Recherche d'une exploitation énergétique optimale des ressources de calcul dans un système de vision sur puce

M. DULONG, T. M. BERNARD

ENSTA ParisTech
32, boulevard Victor, 75739 Paris Cedex 15, France

marc.dulong@ensta.fr, thierry.bernard@ensta.fr

Résumé – La vidéosurveillance sans fil exigera une exploitation énergétiquement optimale des ressources de calcul dans un système de vision sur puce. En phase d'exploration architecturale, des estimations précoces de consommation sont utiles, d'abord pour se faire une idée des partitionnements optimaux de l'application de vision entre les différentes ressources envisagées, et ensuite pour apprécier l'adéquation de l'ensemble. Cette démarche est ici illustrée sur un cas simple dans lequel une tâche de moyen niveau se retrouve répartie entre une unité scalaire et une unité SIMD.

Abstract – Wireless video surveillance will require using future Vision SoC at their lowest possible power consumption. Then architectural exploration of such VSoC will take advantage of early power estimation, first to get an idea of the best partitioning of a visual application among the considered VSoC elements, and then to assess the VSoC architecture. This is illustrated on a simple case study in which a middle-level visual task gets shared among both a scalar and a SIMD unit.

1 Introduction

Il faudra encore fortement réduire la puissance consommée par les architectures de vision avant de pouvoir assurer durablement une vidéosurveillance sans fil. La vision basse consommation recourt volontiers à un parallélisme SIMD massif, pour son excellente efficacité énergétique sur les traitements réguliers [1, 2] ainsi qu'à des ressources de calcul plus classiques tel que des processeurs scalaires. Dans la perspective de réduire fortement la consommation, l'estimation précoce de la puissance consommée devient essentielle pour déterminer une exploitation énergétique optimale des unités de calcul.

On s'intéresse ici au partitionnement d'une tâche de vision de moyen niveau entre une unité SIMD et un processeur scalaire, au sein d'un système de vision sur puce, selon un critère énergétique. La tâche considérée concerne le codage de forme, en vue d'une transmission compacte d'information sur une liaison sans fil. L'optimisation du partitionnement est réalisée par modélisation et simulation transactionnelle TLM2.0 [4] des diverses unités en jeu, enrichie d'aspects énergétiques. Le niveau d'abstraction utilisé permet en outre une visualisation et une analyse proche du temps réel des aspects logiciels et matériels de l'exécution du système à partir de stimuli acquis en temps réel. Ci-dessous, la chaîne algorithmique et l'architecture matérielle du système sont tout d'abord introduites. La modélisation énergétique TLM 2.0 du système sera ensuite présentée. Différentes stratégies de partitionnement seront évaluées et une analyse des résultats obtenus permettra de conclure.

2 Architecture du VSoC

2.1 Architecture matérielle

Un système de vision sur puce (VSoC) simple est considéré, constitué d'un processeur scalaire, d'une unité SIMD, de mémoires et d'une unité de transmission sans fil (figure 1). L'unité SIMD est pilotée par le processeur scalaire, via un contrôleur/séquenceur.

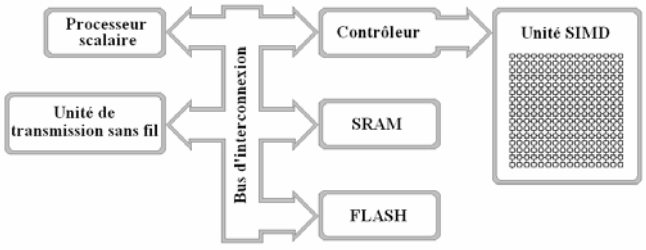

Figure 1 : Architecture matérielle

L'unité SIMD prend naturellement en charge la vision bas niveau et le processeur scalaire celle de haut niveau. Entre les deux, à moyen niveau, l'affectation est moins évidente : c'est là que la minimisation de la consommation d'énergie peut avoir les conséquences les plus inattendues.

2.2 Chaîne algorithmique

Un algorithme de détection de mouvement [5] détecte les formes en mouvement dans la scène observée et les fournit à un algorithme de codage de forme de type quadtree. L'arbre quadtree représentant la forme est élagué afin de ne conserver que la silhouette dominante,

permettant ainsi de réduire les données à transmettre (figure 2).

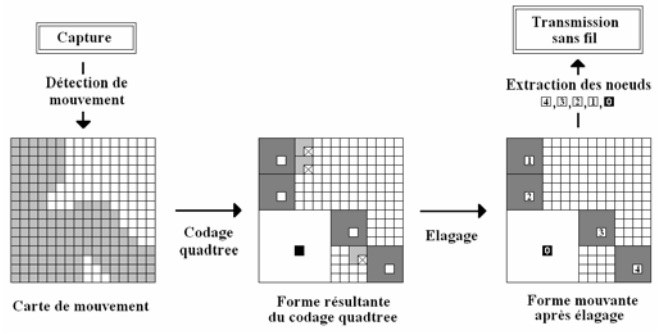

Figure 2 : Chaîne algorithmique

L'algorithme de codage de forme utilise une représentation de l'image sous forme d'arbre quadtree. Chaque noeud de l'arbre correspond dans l'image au centre (décalé d'un demi-pixel) d'un carré de côté 2^n où *n* correspond à la hauteur du noeud dans l'arbre, appelée *niveau de résolution*. L'arbre représentant la forme en mouvement est construit en commençant par les feuilles (niveau de résolution 1). L'ensemble des centres des carrés de largeur 2^1 dont l'union représente la forme en mouvement sont déterminés et regroupés par quatre pour former des carrés de taille 2^2, noeuds du niveau de résolution suivant, et ainsi de suite. Sur la figure 2, seuls trois niveaux de résolution sont illustrés.

Afin de diminuer la consommation liée à la transmission sans fil de la forme mouvante à une unité distante, l'arbre quadtree est élagué en supprimant les niveaux de résolution les plus faibles (détails non significatifs). Sur la figure 2, le niveau 1 est supprimé et seuls les nœuds des niveaux 2 et 3 sont communiqués à l'unité distante. Il est encore possible de distinguer une tête, un torse et un bras.

La détection de mouvement ainsi que la tâche de codage de forme aux faibles niveaux de résolution sont efficacement pris en charge par l'unité SIMD. En revanche, aux niveaux de résolution plus élevés pour lesquels l'information à traiter est parcimonieuse, une unité scalaire est plus adaptée. Il est donc probable qu'une solution énergétique optimale consiste à partitionner la tâche de codage de forme entre ces deux unités, sous réserve d'attacher à l'unité SIMD une ressource d'extraction de points permettant la transmission des nœuds de l'arbre quadtree vers le processeur scalaire.

3 Modélisation du VSoC

3.1 Modélisation de l'architecture

L'architecture matérielle et logicielle a été modélisée en employant la méthodologie transactionnelle TLM 2.0. Le haut niveau d'abstraction permet de simuler le système de vision en temps réel, à partir du flux vidéo fourni en live par une webcam. Afin de réduire le temps de développement, le processeur a été modélisé en utilisant un simulateur de jeu d'instruction SIMIT 3.0 simulant l'architecture ARM v4 [6]. L'unité SIMD modélisée est une rétine artificielle programmable, fusion entre un imageur CMOS et une grille SIMD bidimensionnelle. Un tel rapprochement permet de réduire de façon drastique la consommation liée aux transferts de données. Chaque pixel contient une photodiode, un convertisseur A/N, une unité logique, quelques dizaines de registres binaires et une unité de communication. Les autres composants du VSoC ont également été modélisés : bus, UART, DMA, FLASH, SRAM, SDRAM, cache. Les outils de modélisation développés ne sont pas limités à la simple simulation du système considéré, d'autres architectures peuvent ainsi être simulés. Une stratégie de raffinement des modèles de composant a été utilisée. Les modèles « untimed » ont été progressivement enrichis en «loosely-timed » avec timing. Le mécanisme de découplage temporel a été employé pour augmenter la vitesse de simulation.

Une simulation débute par la lecture d'un fichier de configuration contenant les caractéristiques du système (tension d'alimentation et fréquence de fonctionnement du processeur, nombre de registres binaires pixelliques pour la rétine …) ainsi que les paramètres de simulation (temps total ou nombre de trames à traiter …). La fin d'une simulation génère un fichier de statistiques reprenant l'activité totale et la consommation énergétique des différents composants (figure 3).

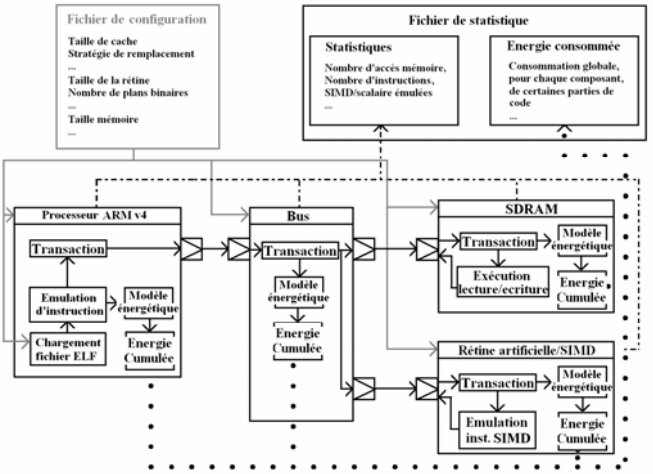

Figure 3 : Plateforme de simulation pour une architecture simplifiée de système de vision

3.2 Estimation énergétique

Les méthodes d'estimation énergétique à haut niveau d'abstraction utilisent pour la plupart des modèles énergétiques sophistiqués se rapprochant au plus près des mesures physiques ou des simulations au niveau transfert entre registres [7, 8]. L'exploitation de ces méthodes dans un contexte d'exploration architecturale est difficile compte tenu de l'effort de modélisation requis et de la vérification exhaustive mise en jeu. L'approche proposée utilise des modèles énergétiques peu complexes, comparables à [9, 10] et une vérification allégée des estimations énergétiques. Cela permet de réduire l'effort de modélisation pour se focaliser sur l'exploration architecturale au détriment de la précision des estimations. La validation joue alors un rôle

essentiel afin de s'assurer que les estimations conservent un sens.

Le modèle énergétique du processeur ARM-V4 est similaire à celui employé dans [9]. Le jeu d'instruction a été différencié en 4 classes selon la valeur du courant moyen mesuré pour une tension d'alimentation de 1,5 Volts et une fréquence de fonctionnement de 206 MHz (tableau 1).

Tab 1 : Classe d'instruction ARM-V4

Classe	Instructions	Courant moyen (A)
1	STR, STRB, STRBT, STRT, LDM, STM	0.230
2	TST, MLA, LDR, LDRB	0.207
3	MRS, MSR, SWP, B, NOP	0.169
4	Autres	0.179

L'énergie pour une instruction est ainsi obtenue en utilisant l'équation (1) :

$$E_{inst} = V_{dd} \times I_{class} \times n_{cyc} \times T_{cyc} \quad (1)$$

Dans ce modèle, seul le mode actif du processeur a été pris en compte, celui-ci peut néanmoins être étendu à d'autres modes, tensions et fréquences de fonctionnement en considérant un modèle de deuxième ordre similaire à [9].

Le modèle énergétique de l'unité SIMD a été développé en utilisant la valeur énergétique moyenne par cycle déduite d'une analyse du layout de la rétine artificielle récemment conçue par l'équipe [3] ainsi que du nombre de cycles d'exécution de chaque instruction SIMD. L'énergie d'une instruction SIMD est donnée par l'équation (2) :

$$E_{inst} = E_{cyc} \times n_{cyc} \quad (2)$$

Pour la mémoire SRAM et cache, seule la consommation énergétique pour une lecture/écriture a été modélisée. La consommation pour les bus d'adresse et de données a été estimée en considérant l'équation (3) :

$$E_{bus} = S_{act} \times C_{bus} \times V_{dd}^2 \quad (3)$$

S_{act} correspond à l'activité de commutation du bus, celle-ci est obtenue en calculant la distance de hamming entre deux mots d'adresses ou de données successifs. C_{bus} représente la capacité par fil de bus. V_{dd} est la tension d'alimentation.

La mémoire FLASH ainsi que l'unité de transmission sans fil, jouant un rôle limité dans la recherche d'un partitionnement optimal de la tâche de codage de forme, ne disposent pas d'un modèle énergétique.

3.3 Validation

La validation des estimations énergétiques repose bien souvent sur des descriptions au niveau RTL ou portes logiques [7, 8]. Il est cependant contradictoire avec une véritable approche « Top-down » de devoir instancier une architecture particulière à des fins de vérification.

Une alternative consiste à vérifier les estimations énergétiques par des mesures physiques sur carte microcontrôleur, similairement à [11]. L'effort de validation est moindre, mais les résultats plus grossiers. Seul le comportement asymptotique des modèles énergétiques peut être vérifié. Ceci permet néanmoins leur calibration et le calcul d'une erreur globale d'estimation. Nous avons suivi ce protocole expérimental en utilisant la carte microcontrôleur AT91EB40A qui comporte un processeur ARM-V4. La présence de jumpers facilite la mesure du courant moyen consommé lors de l'exécution répétitive d'un programme de benchmark. La valeur obtenue est comparée au courant moyen calculé en simulation. Il apparaît des écarts allant jusqu'à 20%.

Il n'est en fait guère utile de chercher à être plus précis puisque les paramètres technologiques de l'ASIC fabriqué peuvent eux-mêmes s'écarter d'environ 30% des valeurs typiques, comme estimé par l'ITRS [12] pour 2010.

Ces incertitudes conditionnent bien sûr la finesse des considérations énergétiques que l'on peut mener a priori, donc les conclusions que l'on peut en tirer au profit de l'exploration architecturale.

4 Résultats

Les outils de modélisation énergétique développés ont été appliqués au VSoC considéré afin de déterminer un partitionnement énergétique optimal de la tâche de codage de forme entre unité scalaire et SIMD. La figure 4 représente le résultat du codage quadtree appliqué à une forme en mouvement.

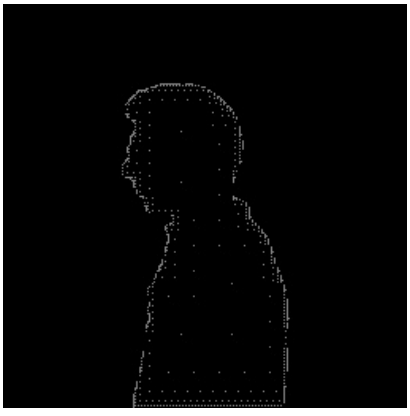

Figure 4 : Détection de mouvement et codage de forme quadtree sur plateforme de simulation

Différents partitionnements de la tâche de codage de forme ont été estimés énergétiquement en utilisant le même stimulus afin de se placer dans des conditions de comparaison équitables. Les tendances présentées restent néanmoins observables pour d'autres stimuli. La

rétine artificielle a été prise de taille 256x256, fixant le niveau de résolution maximal du quadtree à 8. Un partitionnement noté Pn correspond à élaborer les niveaux 1 à n du quadtree sur unité SIMD et les autres, de n+1 à 8, sur unité scalaire. Le partitionnement P0 correspond à un codage de forme réalisé intégralement par le processeur scalaire tandis que pour P8, le codage est réalisé uniquement sur unité SIMD.

La figure 5 représente l'évolution pour différents partitionnements de la consommation due à la détection de mouvement et au codage de forme SIMD et scalaire.

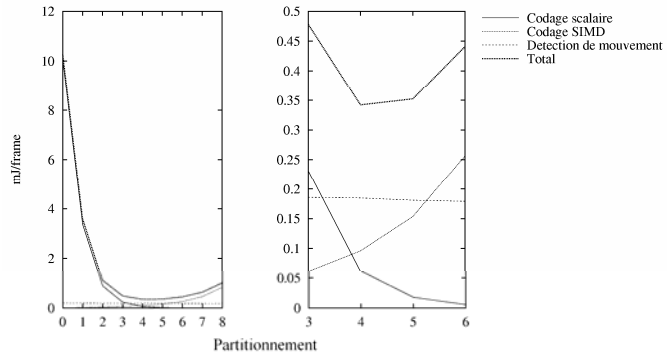

Figure 5 : Energie consommée selon le partitionnement

Les courbes de consommation mettent en évidence une plage optimale située entre P2 et P8. Cette plage présente un gain énergétique supérieur à 10 par rapport à P0. P4 réalise le minimum de consommation, 34% en dessous de P8. Compte tenu des incertitudes d'estimation, la position de ce minimum est imprécise. En revanche, son existence est certaine : pour minimiser la consommation du VSoC réel, le codage en quadtree devra être réparti entre unités scalaire et SIMD. Cet élément précoce d'information est précieux pour l'exploration architecturale. Il incite à s'intéresser de plus près au mécanisme d'extraction de points épars, de l'unité SIMD vers l'unité scalaire.

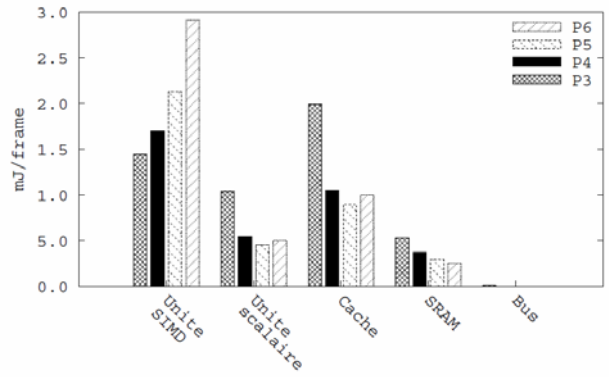

Figure 6 : Répartition énergétique

Par ailleurs, l'examen en figure 6 des consommations individuelles des constituants du VSoC révèle que, autour de l'optimum, c'est l'unité scalaire et les composants de mémorisation qui consomment le plus. Ceci résulte largement du pilotage de l'unité SIMD. Notre modélisation énergétique de haut niveau révèle donc a priori qu'il sera productif, pour diminuer la consommation totale, d'alléger cette charge de pilotage par mise en place de ressources dédiées.

5 Conclusion

Grâce à une modélisation système de haut niveau, enrichie de modèles énergétiques, il est possible à la fois d'essayer en temps réel une application répartie sur les différentes briques d'un système de vision sur puce et d'apprécier la part de chacune dans la consommation totale. Ceci permet de rechercher a priori la meilleure solution pour un couple application architecture donné. Une fois positionné à l'optimum, il est alors possible de faire évoluer efficacement l'architecture pour atteindre de meilleures performances globales. Nous avons illustré cette démarche sur un cas simple mais probant.

Références

[1] Kim, K. and al.: A 125 GOPS 583 mW Network-on-Chip Based Parallel Processor With Bio-Inspired Visual Attention Engine. IEEE Journal of Solid-State Circuits, vol. 44, no. 1, pp. 136-147, 2009.

[2] Cheng, C.-C. and al. : iVisual: An Intelligent Visual Sensor SoC With 2790fps CMOS Image Sensor and 205 GOPS/W Vision Processor. IEEE Journal of Solid-State Circuits, vol 44, no. 1, pp. 127-135, 2009.

[3] Bernard, T.: Capteur matriciel à processeurs numériques distribués. Patent pending n° 0855014, 23 July 2008.

[4] OSCI: TLM 2.0 User Manual. http://www.systemc.org/downloads/standards/.

[5] Manzanera, A.; Richefeu, J.C.: A new motion detection algorithm based on Σ-Δ background estimation. Pattern Recognition Letters, Elsevier, vol. 28, pp. 320-328, 2007.

[6] D'Errico, J.; Qin W.: Constructing Portable Compiled Instruction-set Simulators - An ADL-driven Approach. IEEE/ACM Design Automation and Test in Europe, 2006.

[7] Lee, I. and al.: PowerVip: SoC Power estimation Framework at Transaction Level. Proceedings of the 2006 Conference on Asia South Pacific Design Automation, pp. 551-558, 2006.

[8] Givargis, T.; Vahid F.: Platune: A tuning Framework for System-On-Chip Platforms. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 21, no. 11, 2002.

[9] Sinha, A.; Chandrakasan, A.P.: JouleTrack-a Web based tool for software energy profiling. Proc. Design Automation Conference, pp. 220-225, 2001.

[10] Bansal, N.; Lahiri, K.; Raghunathan, A.; Chakradhar, S.: Power Monitors: A Framework for System-Level Power Estimation Using Heterogenous Power Models. 18th International Conference on VLSI Design, pp. 579-585, 2005.

[11] Varma, A.; Jacob B.: Accurate and Fast System-Level Power Modeling: An XScale-Based Case Study. ACM Transactions on Embedded Computing Systems, vol. 6, no. 4, 2007.

[12] International Technology Roadmap for Semiconductors. 2007. http://www.itrs.net