



HAL
open science

Estimation de la consommation logicielle dans un système embarqué: Etude de cas

Jalel Ktari, Johann Laurent, Mohamed Abid, Nathalie Julien

► **To cite this version:**

Jalel Ktari, Johann Laurent, Mohamed Abid, Nathalie Julien. Estimation de la consommation logicielle dans un système embarqué: Etude de cas. 2005, pp.55-59. hal-00077590

HAL Id: hal-00077590

<https://hal.science/hal-00077590v1>

Submitted on 31 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Estimation de la consommation logicielle dans un système embarqué : Etude de cas

Jalel KTARI¹⁻², Johann LAURENT², Mohamed ABID¹, Nathalie JULIEN²

¹CES-Ecole Nationale d'Ingénieurs de Sfax, BPW 3038 Sfax Tunisie.

²LESTER-Université de Bretagne Sud-Lorient, BP 92116-56321 Lorient Cedex

ktari@iuplo.univ-ubs.fr, johann.laurent@univ-ubs.fr, mohamed.abid@enis.rnu.tn, nathalie.julien@univ-ubs.fr

Résumé

L'optimisation en consommation de l'énergie des systèmes embarqués est devenue une priorité majeure lors de la conception. Cet article présente une nouvelle méthode d'estimation de la consommation dans les systèmes embarqués. L'approche proposée exploite des modèles paramétriques permettant de représenter le comportement en consommation d'une architecture. Il s'agit de dégager les lois de consommation au niveau système permettant une modélisation à un niveau avancé lors de la conception. Cette approche permet de déduire la puissance et l'énergie consommés à partir d'un code en langage évolué (ANSI-C) pour une cible donnée. Nous démontrons la faisabilité et l'intérêt de l'approche sur une application classique de traitement du signal et de l'image.

1. Introduction

Avec la tendance récente vers la communication portative, la consommation de puissance et d'énergie est devenue une des contraintes principales lors de la conception qui s'ajoute à celle de la surface et de la vitesse [1].

Les nouvelles applications telles que les ordinateurs portables, les télécommunications sans fil (radiomobile, GSM,...), les assistants de personnes (PDA) possèdent la plus forte croissance du secteur électronique. Les projections dans le futur montrent que cette croissance doit continuer. Ce type d'applications intègre des fonctionnalités complexes (codage parole ou vidéo, traitements type multimédia,...), qui demandent des calculs performants, tout en ajoutant une contrainte forte sur la consommation de ces systèmes[2].

La partie logicielle constitue un composant principal des systèmes d'aujourd'hui mis en oeuvre avec des microprocesseurs. Plusieurs travaux étudient l'estimation de la consommation selon différents niveaux d'abstraction

Au niveau cycle près, l'outil SES (Seoul National University Energy Scanner) estime la consommation de puissance et d'énergie des données cycle par cycle[3]. Il associe les informations collectées sur la consommation des données avec le code C ou assembleur. SES ne nécessite pas une plate-forme de mesure additionnelle puisqu'il intègre les caractéristiques de puissance de la cible.

Au niveau RTL, on peut citer le DSP-PP [4], c'est un outil de simulation permettant l'estimation de la puissance dissipée pour les DSPs. Il est composé de deux composants: le simulateur de performance au niveau de cycle (CPS) et l'estimateur de dissipation de puissance (PDE). Il est écrit en C++ permettant de considérer des modèles abstraits. Les composants du DSP sont modélisés comme des objets intégrant le modèle de consommation. DSP-PP considère la simulation détaillée au niveau cycle de tous les composants du DSP: les chemins de données et l'interconnexion et estime exactement la valeur de puissance dynamique, de court circuit et de fuite de chaque composant du DSP.

La simulation cycle-précis entraîne une vitesse de simulation extrêmement lente plus particulièrement les applications complexes sur des modèles de processeur détaillés.

Par ailleurs, l'importance croissante de la partie logicielle dans ces systèmes embarqués nécessite l'analyse de la consommation à un niveau avancé de la conception [1]. De plus, les optimisations aux niveaux architecturaux et algorithmiques apportent des gains plus importants

que celles réalisées au niveau technologique. En effet, si l'optimisation de la consommation doit intervenir à chaque niveau de la conception, c'est cependant aux plus hauts niveaux que les gains attendus sont les plus importants. Le gain au niveau technologique peut atteindre 50% et 20x au niveau système [5].

Par exemple, l'outil d'estimation JouleTrack ne modélise pas le programme, le modèle de processeur est composé simplement de la fréquence de fonctionnement et la tension d'alimentation [6].

L'objectif de ce travail est de définir une approche d'estimation de la consommation au niveau système. Pour cela, nous proposons non pas des modèles de consommation de l'architecture cible mais des modèles des algorithmes eux-mêmes. Des modèles paramétriques liant la consommation (puissance et énergie) avec les aspects architecturaux de cibles (DSPs) ont été utilisés [7]. Ici, nous appliquons cette méthode non plus à l'architecture mais directement sur l'algorithme afin de le caractériser d'un point de vue consommation.

Dans un premier temps, nous rappelons la méthodologie d'analyse fonctionnelle dite « FLPA » (Functional Level Power Analysis)[7]. Puis nous introduisons son extension afin de considérer les spécifications algorithmiques. Ensuite nous expertiserons les modèles proposés sur une application de traitement du signal et de l'image. Enfin, nous discuterons des résultats obtenus par rapport aux mesures physiques réalisées sur des cartes de développement.

2. Démarche de conception des modèles

La méthodologie de caractérisation « FLPA » permet, à partir de l'analyse fonctionnelle, de développer un modèle paramétrique permettant de représenter le comportement en consommation d'une architecture cible. Cette méthodologie est étendue ici afin d'expertiser la consommation au niveau algorithmique en fonction de la variation des paramètres influant sur la consommation de l'application.(figure 1)

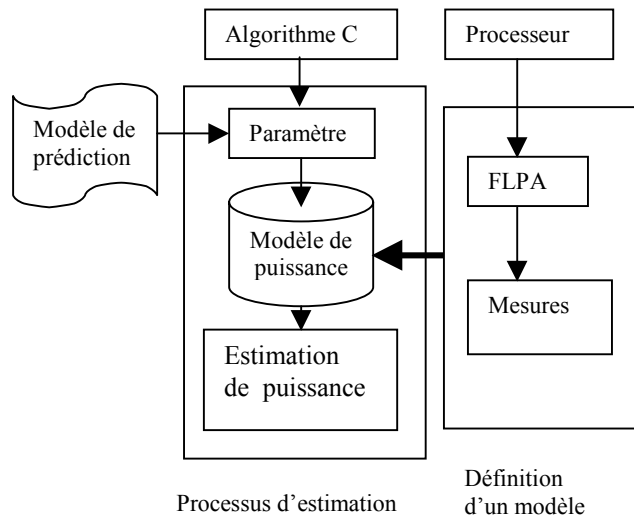


Figure 1 : Méthodologies de l'estimation

Ce travail est basé sur l'utilisation de 2 modèles de DSPs de la firme Texas Instruments (C5510 & C6701) intégrés dans l'outil SoftExplorer [8]. Ils exploitent cette méthodologie d'analyse fonctionnelle « FLPA ».

La structure de SoftExplorer est illustrée dans la figure 2. Il est basé sur trois modèles complémentaires :

- Le modèle du processeur : également appelé le modèle de puissance représente la manière dont la consommation du processeur change avec son activité.
- Le modèle de l'algorithme : représentant le lien entre l'algorithme et l'activité qu'il induit dans le processeur.
- Le modèle du compilateur : également appelé le modèle de prédiction qui représente le comportement du compilateur qui dépend des options choisies par le programmeur pendant la compilation, avec un fort impact sur le code généré, et ainsi sur l'activité du processeur.

Nous proposons une extension de l'approche afin de considérer les spécifications algorithmiques. Nous expertisons les modèles proposés sur une application de traitement du signal et de l'image, un filtre FIR (filtre à réponse impulsionnelle finie). La modélisation de l'application FIR écrite au niveau C porte sur la recherche des paramètres influant sur la consommation. Les paramètres considérés sont :

- Algorithmique : l'ordre du filtre (de 8 jusqu'à 256),
- Architectural : la fréquence de fonctionnement (30 MHz jusqu'à Fmax du DSP) et
- Technologique : le type du DSP utilisé (C6701 & C5510).

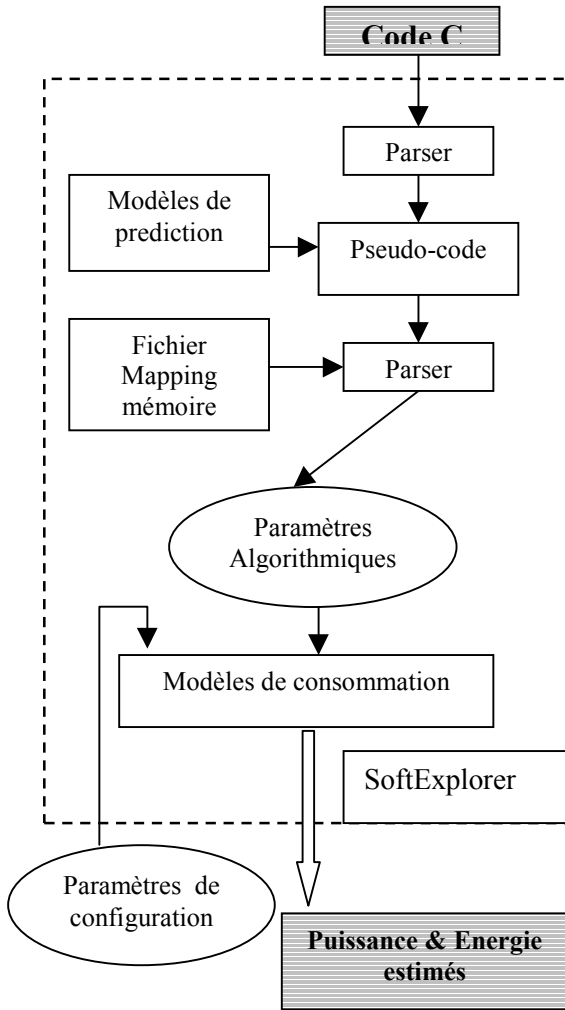


Figure2 : Flot d'estimation avec SoftExplorer

A travers cette étude et grâce à l'outil SoftExplorer, permettant de déterminer le temps, la puissance et l'énergie consommés par le code C pour une cible donnée, un modèle de consommation est établi en fonction de ces paramètres. En effet, ce modèle est déduit en faisant varier les paramètres déjà cités et en exploitant les estimations données par SoftExplorer pour différentes fréquences et cibles.

3. Modèles proposés

La figure 3 donne une idée générale de la consommation du FIR sur le DSP C6701 en fonction de la fréquence pour les ordres de 8 à 256.

Les modèles de consommation (temps, puissance et énergie) pour les deux processeurs (C6701 & C5510) sont donnés dans le Tableau 1.

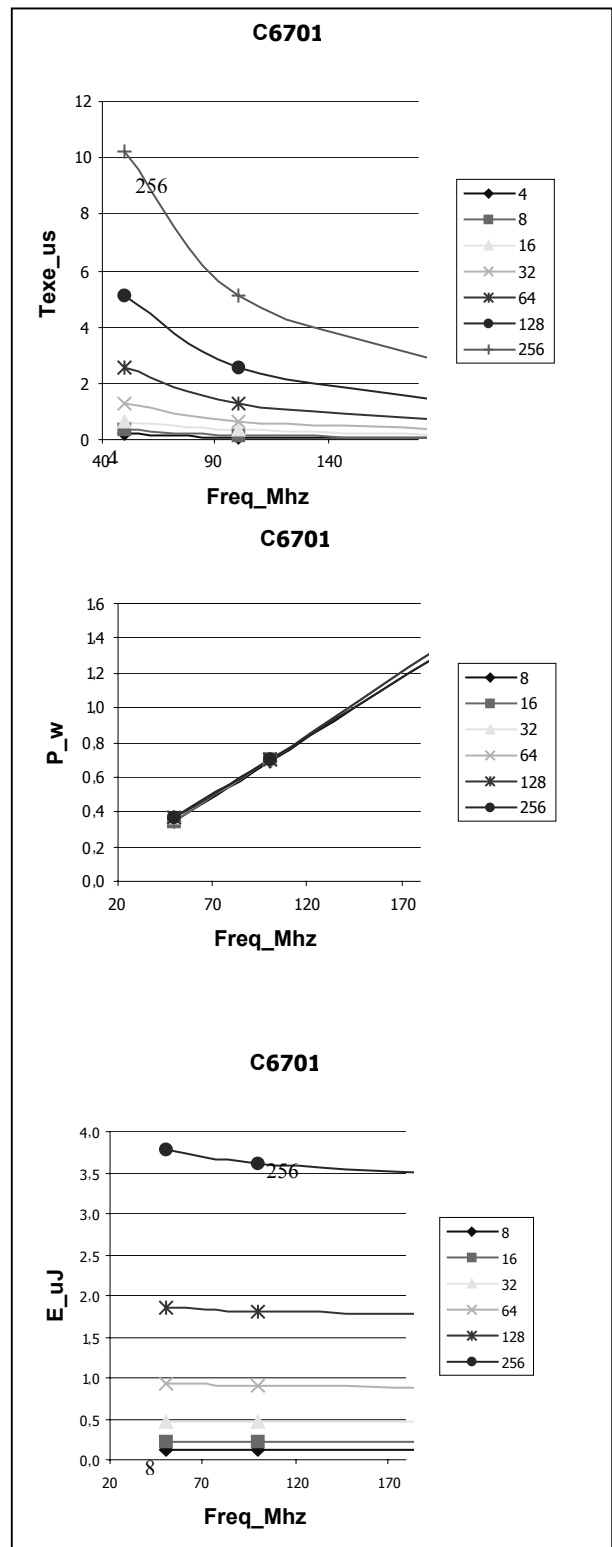


Figure 3 : Tendence de la consommation du FIR sur le C6701 pour différents ordres

Tableau 1 : Modèles de consommation du FIR pour le C6701 & C5510

Fréq (Mhz)	T (uS)	P (W)	E (uJ)
Modèle 1	2,006*ordre/ freq	0,007*freq	0,014*ordre
C6701			
Erreur max /mesure	9,5%	8,8%	13%
Erreur moyenne	5,2%	3,5%	9%
Modèle 2	2,006*ordre/ freq	0,0072*freq	0,0144* ordre
Erreur max /mesure	9,5%	6,5%	10%
Erreur moyenne	5,2%	2,9%	7,1%
C5510			
Modèle 1	3,013*ordre/ freq	0,2758 10 ⁻² *freq	0,831 10 ⁻² *ordre
Erreur max /mesure	9%	7%	14%
Erreur moyenne	5%	4%	9,3%
Modèle 2	3,013*ordre/ freq	0,27589 10 ⁻² *freq	0,83125 10 ⁻² *ordre
Erreur max /mesure	9%	6,6%	13,1%
Erreur moyenne	5%	3%	7,6%

Modèle 1 : Avec l'outil SoftExplorer

Modèle 2 : Avec des mesures physiques sur carte

On remarque bien que le modèle du temps nécessaire à l'exécution du code sur le DSP varie linéairement avec l'ordre (le temps double si on double l'ordre du filtre). L'énergie est quasi invariante même si la fréquence change, contrairement à l'ordre qui influe de façon linéaire.

Concernant l'ordre 8, l'erreur maximale de modélisation théorique du temps d'exécution par rapport à SoftExplorer est de 8,5% alors qu'à partir de l'ordre 16, l'erreur maximale n'est que de 3,6% pour les deux DSPs. Ceci peut s'expliquer par le fait que pour les ordres faibles, le compilateur n'arrive plus à compiler le code de façon à utiliser le parallélisme maximum alors que SoftExplorer considère que l'architecture est utilisée au mieux.

En ce qui concerne la loi de puissance, le modèle est fonction de la fréquence seulement et

dont l'erreur maximale est de 7% par rapport aux estimations fournies par SoftExplorer.

A la vue de ces deux modèles, nous voyons que bien que le C67 permet de diviser par 1,5 le temps d'exécution de l'application, la consommation de puissance et d'énergie sont quant à elles plus élevées. Ces résultats montrent bien que le C55 est un processeur basse consommation comparé au C67.

4. Expérimentations et analyse des Résultats

Des mesures physiques sur des cartes de développement (à base de C6701 & C5510) sont faites afin de vérifier la validité du modèle établi en fonction des paramètres de l'application. Le modèle de l'application FIR pour le C6701 présente une erreur qui diminue lorsque la fréquence et l'ordre augmentent. L'erreur maximale est de 8,8% pour la puissance et de 13% pour l'énergie (cf. Tableau 1). Le modèle de l'application FIR pour le C5510 présente une erreur maximale de 7% pour la puissance et de 14% pour l'énergie (cf. Tableau 1).

Les modèles de consommation ayant été développés en utilisant un outil d'estimation, il est possible de les raffiner en utilisant les mesures physiques de consommation réalisées sur les cartes de développement (Tableau 1). Les nouveaux modèles, basés sur les mesures expérimentales, permettent de diminuer les erreurs d'estimation: l'erreur maximale sur la puissance est alors de 6,5%, alors que l'énergie présente une erreur inférieure à 10% pour le C6701.

Les mêmes améliorations sont faites pour le modèle de l'application exécutée sur le C55. Un deuxième modèle plus raffiné est ainsi établi, minimisant l'erreur maximale sur l'énergie à 13,1%.

Quel que soit le modèle, l'erreur moyenne, que ce soit pour le temps d'exécution, la puissance et l'énergie, reste inférieure à 10%.

Il est à noter que cette approche permet de réduire le temps de conception à travers la modélisation au niveau système et l'outil SoftExplorer. En effet il suffit d'annoter les parties dynamiques du code C [9] afin d'obtenir les estimations (temps d'estimation inférieur à la seconde); alors qu'en utilisant des mesures physiques, le temps de modélisation est plus important et nécessite des plates formes pour les mesures.

Etant donné que SoftExplorer réalise une trace statique du programme, l'utilisateur annote le code C avec "des commentaires magiques" qui permettent à l'outil de savoir la probabilité des

structures conditionnelles et le nombre d'itérations pour les boucles dynamiques.

Ces commentaires magiques sont seulement employés par SoftExplorer et ceux-ci sont traités en tant que commentaires par le compilateur. C'est pour cela l'annotation du code C nécessite une bonne connaissance de la taille des boucles dynamiques ainsi que la probabilité d'accéder à un branchement conditionnel.

L'erreur moyenne induite par l'utilisation de SoftExplorer est de 1,8% (sur l'énergie) par rapport aux modèles dégagés par mesures alors que nous avons un rapport 10 en terme de temps de modélisation en utilisant SoftExplorer.

5. Conclusion et perspectives

A travers ce travail, nous avons montré l'intérêt et la faisabilité de la modélisation de la consommation au niveau algorithmique. Les modèles et l'environnement proposés permettent d'estimer à un niveau système la consommation de l'application en fixant seulement la fréquence et l'ordre avec une erreur maximale sur l'énergie de 13%.

Il est nécessaire de disposer d'une estimation de haut niveau fiable qui permette au concepteur non seulement de choisir le processeur le plus adapté à son application et mais également de régler sa fréquence de fonctionnement en fonction de ses contraintes.

De tels modèles peuvent être utilisés, par exemple, par un système d'exploitation qui pourrait choisir les paramètres de l'algorithme pour respecter les contraintes de consommation en fonction du contexte. Nous aurions donc une approche de gestion de la consommation au niveau algorithmique afin de réaliser un contrôle adaptatif de la consommation. Ces travaux ouvrent donc de nouvelles possibilités dans la prise en compte de la consommation dans la conception système d'applications embarquées.

6. Références

[1] N. Julien « Une approche logicielle de la consommation dans les systèmes embarqués », *1^{er} Congrès international de Signaux, Circuits & Systèmes*, Tunisie, Mars 2004.

[2] O. Sentieys « Réduction de consommation d'énergie en électronique embarquée », *Journée Scientifique Electronique Embarquée*, France, Avril 1997.

[3] D. Shin, H. Shim, Y. Joo, H. Yun, J. Kim, N. Chang, « SES: A highly integrated energy monitoring tool for low-power embedded programs », *IEEE Design and Test of Computers*, Juillet 2002, pp. 7 – 17.

[4] D Q. Minh, L. Bengtsson, PL. Edefors « DSP-PP: A simulator /estimator of power consumption and performance for parallel DSP architectures », *Proc. 21st IASTED International Conference Applied Informatics*, Austria, Février 2003.

[5] J.M. Rabaey, M. Pedram, *Low Power design Methodologies*, Edition Kluwer Academic Publishers London 1996.

[6] A. Sinha, A. P. Chandrakasan "JouleTrack - A Web Based Tool for Software Energy Profiling", in *Proc. DAC*, June 2001, p220.

[7] J. Laurent, N. Julien, E. Senn, E. Martin « Functional Level Power Analysis: An Efficient Approach for Modeling the Power Consumption of Complex Processors », *Conférence IEEE DATE 2004*, Paris, Février 2004.

[8] E. Senn, J. Laurent, N. Julien, E. Martin, « SoftExplorer: estimation of the power and energy consumption for DSP applications », *Conférence IEEE EDERS'04*, Birmingham Royaume Uni, Novembre 2004.

[9] <http://lester.univ-ubs.fr:8080>