proceedings of International Symposium on Low Power Electronic and Design*, pp179-184 August 1998.

[4] D Chillet, "Méthodologie de conception architecturale des mémoires pour circuits dédiés au traitement du signal temps réel," Thèse, université de Rennes1, janvier 1997.

[5J Da Silva, F. Catthoor, F. Verkest, H. De Man "Power Exploration for Dynamic Data Types through Virtual Memory Management Refinement," *In Proceedings of International Symposium on Low Power Electronics and Design*, pp. 311-316, August. 1998.

[6] R. Saied, C. Chakrabarti, "Scheduling for Minimizing the Number of Memory Accesses in Low Power Applications," *in Proceedings of VLSI Signal Processing*, pp. 169-178, October 1996.

[7] Y. Zhang, X. Hu, D. Chen, "Low Energy Register Allocation Beyond Basic Blocks," *In Proceedings of International Symposium on Circuits and Systems,* pp. 290-293, June 1999.

[8] S. Pignolo, E. Martin, N. Julien, E. Senn, B. Saget, "Optimisation de la consommation d'énergie des applications de Traitements Du Signal et de l'Image embarquées sur DSP," *3ème journées francophones d'études Faible Tension Faible Consommation,* juin 2001.

[9] P. R. Panda, N. Dutt, A. Nicolau, "Memory issues in embedded System-On-Chip, optimisation and exploration," *Kluwer Academic Publisher,*1999. ISBN 0-7923-8362-1.

[10] F. Catthoor et al, "Custom memory management methodology : Exploration of memory organisation for embedded multimedia system design," *Kluwer Academic Publisher,*1998. ISBN 0-7923-8288-9.