

Estimation de la consommation au niveau architectural pour les circuits dédiés au traitement du signal

David SAILLE, Matthieu DENOVAL, Olivier SENTIEYS

LASTI - ENSSAT - Université de Rennes I,
6 rue de Kérampont,
BP 447, 22305 Lannion, France
saille@enssat.fr, denoval@enssat.fr, sentieys@enssat.fr

Résumé – La consommation des circuits intégrés, en particulier dans le domaine du traitement de signal embarqué, est devenue une contrainte de conception primordiale. Il est donc nécessaire de mettre en œuvre des outils de conception haut-niveau spécifiques au problème de la basse consommation, aussi bien pour les applications portables que pour les applications hautes performances. Dans cet article nous présentons notre outil d'estimation de la consommation au niveau architectural, **PowerCheck**, intégré dans l'environnement **BSS** (Breizh Synthesis System, <http://archi.enssat.fr/bss>).

PowerCheck est situé en aval de la synthèse architecturale et estime la consommation de l'application à partir des caractéristiques de l'architecture et de l'assignation temporelle de l'algorithme sur ladite architecture. L'intérêt principal de notre approche est la prise en compte des propriétés du signal, à savoir les corrélations spatiale et temporelle, et de l'architecture complète.

Mots clés : Consommation - Estimation haut-niveau - Propriétés du signal.

Abstract – With the emergence of embedded processing systems, the power dissipation of very large scale integrated circuits is becoming a critical concern. Therefore power requirements are driving a new breed of computer-aided design tools. This paper addresses the problem of modeling power dissipation at a high level of abstraction. A modeling approach is presented that captures the dependence of the circuit modules power dissipation on their activities. We also introduce our RT level estimation tool, **PowerCheck**, part of the **BSS** framework (Breizh Synthesis System, <http://archi.enssat.fr/bss>).

PowerCheck takes place after the RT synthesis, then it profits by the architecture and its use. The major interest of our approach is that we take into account the signal properties in our power dissipation estimates of complete architecture.

Key words : Power dissipation - High-level estimation - Signal properties.

1 Introduction

Très tôt dans le processus de conception il est utile d'avoir des estimations de ce que seront la surface, la consommation et le timing du circuit que l'on conçoit. Celles-ci sont généralement obtenues par des analyses simples sur des représentations graphiques du flot de données, ou de contrôle ou encore sur des esquisses simplifiées d'architecture. Ces analyses requièrent l'extraction d'un certain nombre d'informations cruciales comme les caractéristiques de surface, de consommation, de comportement temporel des modules de l'architecture [1]. On peut alors obtenir une estimation de la consommation par exemple en multipliant le nombre estimé d'accès à un module par la puissance dissipée par accès à ce module [1].

Cependant, la précision de ce type d'estimation risque de ne pas être suffisante pour pouvoir faire des choix entre différents types d'architectures. Cette imprécision provient généralement des modèles utilisés pour caractériser les différents modules de l'architecture souvent trop simples ou généraux. D'autre part on ne peut pas envisager de simuler chaque module à bas-niveau à cause du temps que cela implique. Pour permettre une estimation précise et efficace de la consommation à haut-niveau il faut donc un certain nombre de pré-requis. La nécessité d'une bibliothèque de modules caractérisés en consommation n'est pas à démon-

trer. D'autre part, pour pouvoir à tout moment compléter cette bibliothèque il faut un certain nombre de techniques de modélisation pour les différents types de modules. Finalement il faut une base de travail complète avec à la fois la représentation du circuit étudié et son fonctionnement. Notre article présente l'outil **PowerCheck**. Notre outil constitue un module d'estimation de la consommation haut-niveau au sein de l'environnement de conception [2] développé par notre laboratoire. Il bénéficie de ce fait de l'architecture des circuits et de leur fonctionnement. Sur cette base de travail, il nous restait à déterminer ou à choisir des modèles pour la consommation des différents modules ou composants. Après un rapide retour sur les notions de consommation et les méthodes d'estimation de cette dernière, nous présentons les modèles retenus pour les différents modules. Plus en adéquation avec la réalité du traitement numérique du signal, leur caractéristique principale commune est la prise en compte des propriétés des signaux (corrélation).

2 Consommation et estimation

Afin de permettre d'agir très tôt dans le cycle de conception, il est indispensable de posséder des outils permettant d'estimer à un haut niveau d'abstraction la consom-

mation du système (ASIC ou DSP). Comme le montre la formule fondamentale de la consommation $P = \alpha \cdot C \cdot V_{DD}^2 \cdot f$, a contrario de la surface, la puissance moyenne dissipée par un élément ne peut pas être estimée indépendamment de son usage (terme α). On a besoin de considérer l'activité du circuit.

Un tel calcul prévisionnel de la consommation peut être effectué selon deux approches (voir figure 1) :

- l'estimation statistique nécessite une simulation du circuit par un vaste jeu représentatif des vecteurs d'entrées pour la détermination de la consommation moyenne. Les difficultés majeures d'une telle approche se situent sur les mots "vaste" et "représentatif". Plus le concepteur simule à un niveau bas, plus le calcul est précis et plus le temps de simulation est long. Ce type de méthode statistique donne des résultats plus précis que le second type d'approche.

- l'estimation probabiliste nécessite, quant à elle, une seule simulation. Elle s'appuie pour la logique combinatoire, sur la notion de densité de probabilité, hélas très sensible aux délais des portes, et aux corrélations spatiales et temporelles des données.

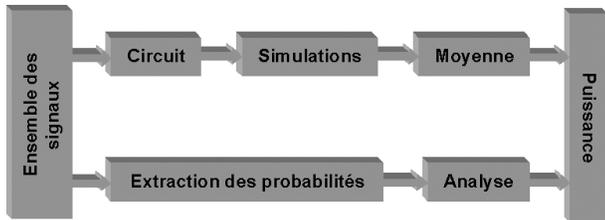


FIG. 1 - Deux approches pour l'estimation

Au niveau architectural, on retrouve les deux mêmes étapes qu'au niveau logique pour l'estimation de la consommation. Celles-ci sont : la caractérisation des composants des bibliothèques en terme de consommation et l'estimation de la consommation moyenne du circuit en combinant les informations sur le comportement du circuit avec les caractéristiques des composants stockées en bibliothèque. Cependant, à ce niveau, les composants sont généralement caractérisés en consommation par des simulations utilisant un *bruit blanc* [3], car on ne sait pas quel type de donnée sera effectivement utilisé. Ceci présente l'inconvénient de négliger les effets de la corrélation des données. Il en résultera des estimations moins précises qu'au niveau logique, du fait de la perte d'une partie d'un élément essentiel de la consommation : l'activité du signal.

Voilà pourquoi dans la méthode d'estimation de la consommation des architectures que nous avons développée, nous avons choisi ou déterminé des modèles de consommation des composants dépendant de l'activité du signal.

Outils d'estimation de la consommation :

les estimations peuvent s'effectuer à tous les niveaux d'abstraction permettant de décrire un circuit. Les outils industriels se sont focalisés sur les niveaux logiques et transistors (figure 2). Seul Watt/Watcher Architect [4] permet actuellement de prédire la consommation au niveau RT.

Nous avons développé une méthode d'estimation de la consommation des architectures synthétisées par nos outils. La prise en compte de l'ensemble des composants

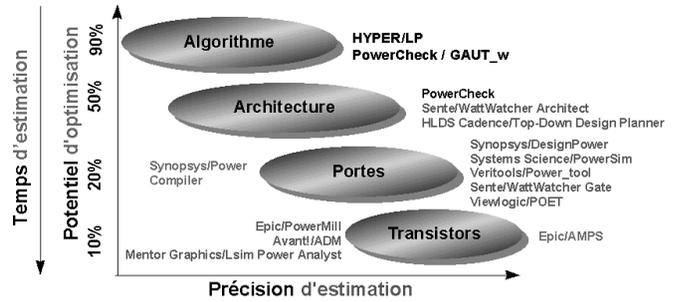


FIG. 2 - Outils d'estimation de la consommation [DAC97]

de l'architecture : opérateurs arithmétiques, arbre d'horloge, composants d'interconnexion, bus, registres, unités de contrôle, mémoires et générateurs associés, donne une valeur très précise de la dissipation de puissance. L'outil **PowerCheck** permet également de prendre en compte divers paramètres technologiques ainsi que des probabilités associées aux signaux (bruit blanc, signal réel, DBT, ...) (voir figures 4 et 7). Nous avons en particulier développé des méthodes originales pour les parties traitement, contrôle et mémoire.

3 Modèles adoptés

L'idée de base est de dissocier les parties actives des inactives suivant les propriétés du signal pour un même module. Prenons le cas d'un registre. Il est composé de bascules qui suivant la valeur présentée à leurs entrées vont voir leurs sorties commuter ou non. Quand on sait que la consommation dynamique d'une bascule est directement liée à la commutation de ses sorties, on voit clairement le lien qui existe entre les propriétés du signal et la consommation d'un registre.

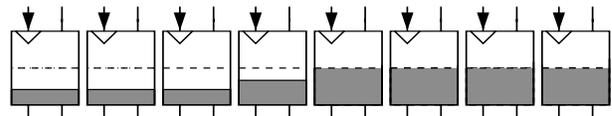


FIG. 3 - Registre avec activité des bascules

Ainsi la consommation associée à un registre n'est pas pour nous une consommation moyenne pour un signal aléatoire, mais la consommation directement liée au signal traversant ce registre, par l'intervention d'un terme représentant l'activité des bits du registre, introduisant ainsi le modèle DBT du signal dans notre formule. En effet ce modèle de signal défini par Landman [5] donne pour chaque bit du signal sa probabilité de transition, voir figure 4. Donc l'expression de la consommation de registres sous forme mathématique est :

$$Conso_{reg} = dim_{reg} \cdot C_1 + \sum_{bascules}^{dim_{reg}} P_{trans_i} \cdot C_2$$

avec :

- dim_{reg} : la dimension du registre
- C_1 : consommation statique d'une bascule
- C_2 : consommation dynamique d'une bascule
- P_{trans_i} : probabilité de transition du i^{ieme} bit

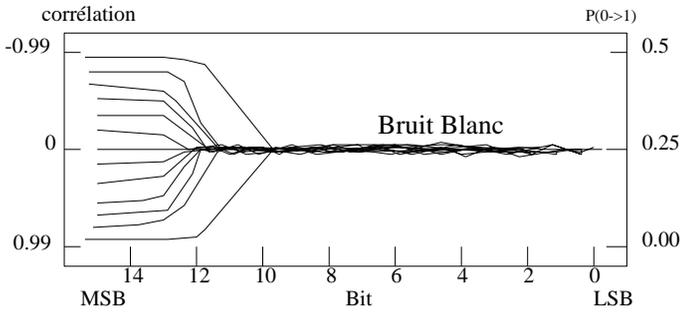


FIG. 4 - *Modèle DBT, activité de transition en fonction des bits*

Les probabilités de transition de chaque bascule sont représentées sur la figure 3 par les zones grisées. Chaque bascule possède une probabilité de transition spécifique selon le modèle DBT appliqué. À l'intérieur du registre (figure 3), on distingue une partie aléatoire à droite (quatre bascules), et à gauche des bits de plus en plus corrélés.

La démarche reste la même pour les chemins de données, bus et multiplexeurs, pour lesquels d'une manière similaire on peut exprimer la consommation en fonction des propriétés probabilistes des données. Un autre intérêt de cette méthode, outre la précision, est la généralité ; en effet si la dimension des données varie, il n'est pas nécessaire de recharacteriser en consommation les composants pour les stocker dans un bibliothèque puisqu'il suffit de modifier la variable représentant la dimension des composants.

Par contre, pour les opérateurs (additionneurs, multiplieurs...) la démarche n'est pas si simple, puisqu'on ne peut pas aisément déterminer les zones actives ou inactives suivant les propriétés des données. Dans ce cas, nous sommes donc contraints à reprendre la démarche habituelle de caractérisation en consommation des composants et créer des tables contenant les valeurs de consommation des opérateurs pour différents types de signaux probabilistes. Cependant une analyse de ces tables a permis de définir des formules pour obtenir directement la puissance dissipée par les opérateurs en fonction de leur utilisation (typiquement : $P = \rho_1 \cdot \rho_2 \cdot \lambda_0 + \rho_1 \cdot \lambda_1 + \rho_2 \cdot \lambda_2 + \Lambda$), voir figure 5.

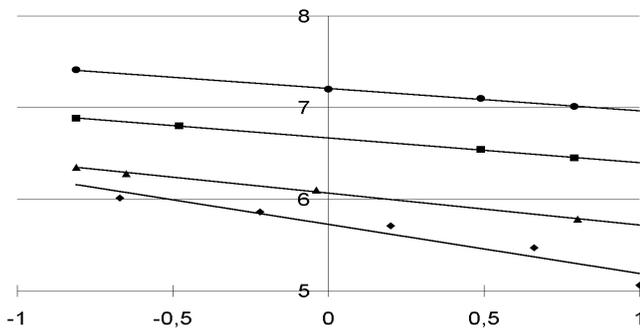


FIG. 5 - *Consommation d'un multiplieur en fonction de la corrélation de ses entrées*

Pour la partie contrôle, nous avons développé pour le registre d'états de la FSM, une méthode d'estimation de la consommation par résolution des équations de Chapman-

Kolmogorov délivrant les probabilités de transitions entre états [6]. En ce qui concerne le bloc logique de la FSM nous avons développé une méthode originale, applicable à tout type de bloc logique, permettant d'obtenir la puissance dissipée par un bloc à partir d'un ensemble de critères que les expérimentations et les simulations nous ont soulignés comme primordiaux. Ces critères sont de deux ordres, les uns liés à la taille du bloc logique (nombre de bits d'entrée, nombre de bits de sortie) et les autres liés à l'activité du bloc (activité des entrées, activité des sorties). À ceux-ci, nous avons ajouté un critère de dépendance entre les entrées et les sorties. Nous avons ainsi déterminé une méthodologie en 3 étapes. Premièrement, on détermine une matrice de dépendance entre les entrées et les sorties, en introduisant par ailleurs des notions de chemins complets ou évanescents dans le bloc logique. Deuxièmement, on estime le coût de ces chemins. Finalement, on procède à une analyse transition par transition. Cette méthode permet ainsi une estimation de la consommation du bloc logique sans connaissance a priori de la future structure de celui-ci après l'étape de synthèse logique.

Cas de la consommation des mémoires :

l'estimation de la consommation d'une mémoire est issue d'informations qui sont $f_{réelle}$ donnée par le nombre d'accès à cette dernière sur un certain laps de temps ($f_{réelle} = \alpha \cdot f$) et l'énergie mise en jeu lors d'un accès ($C \cdot V_{dd}^2$). Il est donc indispensable de connaître les valeurs des capacités internes aux différentes entités (wordline, bitline, drivers...) constituant une mémoire. Des modèles développés par S. Wilton et N. Jouppi [7] ou encore S. Schmidt [8] permettent d'accéder à ces valeurs.

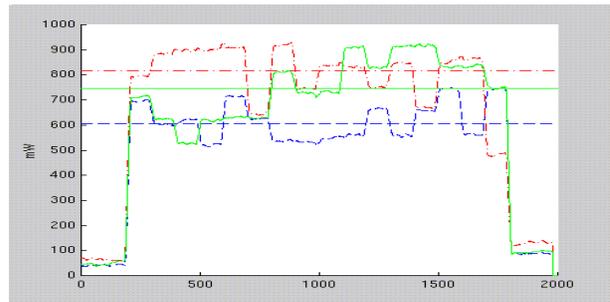


FIG. 6 - *Mesures de consommation sur une SRAM 8K*8. signal aléatoire (.), signaux corrélés (-. et -)*

À partir de ces informations et en intégrant différents codages d'adresses et divers modèles de signaux de données, nous estimons la consommation des mémoires pour des applications de traitement du signal. Ainsi, tout d'abord, l'énergie mise en jeu lors d'un accès est déterminée pour les différents éléments de la mémoire. Par sommation, on obtient l'énergie par accès $E_{accès}$. Ensuite, connaissant le nombre d'accès N et la durée de l'application T_{appl} (fournis par l'outil de synthèse **BSS**), nous calculons la puissance consommée par la formule $P = N \cdot E_{accès} / T_{appl}$. L'innovation de la méthode se situe dans le fait d'intégrer des caractéristiques spécifiques aux signaux d'adresses et de données au moment du calcul de $E_{accès}$. La figure 6 montre bien que la consommation dépend fortement des

signaux manipulés. Des valeurs de consommation plus précises seront donc trouvées en utilisant lors des estimations des signaux représentatifs des signaux réels.

Il faut remarquer que les spécificités des signaux n'influent pas sur la totalité des modules constituant une mémoire, du fait de la logique complémentaire utilisée. Ainsi les signaux devront être pris en compte lors de l'estimation de la consommation du décodage des adresses, de l'écriture dans les cellules et également des buffers de sorties.

Propagation du signal dans l'architecture :

Le fait d'utiliser les propriétés du signal pour nos estimations de la consommation des modules nous contraint à propager les caractéristiques du signal à travers l'architecture. Alors que la propagation du modèle DBT à travers les chemins de données (registres, bus) est simple, celle à travers les opérateurs est moins évidente. Pour ces derniers nous avons déterminé des formules mathématiques donnant le modèle DBT du signal de sortie d'un opérateur en fonction des modèles DBT des signaux de ses entrées.

L'outil PowerCheck :

Il s'agit d'un outil probabiliste d'estimation de la consommation au niveau architectural (voir synoptique, figure 7). Il admet en entrée, soit directement un modèle probabiliste des signaux, soit des fichiers de vecteurs test dont seront extraits les modèles probabilistes du signal utiles à l'outil. De plus un fichier de paramètres permet de caractériser la technologie employée pour le circuit.

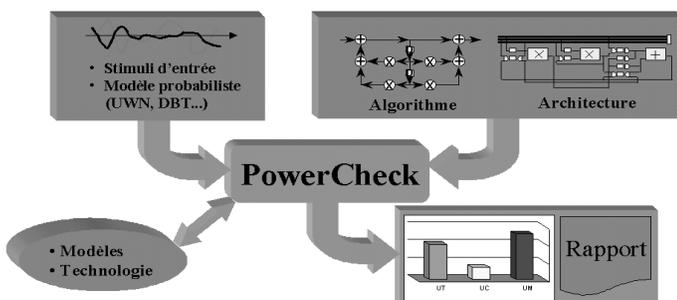


FIG. 7 - Synoptique de PowerCheck

L'outil s'appuie sur une architecture fournie par BSS. Les résultats sont délivrés sous forme de fichier texte ou d'histogrammes. Ces derniers donnent les valeurs relatives des consommations des différentes catégories de modules (opérateurs, registres, chemin de données, mémoire) ainsi que la variation de consommation cycle par cycle au fur et à mesure du déroulement de l'algorithme de traitement du signal sur l'architecture.

4 Conclusion

Nous avons exposé des méthodologies d'estimation de la consommation pour la partie traitement et mémoire à partir d'une architecture synthétisée par notre plateforme BSS. Ces méthodes sont intégrées au sein de l'outil

PowerCheck. L'innovation apparaît dans le fait que les caractéristiques des signaux sont prises en compte pour l'estimation. Les résultats obtenus jusqu'ici montrent bien que la consommation est très sensible aux données manipulées.

L'outil **PowerCheck** s'inscrira ensuite dans la conception d'architectures optimisées en consommation. En effet, à l'intérieur du flot de conception de BSS, un rebouclage permettra d'explorer différentes implémentations utilisant des techniques d'optimisation tels que le "gated clock", l'application de différents codages et une architecture différente de la partie mémoire (division d'une mémoire en plusieurs modules, hiérarchisation). Ces optimisations demanderont ensuite une réestimation de la consommation de toutes les parties composant l'architecture resynthétisée par BSS. D'autre part pour rendre les résultats plus précis, une estimation de la consommation des interconnexions est prévue grâce à une estimation rapide du floorplan au niveau architectural.

Il sera bientôt possible d'accéder à **PowerCheck** par le biais du framework BSS (<http://archi.enssat.fr/bss>). D'autre part des résultats seront fournis avec la version finale de l'article.

Par exemple, pour une application *FIR 16*, en considérant les coefficients aléatoires, pour différentes valeurs de la corrélation des échantillons d'entrée, on obtient des valeurs différentes de consommation.

TAB. 1 - Consommation d'une application FIR 16 en fonction de la corrélation des échantillons

FIR 16	corrélation des échantillons	
	-0.1	0.99
puissance filaire	38.75 mW	33.7 mW
puissance des cellules	72 mW	60.07 mW
puissance totale	110.75 mW	93.8 mW

Références

- [1] Raghunathan, Jha, and Dey. *High-level power analysis and optimization*. Kluwer Academic Publishers, 1998.
- [2] J-P. Diguët, O. Sentieys, J-L. Philippe, and E. Martin. Modèle unifié pour la conception conjointe logiciel-matériel. *Traitement du signal, Adéquation Algorithme Architecture*, 14(6):579-589, 1994.
- [3] S. Gailhard, N. Julien, and E. Martin. Intégration de méthode d'optimisation faible consommation dans l'outil de synthèse architecturale gaut-w. In *AAA '97, Saclay, 29-30*, January 1998.
- [4] Jerry Frenkil. Tools and methodologies for low-power design. *Design Automation Conference*, 1997.
- [5] Paul E. Landman and Jan M. Rabaey. Black-box capacitance models for architectural power analysis. In *Proceeding of 1994 International Workshop on Low-Power Design*, editor, *Black-Box Capacitance Models for Architectural Power Analysis*, avril 1994.
- [6] Fornaciari, Gubiani, Sciuto, and Silvano. A vhdl-based approach for power estimation of embedded systems. *Journal of systems architecture*, 44:37-61, 1997.
- [7] Steven J. E. Wilton and Norman P. Jouppi. An enhanced access and cycle time model for on-chip caches. Technical report, Digital-Western Research Laboratory, July 1994.
- [8] E. Schmidt, L. Kruse, G. Jochens, E. Huijbregts, W. Nieuweboer, E. Seelen and W. Nebel. Power consumption of on-chip roms: Analysis and modeling. *PATMOS Workshop*, 1998.