



Détermination d'un profil de puissance et réduction des pics de puissance dans les systèmes embarqués

Denis Dupont, Jean-Marc Delosme, Frédéric Amiel, Thomas Ea

► To cite this version:

Denis Dupont, Jean-Marc Delosme, Frédéric Amiel, Thomas Ea. Détermination d'un profil de puissance et réduction des pics de puissance dans les systèmes embarqués. 5èmes Journées d'Etude Faible Tension Faible Consommation (FTFC 2005), May 2005, Paris, France. pp.81–85. hal-00341372

HAL Id: hal-00341372

<https://hal.science/hal-00341372>

Submitted on 16 Apr 2014

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.

Détermination d'un profil de puissance et réduction des pics de puissance dans les systèmes embarqués

D. DUPONT, J.-M. DELOSME
IBISC FRE 2873
Université d'Evry
dupontldelosme@ibisc.univ-evry.fr

F. AMIEL, T. EA
ISEP
28, rue Notre-Dame des Champs - 75006 PARIS
thomas.ealfrederic.amiel@isep.fr

Résumé

Dans cet article nous proposons d'estimer le profil de puissance d'un système embarqué et de réduire par l'utilisation de la technique DVS les pics de puissance demandés à l'alimentation. Ces deux méthodes sont basées sur le placement relatif des tâches sur un plan obtenu par divisions récursives en deux parties par des segments de droites verticaux ou horizontaux. Chaque tâche est caractérisée par un ensemble de couples (temps, puissance). On parle alors de tâches flexibles. En jouant sur cette flexibilité, la première méthode définit de manière constructive le profil de puissance du système et permet de fixer parmi tous les choix possibles un couple (temps, puissance) pour chaque tâche. La seconde méthode, indépendante de la première, tente de manière itérative de décongestionner des intervalles de temps à forte demande en puissance.

1. Introduction

La complexité croissante des systèmes embarqués impose de les implanter sur plusieurs entités. L'utilisation conjointe de processeurs programmables (logiciel) et de circuits spécialisés (matériel) permet de trouver un bon compromis entre les performances et le coût du système. Le partitionnement logiciel/matériel désigne ce compromis. Il se divise en trois étapes interdépendantes : l'allocation, l'affectation et l'ordonnancement. L'allocation détermine les types et nombres de composants matériels utilisés, l'affectation assigne chaque fonctionnalité issue des spécifications à un de ces composants et l'ordonnancement donne le séquençement de chaque fonctionnalité en tenant compte des communications inter-composants. Le système est validé si les estimations de performances et de coût respectent les contraintes.

La puissance dissipée par le système est un paramètre critique pour la conception. La méthode DVS (Dynamic Voltage Scaling) permet de diminuer la puissance dissipée par une tâche en diminuant la tension d'alimentation appliquée à la ressource qui exécute cette tâche, en effet :

$$(1) \quad P_d = \alpha C_l V_{dd}^2 F + I_0 10^{-V_t/S} V_{dd}$$

$$(2) \quad F = k \frac{(V_{dd} - V_t)^2}{V_{dd}}$$

α étant l'activité du circuit, C_l les capacités parasites V_{dd} la tension d'alimentation, F la fréquence de

fonctionnement, V_t la tension de seuil, S la pente sous le seuil (liée au temps de montée), et k une constante qui dépend du circuit [2].

Au niveau d'abstraction système, l'énergie est généralement réduite en mettant le processeur en veille ou en diminuant la tension d'alimentation et en ralentissant sa fréquence de fonctionnement. Les techniques DVS et AVS (Adaptive Voltage Scaling) appartiennent à la dernière catégorie et permettent des gains de 36% et 64% par rapport à la première [1].

L'intérêt de la minimisation de l'énergie consommée dans les systèmes n'est plus à démontrer. Il est toutefois également important de maîtriser les fluctuations rapides de puissance, et plus généralement, de limiter les pics de puissance pour différentes raisons [3]:

- Amélioration du bon fonctionnement des batteries : la conversion électrochimique qui génère les tensions d'alimentation est perturbée par les fluctuations de consommation, la durée de vie des batteries diminue et le rendement des batteries est moins bon.
- Réduction du bruit d'alimentation généré par les fluctuations de consommation : ce bruit génère des chutes de tensions ($L di/dt$) dans les fils conducteurs, qui peuvent être préjudiciable au bon fonctionnement des circuits.
- Réduction de l'effet de couplage inductif et capacitif : les appels de courants provoquent des perturbation du spectre électromagnétique, qui dénaturent le rapport signal sur bruit.

D'importants travaux de recherche sur la réduction de la puissance ont été menés au niveau comportemental [6] et au niveau système [7]-[12]. Dans les méthodes au niveau système un premier ordonnancement est généralement réalisé à fréquence nominale, l'objectif premier étant de respecter les contraintes de temps. La gestion de la puissance est ensuite prise en charge de façon statique ou dynamique en calculant un facteur de ralentissement. A notre connaissance aucune des méthodes actuelles ne détermine un profil de puissance permettant d'adapter au mieux le système à un environnement contraint de consommation. Pour ce faire, nous adaptons des stratégies d'optimisation utilisées en « floorplanning » pour le placement-routage [4][5].

Le paragraphe 2 présente le cadre général de notre étude. La méthode d'estimation d'un profil de

puissance est donnée au paragraphe 3. La structure de données et l'heuristique de réduction des pics puissance sont développées au paragraphe 4.

2. Méthodologie

Nous proposons d'estimer le profil de puissance du système et de réduire par l'utilisation de la technique DVS les pics de puissance que doit fournir l'alimentation. Les deux méthodes que nous proposons sont réalisées à l'issue de l'étape de partitionnement logiciel/matériel (voir Fig. 1). Elles sont donc indépendantes de la façon dont est obtenu le partitionnement.

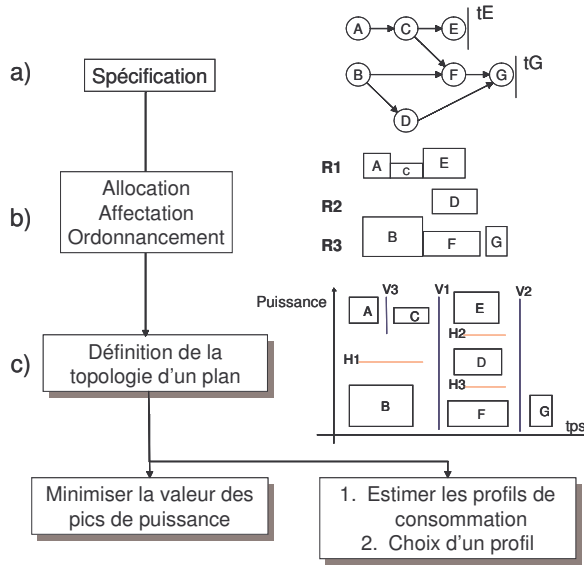


Fig. 1 Flot de conception

A ce niveau de la conception du système certaines tâches possèdent des degrés de liberté sur la localisation, leur forme et leur surface. On parle alors de tâches flexibles. Leur profil est donné par l'ensemble des couples (temps, puissance). Nous considérons un ensemble discret de valeurs. Dans la Fig. 2, les couples $A(1,3)$, $A(3,2)$ et $A(4,1)$ représentent les valeurs (temps, puissance) potentielles associées à une tâche flexible A.

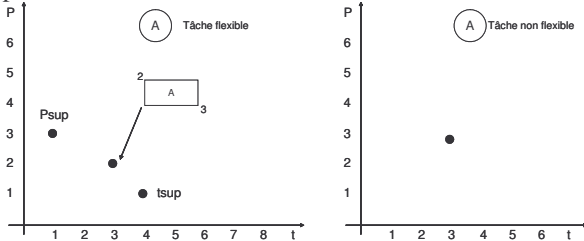


Fig. 2 Profil d'une tâche

Le temps et la puissance sont bornés respectivement par les valeurs $tsup$ et $Psup$. L'aire de chaque tâche, correspondant au produit de son temps d'exécution par la puissance, représente l'énergie dissipée. Certaines tâches peuvent être non flexibles. Le résultat du partitionnement définit la topologie d'un plan. Les axes horizontal et vertical sont

respectivement l'axe du temps et celui de la puissance. Le placement relatif des tâches sur le plan est obtenu par divisions récursives en deux parties par des segments de droites verticaux (V_i) ou horizontaux (H_i).

La première méthode détermine de manière constructive les profils de puissance du système basée sur cette notion de topologie du plan. La seconde méthode décongestionne de manière itérative les zones à forte consommation rendant ainsi plus uniforme la répartition de la puissance consommée.

3. Estimation des profils de puissance

La hiérarchie du partitionnement du plan est représentée par un arbre binaire (voir Fig. 3). Les feuilles sont composées des tâches fonctionnelles issues de la spécification. Un nœud V_i (resp. H_i) représente une division verticale (resp. horizontale).

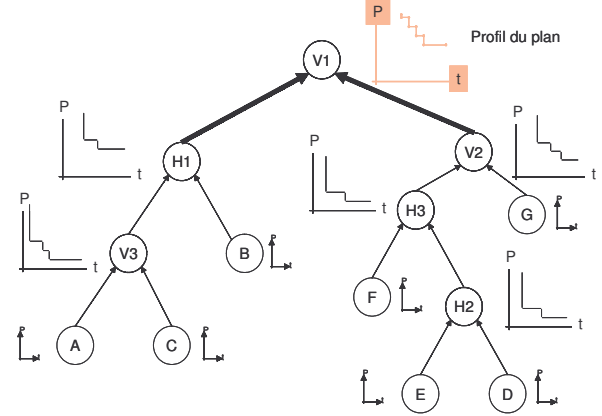


Fig. 3 Composition ascendante

L'arborescence ainsi obtenue est alors traversée de manière ascendante. A chaque niveau les profils sont obtenus par composition verticale ou horizontale des deux profils de niveau inférieur.

3.1 Composition ascendante

Pour une composition horizontale, à un couple (t, P) appartenant au profil d'une tâche T on associe le couple (t', P') du profil de l'autre tâche, T' , où t' est la plus grande valeur du temps possible pour T' satisfaisant $t' \leq t$. Pour une composition verticale, les rôles de t et de P sont inversés.

Graphiquement, la composition horizontale est obtenue en sommant les hauteurs (représentant la puissance) et en prenant la valeur maximale des largeurs (représentant le temps d'exécution). A titre d'exemple, pour la topologie de la figure 4a, où une division horizontale sépare les tâches E et D, la figure 4b illustre la composition horizontale des tâches $D(2,2)$ et $E(1,3)$ en $H(2,5)$. La figure 4c donne le résultat pour l'ensemble des valeurs.

A l'issue de la phase ascendante, le profil associé à la racine de la Fig. 3 est obtenu. Le nombre de couples (temps, puissance) est inférieur ou égal à la somme des nombres des couples (temps, puissance) de chacune des feuilles (tâches). De plus, le temps de

calcul est proportionnel au nombre de couples (fusion de listes triées).

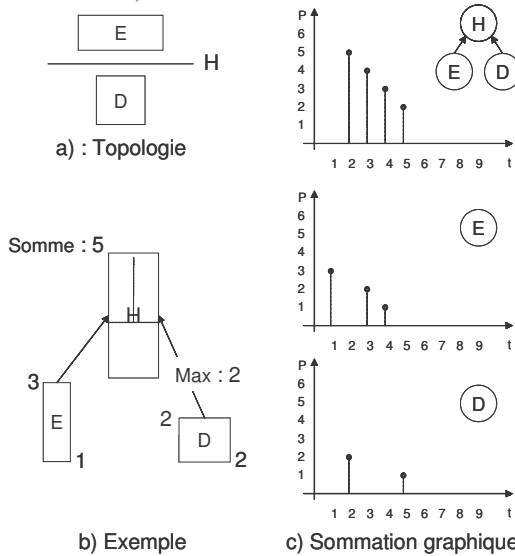


Fig. 4 : Exemple de composition ascendante

Nous passons alors à la phase descendante.

3.2 Composition descendante

L'utilisateur définit au sommet de l'arborescence un profil particulier que doit vérifier le système (voir Fig. 5).

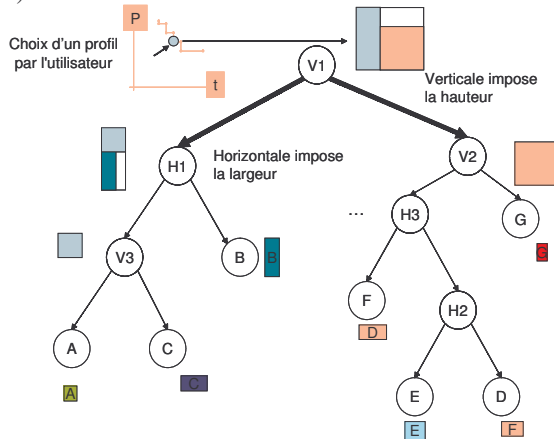


Fig. 5 : Composition descendante

Cette information est alors imposée de manière descendante à l'ensemble de l'arborescence. Une coupure verticale (resp. horizontale) détermine la hauteur (resp. la largeur) pour le niveau inférieur. Par exemple sur la composition descendante de la figure 6. lorsque l'utilisateur choisit un profil de puissance (8,2), la composition descendante impose la puissance de valeur égale à 2 aux nœuds A et C. Ainsi, pour un profil $V(8,2)$ sera propagé un $A(3,2)$ et $C(2,2)$. Il est à noter que la solution de composition descendante n'est pas unique. Par exemple, pour le profil $V(8,2)$ peuvent être propagés les profils donnés dans la table 1.

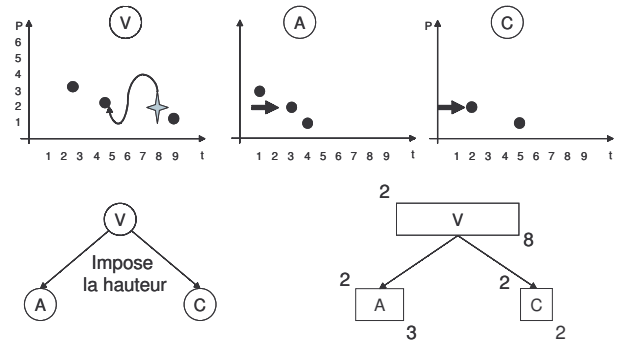


Fig. 6 : Exemple de composition descendante

Nous limitons la recherche à une seule solution. Il est intéressant de propager l'ensemble des solutions lorsque l'on veut prendre en compte d'autres critères comme les écarts de puissance.

V	A	C
(8,2)	(3,2)	(5,1)
(8,2)	(3,2)	(2,2)
(8,2)	(4,1)	(2,2)

Table 1 : Propagation des profils

Il se peut qu'une telle recherche ne trouve aucune solution satisfaisant les contraintes de temps et de puissance. Cela peut-être dû à la topologie utilisée. Ainsi sur l'exemple de la figure 1c, le temps est calculé sous l'hypothèse que G ne s'exécute pas en concurrence avec E . Cependant, la division verticale $V2$ peut être réduite de sorte à séparer seulement les tâches D et F de G , et ainsi permettre de réduire encore plus la puissance de E tout en garantissant que la contrainte de temps soit satisfaite.

4. Minimisation des pics de puissance

Intuitivement le problème de la réduction des pics de puissance ramené à un problème de dimensionnement d'un rectangle est simple. Il faut en augmenter la largeur dans la limite de la contrainte de temps fixée afin d'en réduire au maximum la hauteur (puissance). Nous proposons un algorithme d'optimisation qui réduit itérativement la puissance maximale (hauteur du rectangle) en redimensionnant à chaque étape la tâche flexible sélectionnée. La sélection de la tâche flexible est basée sur la notion de marge.

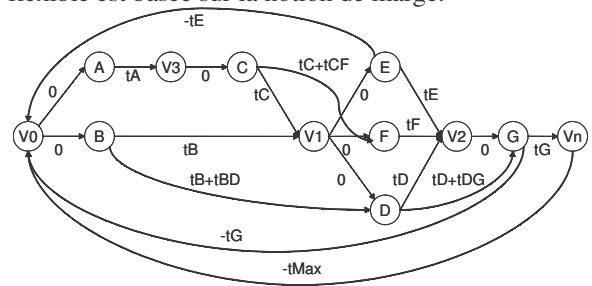


Fig. 7 Graphe en temps

La hiérarchie du partitionnement du plan est représentée par deux graphes série-parallèle.

Pour construire le graphe de temps, on associe :

- à chaque tâche T un nœud du graphe, noté T
- à chaque segment de droite vertical V_i un nœud du graphe, noté V_i .
- un arc entre le nœud T et le nœud V_i si la tâche T est immédiatement à gauche de V_i
- un arc entre le nœud V_i et le nœud T si le segment V_i est immédiatement à droite de T .

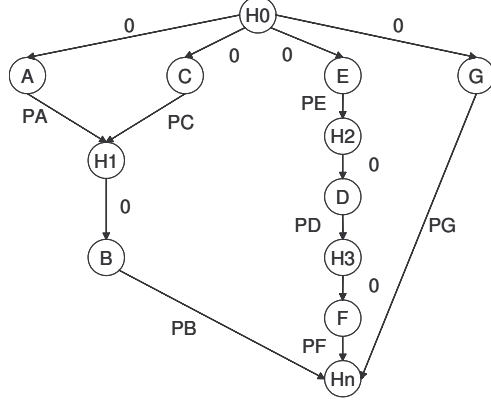


Fig. 8 : Graphe en puissance

Un nœud source V_0 et un nœud puits V_n sont associés respectivement aux limites gauche et droite du plan. Le poids des arcs représente le temps d'exécution des tâches.

Des arcs supplémentaires permettent de prendre en compte les coûts de communications associés aux transferts de données entre différents composants matériels.

Par analogie avec le graphe de temps, nous pouvons construire le graphe de puissance. Le poids des arcs représente la puissance des tâches.

Définitions

La largeur L du rectangle et sa hauteur H sont données par le plus long chemin respectivement dans le graphe de temps et dans celui de puissance.

On appelle marge horizontale ou marge en temps

$Mt(T)$ d'une tâche T la quantité suivante :

$Mt(T) = L - (t_{V_0 \rightarrow T} + t_{T \rightarrow V_n})$ avec $t_{V_0 \rightarrow T}$ (resp. $t_{T \rightarrow V_n}$) le plus long chemin des nœuds V_0 à T (resp. T à V_n).

On appelle marge horizontale de déformation ou marge de déformation en temps $Mdt(T)$ d'une tâche T la quantité suivante : $Mdt(T) = \min(Mt(T), t_{sup}(T) - t(T))$.

Cette marge permet de prendre en compte les limites de déformation en temps imposées par la courbe de déformation de chaque tâche flexible.

De la même façon on définit les marges verticales ou marge en puissance la quantité suivante :

$MP(T) = H - (P_{H_0 \rightarrow T} + P_{T \rightarrow H_n})$

et les marges de déformation :

$MdP(T) = \min(MP(T), P_{sup}(T) - P(T))$.

Algorithme

La boucle de l'algorithme est la suivante :

Pour chaque tâche flexible T

Calcul des marges $Mt(T)$ et $MP(T)$

Pour les tâches telles que $MP(T) = 0$

Calcul de $Mdt(T)$

Sélection du Max des $Mdt(T)$

Redimensionnement de la tâche sélectionnée.

Pour éviter des changements de dimension trop importants, on limite la déformation à une quantité égale à $kMdt(T)$ avec $0 < k < 1$. La nouvelle durée est donnée par $t'(T) = t(T) + kMdt(T)$.

L'arrêt de la boucle se fait lorsque les chemins critiques ne contiennent que des tâches flexibles dont la déformation est maximale.

La complexité d'une itération de cet algorithme est en $O(N)$, où N est le nombre de tâches [4].

5. Conclusion

Dans cet article nous avons présenté le cadre général d'une méthodologie permettant d'estimer le profil de puissance d'un système multiprocesseur et de réduire par l'utilisation de la technique DVS les pics de puissance demandés à l'alimentation. Les deux méthodes que nous avons proposées sont réalisées à l'issue de l'étape de partitionnement logiciel/matériel.

Références :

- [1] Reducing your system power management. European ARM Technology Congress.
- [2] Burd, T., Brodersen, R. W. 1995. Energy Efficient CMOS Microprocessor Design. In Proceedings of the 28th Hawaii International Conference on System Sciences. 288-297.
- [3] Singh, D., Rabaey, J. M., Pedram, M., Catthor, F., Rajgopal, S., Sehgal, N., and Mozden, T. J. 1995. Power Conscious CAD Tools and Methodologies: A Perspective. Proceedings of the IEEE 83, 4(Apr), 570-594.
- [4] W-W. Dai, B. Eschermann, E.S. Kuh and M. Pedram, "Hierarchical placement and floor-planning for BEAR," IEEE Trans. on Computer Aided Design, Vol. 8, No. 12 (1989), pp. 1335-1349.
- [5] R. Otten, "Efficient floorplan optimization," in Proc. of Int. Conf. on Comp. Design, Silver Spring, MD, 1983, pp. 499-502, IEEE Comp. Soc. Press.
- [6] S. P. Mohanty, N. Ranganathan, S. K. Chappidi: ILP Models for Energy and Transient Power Minimization During Behavioral Synthesis. VLSI Design 2004: 745-748.
- [7] M. T. Schmitz, Al-Hashimi, B. M. and Eles, P. (2004) System-Level Design Techniques for Energy-Efficient Embedded Systems. Kluwer Academic Publishers.

- [8] D. Zhu, R. Melhem, and B. Childers. Scheduling with Dynamic Voltage/Speed Adjustment Using Slack Reclamation in Multi-Processor Real Time Systems, RTSS'01 (Real-Time Systems Symposium), London, England, Dec 2001.
- [9] R. Mishra, N. Rastogi, D. Zhu, D. Mosse, R. Melhem, "Energy Aware Scheduling for Distributed Real-Time Systems", Proc. of the International Parallel and Distributed Processing Symposium (IPDPS'03), Nice, France (April 2003).
- [10] D. Zhu, N. AbouGhazaleh, D. Mosse and R. Melhem, "Power Aware Scheduling for AND/OR Graphs in Multi-Processor Real-Time Systems", Proc. of the International Conference on Parallel Processing (ICPP), Vancouver, B.C. (Aug. 2002).
- [11] J. Luo and N. K. Jha. "Power-conscious joint scheduling of periodic task graphs and aperiodic tasks in distributed real time embedded systems." In Proc. of International Conference on Computer Aided Design (ICCAD), San Jose, CA, Nov. 2000.
- [12] J. Liu, Pai H. Chou, N. Bagherzadeh, F. Kurdahi, "Power-Aware Scheduling under Time Constraints for Mission-Critical Embedded Systems", Proceedings of the 38th Design Automation Conference, pages 840-845, Las Vegas, NV, USA, June 2001.