HAL
open science

Interconnexions et consommation: où en sommes nous?

Antoine Courtay, Olivier Sentieys, Nathalie Julien

To cite this version:

Antoine Courtay, Olivier Sentieys, Nathalie Julien. Interconnexions et consommation: où en sommes nous?. MajecSTIC, Nov 2006, LORIENT, France. pp.Interconnexions et consommation: où en sommes nous?. hal-00169249

HAL Id: hal-00169249

https://hal.science/hal-00169249

Submitted on 3 Sep 2007

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.

Interconnexions et consommation: où en sommes nous?

Antoine Courtay[1,2], **Olivier Sentieys**[1], **Nathalie Julien**[2]
[1]IRISA - Université de Rennes 1 (ENSSAT)
6, rue de Kerampont 22300 Lannion, France
[2]LESTER - Université de Bretagne Sud
rue saint Maudé 56321 Lorient, France
antoine.courtay@univ-ubs.fr - sentieys@irisa.fr - nathalie.julien@univ-ubs.fr

Résumé : Cet article se propose d'aborder ce qu'il en est de la consommation des interconnexions dans les systèmes sur puce (SOC : Sytem On Chip) à l'heure actuelle. L'efficacité des différentes méthodes qui visent à réduire la consommation des interconnexions et leur influence en termes d'activité, de vitesse et de surface seront vues de manière détaillée. Les expérimentations nous ont permis de mettre au point un modèle de consommation pour les bus. A partir de ce modèle, nous avons développé un outil d'estimation rapide et précis en termes de surface, de vitesse de transfert et de consommation (instantanée, moyenne et maximale) sur le bus. Cet outil permet de tester rapidement les différentes méthodes et de conclure sur leur efficacité.
Mots-clés : Réseaux sur puce (NOC : Network On Chip), bus, consommation, codage, $crosstalk$, activité, surface, vitesse.

1 INTRODUCTION

Dans les technologies $CMOS$ (Complementary Metal Oxide Semi-conductor) actuelles la part de consommation due aux interconnexions peut représenter jusqu'à 50% de la consommation totale ainsi que de la surface occupée par le circuit [Magen, 2004]. Aujourd'hui, les applications portables possèdent de plus en plus de fonctionnalités, sont de plus en plus complexes et demandent de plus en plus de mémoire. Ces accès aux mémoires se font par l'intermédiaire des systèmes d'interconnexion, le trafic des données sur les bus est donc de plus en plus important.

Les prévisions de l'ITRS (International Technology Roadmap for Semiconductors) (tableau 1) montrent une diminution de la finesse de gravure des transistors ainsi qu'une augmentation de la surface des puces. Le nombre de transistors subit aussi une augmentation très importante qui va de pair avec une forte augmentation de la puissance consommée. L'évolution des dimensions des transistors et des fils se traduit par une évolution du comportement du circuit tout particulièrement au niveau temporel. Ainsi, le délai d'un fil devient supérieur au délai de commutation d'une porte [Ho, 2001]. Cette augmentation du délai est due en partie à l'augmentation de la résistance du fil causée par la diminution de sa section ainsi qu'aux multiples phénomènes de couplage capacitifs, couplages qui contribuent également à faire augmenter la consommation.

Année	2003	2005	2007	2010
Technologie (μm)	0.13	0.1	0.07	0.05
Fréquence (*MHz*)	2100	3500	6000	10000
Densité (x10^6)	77	202	520	1350
Surface (mm^2)	430	520	620	750
Puissance (*W*)	130	160	170	175
Alimentation (*V*)	1.2	0.9	0.6	0.5

TAB. 1 – Evolutions technologiques tirées de [ITRS04].

Nous allons voir dans une première partie comment modéliser un bus en y incluant tous les phénomènes résistifs et capacitifs. Dans une seconde partie nous verrons les facteurs influant sur la consommation ainsi que le modèle de consommation obtenu. La troisième partie présentera un état de l'art des techniques qui visent à réduire le délai et la consommation sur les bus. La section suivante présentera les résultats expérimentaux auxquels nous avons abouti grâce à notre outil d'estimation. La dernière section conclura cet article.

2 DU FIL AU BUS

2.1 Modélisation physique d'un fil

Nos travaux de recherche[1] sur la modélisation de la consommation des interconnexions se situent notamment au niveau physique. Pour cela nous avons effectué les différentes expérimentations avec un simulateur SPICE (ELDO V5.7) ce qui nous permet d'obtenir des résultats précis en termes de délai et de consommation.

Les grandeurs physiques qui permettent de modéliser le fil sont au nombre de trois :
- R, la résistance du fil, exprimée en Ohm [Ω] ;
- C, la capacité du fil, exprimée en Farad [F] ;
- L, l'inductance du fil, exprimée en Henry [H].

Ces grandeurs dépendent des caractéristiques du fil (sa composition métallique) ainsi que de ses dimensions. L'inductance n'a d'importance que pour les technologies très submicroniques $(45nm)$ et pour des fils extreme-

[1]Ces travaux sont cofinancés par la région Bretagne et l'Union européenne dans le cadre du programme Objectif 2 Bretagne 2000-2006

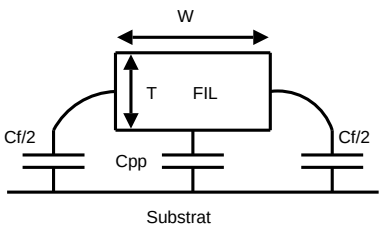

FIG. 1 – Détail des contributions des capacités d'un fil.

ments longs. Nous avons donc choisi de ne considérer seulement que le modèle RC pour le fil, dont la précision est tout à fait satisfaisante (moins de 5% d'erreur sur le délai notamment [Rabaey, 2003]). De plus, pour les fils de tailles raisonnables (quelques mm) propageant des signaux au sein d'un circuit, seules la capacité et la résistance sont significatives [Dally, 1998]
Les grandeurs élémentaires permettant de caractériser le fil que l'on trouve dans les *Design Kit* des constructeurs sont au nombre de trois :

- $R_{\square} = \frac{\rho}{T}$, résistance par carré, exprimée en Ohm par carré [$\Omega/\square$] avec ρ la résistivité du métal et T l'épaisseur du fil ;
- C_{sq}, capacité élémentaire de la face inférieure du fil par rapport au substrat, exprimée en Farad par mètre [F/m] ;
- C_e, capacité élémentaire des côtés du fil par rapport au substrat, exprimée en Farad par mètre [F/m].

A l'aide de ces trois grandeurs, il est possible de calculer la résistance ainsi que la capacité du fil en fonction de ses dimensions (sa longueur (L) et sa largeur (W) exprimées en mètre [m]). Notons également que C_{sq} et C_e dépendent de la hauteur (H) du fil par rapport au substrat et donc du niveau de métal utilisé.
La résistance globale du fil est donnée par l'équation suivante :

$$R = R_{\square}.\frac{L}{W} \tag{1}$$

La capacité globale du fil est en fait la somme de deux capacités : la capacité globale de la partie inférieure du fil par rapport au substrat *(parallel-plate capacitance)* notée C_{pp} et la capacité globale des cotés du fil par rapport au substrat *(fringing capacitance)* notée C_f. Ces capacités sont représentées sur la figure 1. Les capacités C_{pp} et C_f sont données par les équations suivantes :

$$C_{pp} = C_{sq}.W.L \tag{2}$$

$$C_f = 2.C_e.L \tag{3}$$

Le facteur 2 dans l'équation de C_f est destiné à inclure le fait que les deux côtés du fil contribuent à la capacité des bords. On obtient alors la capacité globale du fil par rapport au substrat qui vaut :

$$C_s = L.[C_{sq}.W + 2.C_e] \tag{4}$$

Il reste maintenant à choisir le modèle du fil, à savoir comment sont distribuées les valeurs de R et de C

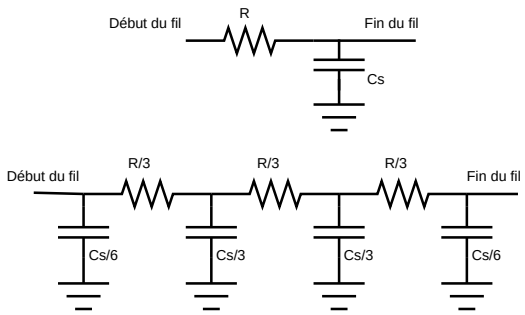

FIG. 2 – Modèle *lumped* suivi du modèle distribué $(\pi 3)$.

sur le fil afin de modéliser son comportement le plus précisément possible.
Le modèle *lumped* est un modèle simple d'interconnexion, il consiste à mettre bout à bout les valeurs de R et de C trouvées précédemment. Cependant, sa précision est beaucoup moins fiable notament en termes de délai [Rabaey, 2003] qu'un modèle où l'on distribue R et C. Par exemple, pour un modèle $\pi 3$ qui consiste à répartir la résistance du fil sur trois résistances et la capacité du fil sur quatre capacités, les valeurs obtenues en termes de délai ne sont éloignées des valeurs expérimentales que de 3 à 5%. On peut de cette manière fractionner les valeurs de R et de C indéfiniment. Nous avons retenu le modèle $\pi 3$ (figure 2) pour nos expérimentations du fait de sa simplicité et de sa précision.

2.2 Modélisation du bus

Comme il a été vu dans la section précédente, le fil peut se modéliser par une résistance R et une capacité C. Un bus n bits est simplement constitué de n fils de même longueur disposés parallèlement permettant de véhiculer des données entre deux blocs. Le fait d'utiliser plusieurs fils de cette manière fait apparaître un nouveau phénomène de couplage capacitif qui est le couplage entre fils. La capacité de couplage *(crosstalk)* entre deux fils adjacents dépend quant à elle de la surface en regard entre ces deux fils, et donc de l'épaisseur du fil (T), de la longueur (L) ainsi que de l'espacement (S) entre ceux-ci.

$$C_c = \varepsilon_0.\frac{T.L}{S} \text{ avec } \varepsilon_0 \text{ permitivité du SiO2.} \tag{5}$$

Lors de la transition des fils adjacents, il y a génération d'un bruit parasite sur le fil victime dû au couplage entre les fils. Le bruit dû au $crosstalk$ capacitif est relativement localisé. On modélise en général un système soumis au $crosstalk$ en négligeant les ordres supérieurs au premier : ainsi on ne considère que trois fils comme le montre la figure 3. Il sera expliqué plus en détail dans la partie suivante les phénomènes qui découlent du $crosstalk$ et les méthodes existantes pour en réduire les effets.
Le couplage entre les fils peut également se répartir sur les noeuds du modèle RC distribué défini précédemment. Nous obtenons alors sur la figure 4 le modèle complet du bus.

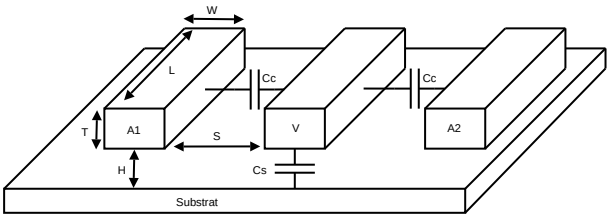

FIG. 3 – Un fil victime V soumis au couplage capacitif de ses deux agresseurs A1 et A2.

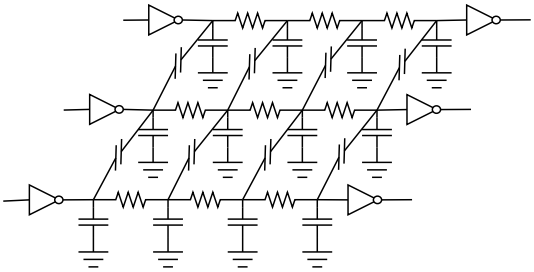

FIG. 4 – Modèle complet $\pi 3$ pour 3 fils avec couplage crosstalk.

3 ESTIMATION ET MODÈLE DE CONSOMMATION D'UN BUS

3.1 A quoi est due la consommation ?

La consommation de puissance pour un bus peut se représenter de la manière suivante :

$$P_{dynamique} = \sum_{i \in Nbit} \alpha_i . C_{L_i} . V_{dd} . V_{swing} . F \qquad (6)$$

La puissance statique peut être négligée puisque sa contribution dans la consommation des bus est très faible ; en effet les données qui circulent sur le bus changent très souvent d'état. C'est ce nombre important de transitions qui va faire que la consommation sur les bus est exclusivement de la consommation dynamique.

Dans l'équation de $P_{dynamique}$:

- N_{bit} représente le nombre de bits du bus considéré ;
- α_i représente l'activité du fil i ;
- C_{L_i} représente la capacité du fil i (capacité que nous allons détailler ci-dessous) ;
- V_{dd} représente la tension d'alimentation ;
- V_{swing} représente la tension d'excursion ;
- F représente la fréquence des transitions.

Comme nous l'avons vu dans le paragraphe précédent le couplage capacitif sur le bus se décompose en deux termes, le premier noté C_s qui est le couplage du fil par rapport au substrat et le second C_c qui est le couplage capacitif autrement appelé $crosstalk$ entre deux fils. Ce sont ces deux capacités qui vont composer C_L. Le problème est que C_L n'est pas une capacité constante et va varier en fonction des données qui transitent sur le bus.

Reprenons par exemple le modèle de la figure 3, si tous les fils ont le même niveau logique, la capacité de couplage C_c n'existe pas ; en revanche s'ils ont des niveaux différents, elle peut varier de une fois à quatre fois sa valeur selon le type des transitions sur les fils victime et agresseurs.

Les différents types de transition sont résumés dans le tableau 2 où g représente le facteur de délai introduit par le couplage et r représente le rapport entre C_c et C_s.

Dans ce tableau, $\uparrow$ représente une transition montante, $\downarrow$ représente une transition descendante et - signifie qu'il n'y a aucune transition sur le fil. Dans le meilleur cas, lorsque les trois fils changent de niveau dans la même direction, le délai introduit sur le fil victime est le délai sans $crosstalk$ (i.e. $g = 1$), mais le cycle de l'horloge doit être dimensionné exclusivement en prenant compte du délai pire cas (i.e. $g = 1 + 4.r$) afin d'assurer une transmission des données sans erreur. Nous avons expérimenté les différents cas du tableau 2 et nous nous sommes aperçus qu'il est possible d'avoir entre des transitions de type $g = 1 + 4.r$ et $g = 1$ des différences de consommation jusqu'à dix fois plus importantes ainsi qu'un délai de propagation sur le fil jusqu'à cinq fois plus important.

Outre le problème de diminution de la vitesse de transmission, le $crosstalk$ est également une source de bruit et peut induire des erreurs à la réception des données. Prenons par exemple le cas où les fils agresseurs A1 et A2 effectuent une transition descendante et où le fil victime reste constant à V_{dd} $(\downarrow, -, \downarrow)$. L'expérimentation effectuée dans la configuration normale de la technologie UMC 0.13μm est illustrée à la figure 5. Le pic de tension sur le fil victime est très élevé ; le bruit résultant parvient presque à la tension de basculement d'un inverseur qui pourrait servir de récepteur en fin de bus. Il est facilement possible d'imaginer ce que pourrait donner ce bruit en addition de tous les autres bruits du circuit (bruit de l'alimentation, variations de paramètres etc).

Le phénomène de $crosstalk$ contribue donc, à faire augmenter la puissance dynamique puisque celle-ci dépend linéairement de C_L ; il contribue également à faire augmenter le délai de propagation d'une donnée et peut induire des erreurs à la réception des données.

Aujourd'hui avec les technologies dites submicroniques (diminution des dimensions et des espacements entre fils entre autre), l'effet du $crosstalk$ devient de plus en plus important. La section 4 présentera les techniques qui essaient de limiter ce phénomène.

La partie suivante va expliquer notre approche de modélisation de la consommation.

C_L	Types de transition			g
C_s	$(\uparrow,\uparrow,\uparrow)$	$(\downarrow,\downarrow,\downarrow)$		1
$C_s + C_c$	$(-,\uparrow,\uparrow)$ $(\downarrow,\downarrow,-)$	$(-,\downarrow,\downarrow)$	$(\uparrow,\uparrow,-)$	$1 + r$
$C_s + 2.C_c$	$(-,\uparrow,-)$ $(\uparrow,\uparrow,\downarrow)$	$(-,\downarrow,-)$ $(\uparrow,\downarrow,\downarrow)$	$(\downarrow,\downarrow,\uparrow)$ $(\downarrow,\uparrow,\uparrow)$	$1 + 2.r$
$C_s + 3.C_c$	$(-,\uparrow,\downarrow)$ $(\downarrow,\uparrow,-)$	$(-,\downarrow,\uparrow)$	$(\uparrow,\downarrow,-)$	$1 + 3.r$
$C_s + 4.C_c$	$(\uparrow,\downarrow,\uparrow)$	$(\downarrow,\uparrow,\downarrow)$		$1 + 4.r$

TAB. 2 – Capacité parasite (C_L) et facteur de délai (g) du fil victime en fonction du type de transition.

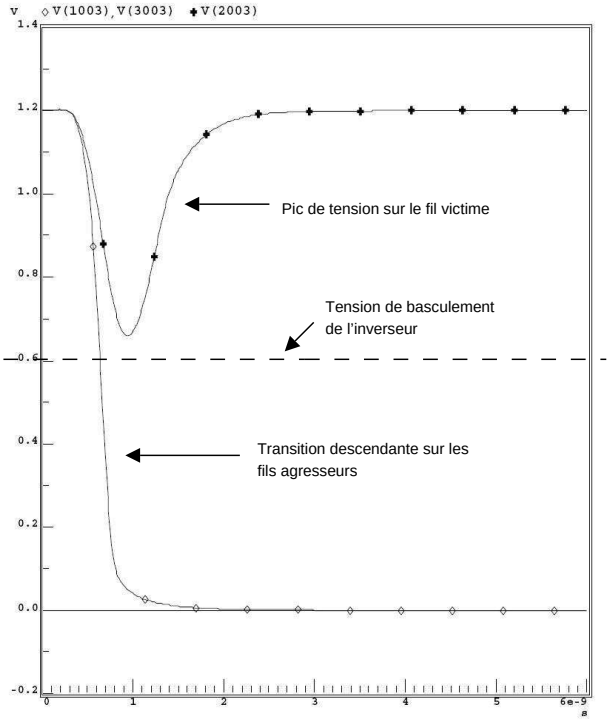

FIG. 5 – Bruit sur un fil victime causé par la transition simultanée de deux fils agresseurs.

3.2 Modélisation de la consommation

Comme nous l'avons vu précédemment la résistance R et la capacité C du fil varient en fonction de la longueur L de ce dernier. La longueur du bus sera donc un paramètre pour l'estimation de consommation.
Notons également que la capacité du fil C_s varie selon le niveau de métal utilisé puisque la hauteur (H) du fil par rapport au substrat est différente ; la capacité de $crosstalk$ C_c varie elle aussi puisque l'espacement (S) entre les fils est différent ; le niveau de métal utilisé sera donc un paramètre de l'estimation.
Pour finir, il a été mis en évidence que selon le type de transition la capacité totale C_L varie également, donc le type de transition sur le bus sera un autre paramètre pour l'estimation.
Différentes expérimentations ont été effectuées avec un simulateur SPICE (ELDO V5.7) et une technologie CMOS UMC 0.13μm afin d'obtenir un modèle précis au niveau physique en termes de consommation et de vitesse. Au terme des expérimentations, nous avons développé un outil qui fournit à l'utilisateur par l'intermédiaire d'une interface graphique des résultats en termes de surface, de consommation (instantanée, moyenne et maximale) et de vitesse maximale de transmission. Les paramètres d'entrée de cet outil sont listés dans le tableau 3. La question qui se pose maintenant est de savoir comment faire pour réduire la consommation du transfert de données ou d'en augmenter la vitesse en supprimant par exemple les pires cas de commutations du tableau 2. Pour cela, il faut pouvoir jouer sur les paramètres qui entrent en compte dans l'équation de $P_{dynamique}$. Les seuls paramètres qui peuvent être optimisés sont, l'activité α ainsi que la capacité globale C_L (V_{dd} et V_{swing} étant dépendants de la technologie). La section suivante va présenter les techniques qui permettent de jouer sur ces paramètres.

Paramètre	Type	Variation
Niveau de métal	Technologique	1 à 6
Largeur du bus (n bits)	Architectural	¿ 0
Longueur du bus (m)	Architectural	¿ 0
Frequence (MHz)	Architectural	¿ 0
Type de bufferisation	Technologique	Single / Full
Flux de données	Algorithmique	-
Blindage	Techno / Algo	GND/Vdd...

TAB. 3 – Paramètres d'entrée de l'outil avec leur type et zone de variation.

4 TECHNIQUES DE RÉDUCTION DE LA CONSOMMATION ET DU DÉLAI

Cette section va illustrer quelles sont les techniques existantes afin de réduire la consommation et le délai sur les bus et également voir à quel niveau d'abstraction elles interviennent comme le montre la figure 6 [Ragunathan, 1998].

4.1 Au niveau technologique

Les techniques au niveau technologique visent à modifier les paramètres physiques des fils. Il est possible de, soit jouer sur les dimensions des fils (H, T, W), soit jouer sur l'espacement (S) entre les fils afin de diminuer la capacité C_L [Macchiarulo, 2002].
Au niveau technologique, la solution la plus simple et la plus utilisée est la technique du blindage $(shielding)$. Les méthodes de blindage consistent à insérer un fil relié à la masse ou à l'alimentation entre deux fils adjacents de manière à se ramener au cas où les agresseurs sont stables (figure 7a). Tous les cas où deux fils adjacents effectuent des transitions en sens inverse sont alors éliminés. Une évolution de cette technique peut être trouvée dans [Khatri, 2001] où par exemple on effectue une alternance de blindage à la masse et à V_{dd} (figure 7b) ; dans [Taylor, 2001], la technique de blindage rete-

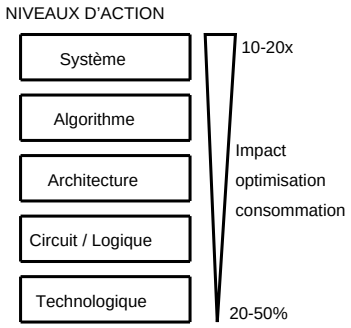

FIG. 6 – Niveaux d'action et gains pour l'optimisation en consommation [Ragunathan, 1998].

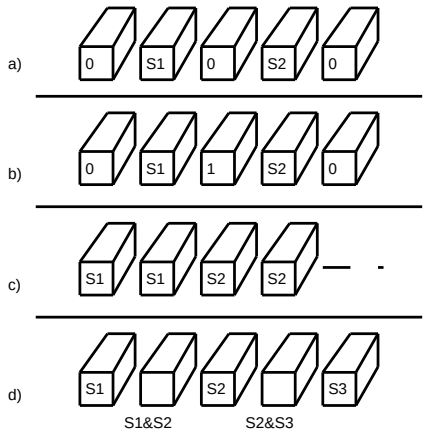

FIG. 7 – Illustration des types de blindages (0 :masse 1 :V_{dd} S :signal).

nue est que le fil inséré a le niveau logique du ET logique de ses deux voisins (figure 7d). Une autre méthode très simple d'application du blindage consiste à dupliquer chaque fil en transmettant sur le fil adjacent le même signal (figure 7c). L'accélération apportée par cette technique est supérieure à celle de [Khatri, 2001] car les cas où les deux agresseurs sont stables sont en plus éliminés. Le principal atout de ces méthodes est qu'elles permettent d'augmenter la vitesse de transmission des données puisque les pires cas du tableau 2 sont éliminés. Du fait du doublement de la surface (car doublement du nombre de fils) les techniques de blindage ne sont pas efficaces en termes de consommation car par exemple pour la méthode de duplication, l'activité α est doublée.
Une autre technique très connue est la technique qui consiste à insérer des répéteurs sur les chemins de données afin d'en réduire le délai [Bakaglu, 1985][Chen, 2004]. Cette technique permet d'augmenter fortement la vitesse sur les bus mais se fait au détriment de la consommation puisque les répéteurs contribuent à faire augmenter la capacité C_L
Bien que les techniques de blindage et d'insertion de répéteur contribuent à augmenter la consommation, elles sont très utilisées ; nous les avons donc intégrées au sein de l'outil.

4.2 Au niveau circuit/logique

Une solution consiste ici à décaler intentionnellement les signaux du bus pour éviter d'avoir les transitions des fils adjacents au même instant $(signalskewing)$ [Hirose, 2000] ; les fils pairs et impairs du bus sont déphasés alternativement, ainsi le fil du milieu effectue sa transition lorsque ses voisins sont stables. Cette technique nécessite malheureusement une conception très complexe des émetteurs et récepteurs.
Une autre méthode consiste à utiliser des répéteurs à tensions de seuil variables [Shang, 2003] afin de limiter les excursions sur les lignes de bus et donc de limiter C_L. L'inconvénient de cette solution est que les composants utilisés ne sont pas standards.

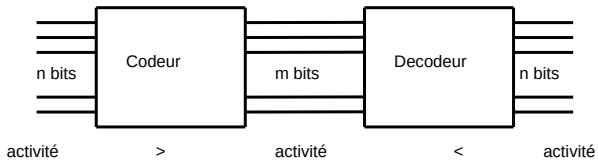

FIG. 8 – Principe du codage des données.

4.3 Au niveau architectural

C'est à ce niveau que le plus grand nombre de techniques a été proposé afin de réduire l'activité α et la capacité C_L. Elles consistent toutes en un codage de données tel que le montre la figure 8. Le but du codage est de transmettre l'information sur m bits avec $(m \geq n)$ tel que l'activité des données codées soit inférieure à celle des données non codées. Ces différents codages sont soit adaptés pour le bus d'adresses soit pour le bus de données.

4.3.1 Codages dédiés au bus d'adresses

La majeure partie du temps les adresses accédées sur la mémoire d'instructions ou de données sont consécutives. L'idée du codage présentée dans [Su, 1994, Su, 1995] est de faire en sorte que l'on ait une seule transition sur le bus à chaque fois que l'on accède à une adresse consécutive de celle accédée au cycle précédent. Ce codage est appelé *codage de Gray*.
Dans [Benini, 1997], l'idée proposée est de rajouter un fil que l'on positionne à un niveau logique défini lorsque les adresses accédées sont consécutives. Ce codage est appelé *T0 Code*.
Dans [Fornaciari, 2000], on peut trouver une variante du *T0 Code* où il est possible de définir plusieurs pas d'incrémentations pour les accès consécutifs. Ces deux techniques ont l'avantage de réduire l'activité sur le bus à zéro lorsque les valeurs sont consécutives, mais la surface et la complexité des codecs (codeurs et décodeurs) est très importante puisqu'ils nécessitent plusieurs bancs de registres ainsi que des additionneurs, multiplexeurs etc.

4.3.2 Codages dédiés au bus de données

Les données transitant sur le bus de données sont considérées comme aléatoires. L'idée du codage présentée dans [Stan, 1995] est de comparer le nombre de bits changeant entre la donnée $n-1$ et la donnée n, si cette différence est supérieure à la moitié de la largeur du bus, alors la donnée n envoyée est inversée. Cette technique est efficace pour de larges bus et est appelée *Bus Invert*.
Afin de rendre la technique plus efficace, [Shin, 1998] propose d'appliquer le *Bus Invert* à la seule partie du bus qui a la plus forte activité. L'inconvénient est qu'il faut donc connaître à l'avance les données qui vont circuler sur le bus, afin de faire l'inversion sur les fils ayant la plus forte activité. Cette technique est appelée le *Partial Bus Invert*.
Une autre technique présentée dans [Komatsu, 1999] appelée *CodeBook* vise à stocker i anciennes valeurs transmises sur le bus et transmettre au cycle courant la valeur qui a le moins de différence avec celles transmises

	Code1	Code2	
Bloc original	Codage	Codage1	Codage2
00	0000	0001	0000
01	0001	0011	1000
10	0011	0111	1100
11	0111	1111	1110

TAB. 4 – Correspondance entre les blocs originaux et les blocs codés.

aux i cycles précédents. On doit également transmettre sur plusieurs fils supplémentaires le code de la valeur envoyée afin de décoder la bonne valeur à la réception
Nous avons vu précédemment des techniques de blindage spatial, il est également possible de faire du blindage temporel, *Code 0* présenté dans [Philippe, 2006]. Pour cela entre chaque émission d'une donnée tous les bits du bus sont remis à 0. Dans [Philippe, 2006], il est également présenté deux autres types de codages, le *Code 1* et le *Code 2*. Le *Code 1* consiste à coder des blocs de deux bits en blocs de quatre bits, ce code élimine les pires cas mais malheureusement augmente l'activité puisque le codage du bloc 11 comprend une transition montante. Le *Code 2* consiste à coder des blocs de deux bits en deux blocs de quatre bits qui seront alternativement envoyés sur le bus. L'inconvénient des *Code 0, Code1, et Code 2* est qu'il est nécessaire de fonctionner à fréquence double sur le bus car le transfert des informations est doublé. Il existe encore d'autres méthodes telles que, le *XOR Code* qui consiste à transmettre seulement les transitions entre la donnée n et la donnée $n-1$ en effectuant un ou-exclusif du bit à l'instant n avec le bit à l'instant $n-1$ sur chaque fil ; l'*Offset Code* qui consiste à transmettre la différence entre la donnée n et la donnée $n-1$. Diverses autres techniques [Benini, 1998] mélangent plusieurs des techniques présentées ci-dessus, par exemple pour des bus multiplexés lorsque l'on transmet des données le codage utilisé est le *Bus Invert* et au cycle où l'on transmettra l'adresse, le codage utilisé sera le *T0 Code*.

5 RÉSULTATS EXPÉRIMENTAUX

L'outil d'estimation que nous avons développé, va permettre d'analyser les performances de ces techniques en termes de variation de l'activité α, de la capacité globale C_L, de la vitesse, de la surface et de la consommation énergétique sur le bus. Les mesures de consommation ont été effectuées sur le niveau de métal 2 de la technologie UMC $0.13\mu m$ pour des bus de longueur 3 et 10 mm. Une première campagne de mesure a été effectuée pour un flot de 6.10^4 bits aléatoires par fil puis une seconde pour un flot de 6.10^4 valeurs consécutives. Notons que quelque soit le niveau de métal utilisé ainsi que la longueur, les évolutions des paramètres représentés par les symboles $\nearrow$ (augmentation), $\searrow$ (diminution) et $-$ (pas de variation) restent identiques. Les résultats sont répertoriés dans les tableaux 5 et 6.
D'une manière générale, nous pouvons remarquer que quand la capacité C_L diminue, la vitesse de transfert des données est augmentée. De plus comme nous l'avons souligné précédemment certaines techniques sont inefficaces si elles ne s'appliquent pas sur le bus pour lesquelles elles ont été créées. Il est possible de citer les exemples de *Gray, T0, XOR, Offset* qui sont inefficaces pour un flot de données aléatoires car elles ne permettent ni de réduire l'activité α ni de réduire la capacité C_L ; et *Bus Invert, Code 0, Code 1, Code2* qui sont inefficaces pour un flot de valeurs consécutives.
Les techniques au niveau technologique sont très efficaces pour réduire C_L et donc augmenter la vitesse de transmission, mais le fait que la surface soit systématiquement doublée les rend inefficaces en termes de consommation car plus de fils commutent.
Aucune de ces techniques n'est optimale en termes de surface ceci est dû au rajout des codecs plus ou moins complexes ou au doublement de la surface dédiée aux fils. Comme le présente les tableaux 5 et 6, certaines de ces techniques semblent vraiment performantes pour réduire la consommation sur le bus telles que le *Bus Invert, T0, Gray, Offset* par exemple.
Au vu de la complexité de leur codecs qui utilisent des éléments tels que des bancs de registres, des portes XOR, des multiplexeurs et même des additionneurs soustracteurs on peut se poser la question de savoir si le surcoût de consommation engendré par la consommation des codecs n'est pas plus important que la diminution de consommation qui est apportée sur le bus par le codage ?
Prenons l'exemple d'un bus de n bits, le surcoût en consommation (d'après les valeurs fournies dans la bibliothèque UMC $0.13\mu m$) apporté par les $2.n$ portes XOR utilisées dans un codage de type *Bus Invert ou Gray* par exemple (sans regarder la consommation des registres et autres portes), est déjà supérieur à la réduction de consommation apportée sur le bus. Il en est de même pour les additionneurs et soustracteur utilisés dans les codages de type *T0 et Offset*.
Bien que ces techniques de codage permettent de réduire la consommation sur le bus, elles ne sont efficaces que pour des bus très longs où la capacité C_L devient très grande.
Dans [Kretzschmar, 2004] il a été démontré que plusieurs de ces techniques ne sont efficaces que pour des bus dont la longueur excède 15 mm pour la technique la plus performante (certaines techniques sont même efficaces pour des bus de 100 mm minimum). Or ces longueurs de bus ne sont pas réalistes dans les SOC actuels où les longueurs maximales n'excèdent guère une dizaine de mm.

6 CONCLUSION

Dans un premier temps, tous les problèmes qui découlent des couplages capacitifs tels que le couplage entre les fils (*crosstalk*) ont été présentés. Nous avons souligné que le *crosstalk* est à l'origine d'une augmentation de la consommation (jusqu'à dix fois plus pour des transitions de type $g = 1 + 4.r$ par rapport à des transitions de type $g = 1$). Il est également à l'origine d'une diminution de la vitesse de transmission des données (jusqu'à cinq fois

	Activité α	Capacité C_L		Vitesse	Surface	Consommation du bus E(nJ)		
	Δ°	Δ°	Pire cas	Δ°	Δ°	Δ°	L=3mm	L=10mm
Données aléatoires	1/2	-	$C_e + 4.C_c$	-	-	-	124	391
TECHNIQUES AU NIVEAU TECHNOLOGIQUE								
Spacing	-	$\searrow C_c$	$C_e + 4.C_c'$	$\nearrow$	$\nearrow$ xfois surface des fils	$\searrow$	x	x
Shielding GND	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ x2	$\nearrow$	142	419
Shielding GND/Vdd	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ x2	$\nearrow$	142	418
Shielding AND	$\searrow$	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ x2	$\searrow$	122	352
Duplication fils	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ x2	$\nearrow$	150	451
TECHNIQUES AU NIVEAU ARCHITECTURAL								
Gray	-	-	$C_e + 4.C_c$	-	$\nearrow$ surface codecs	-	124	392
T0	-	-	$C_e + 4.C_c$	-	$\nearrow$ surface codecs + 1fil	-	124	392
Bus Invert	$\searrow$	-	$C_e + 4.C_c$	-	$\nearrow$ surface codecs + 1fil	$\searrow$	110	341
XOR Code	-	-	$C_e + 4.C_c$	-	$\nearrow$ surface codecs	-	124	392
Offset Code	-	-	$C_e + 4.C_c$	-	$\nearrow$ surface codecs	$\searrow$	122	385
Code 0	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\nearrow$	155	466
Code 1	$\searrow$	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\nearrow$	157	464
Code 2	$\searrow$	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\searrow$	97	283

TAB. 5 – Evolution des paramètres activité, capacité, vitesse, surface et énergie pour un flux de données aléatoires en fonction des techniques de codage.

plus lent pour des transitions de type $g = 1 + 4.r$ par rapport à des transitions de type $g = 1$) et, de plus il peut être à l'origine d'erreurs au niveau de la réception.
Dans un second temps nous avons expliqué le type de modélisation du bus que nous avons utilisé pour mener nos expérimentations.
Beaucoup des techniques de blindage et de codage des données présentées dans la section 4 permettent de réduire la capacité C_L (donc le phénomène de *crosstalk*) et donc la consommation sur le bus. Celles-ci mettent en oeuvre des architectures de codecs assez complexes et contribuent considérablement à augmenter la surface occupée. Pour qu'elles soient efficaces, il faut que le surcoût en consommation apporté par les codecs ne soit pas plus important que la baisse de consommation apportée sur le bus. Ce qui n'est malheureusement plus le cas pour les longueurs de bus que l'on trouve actuellement dans les SOC.
L'évolution des systèmes sur silicium vers des architectures de plus en plus complexes, intégrant toujours plus de composants sur des surfaces de circuits toujours plus grandes, impose l'utilisation de nombreuses lignes d'interconnexions malgré les difficultés de délai et de consommation que cela occasionne. L'objectif de nos futurs axes de recherches sera de s'orienter vers une exploration de nouvelles solutions, qui permettront de réduire la consommation sur le bus et les phénomènes capacitifs, tout en essayant de ne pas augmenter la consommation globale.

BIBLIOGRAPHIE

[Bakaglu, 1985] Bakaglu, H.B. & Meindl, J.D. "Optimal Interconnection Circuits for VLSI" IEEE Transactions on Electron Devices, vol 32, No 5, p. 903-909, 1985

[Benini, 1997] Benini, L. ; Micheli, G.D. ; Macii, E. ; Sciuto, D. & Silvano, C. "Asymptotic Zero-Transition Activity Encoding for Address Busses in Low-Power Microprocessor-Based Systems" in the Proceedings of the 7th IEEE Great Lakes Symposium on VLSI (GLS), p. 77-82, Urbana, USA (1997)

[Benini, 1998] Benini, L. ; Micheli, G.D. ; Macii, E. ; Sciuto, D. & Silvano, C. "Address bus encoding techniques for system-level power optimization" in the Proceedings of the conference on Design, automation and test in Europe (DATE), p. 861-867, Paris, France (1998)

[Chen, 2004] Chen, G. & Friedman, E. G. "Low Power Repeaters Driving RC Interconnect with Delay and Bandwidth Constraints" in the Proceedings of the IEEE International SOC Conference, p. 335-339, 2004.

[Dally, 1998] Dally, W.J. & Poulton, J.W. "Digital Systems Engineering" Cambridge University Press, 1998

[Fornaciari, 2000] Fornaciari, W. ; Polentarutti, M. ; Sciuto, D. & Silvano, C. "Power optimization of system-level address buses based on software profiling" in the Proceedings of the 8th international workshop on Hardware/software codesign (CODES), p. 29-33, San Diego, USA (2000)

[Hirose, 2000] Hirose, K. & Yasuura, H. "A bus delay reduction technique considering crosstalk" in the Proceedings of the conference on Design, automation and test in Europe (DATE), p. 441-445, Paris, France (2000)

[Ho, 2001] Ho, R. ; Mai, K. & Horowitz, M. "The Future of Wires" Proceedings of the IEEE, Vol 89, p. 490-504, 2001

[ITRS, 2004] ITRS. "Technical report", International Technology Roadmap for Semiconductors, 2004

[Khatri, 2001] Khatri, S.P. ; Brayton, R.K. & Sangiovanni-Vincentelli, A.L. "Crosstalk Noise Immune VLSI Design Regular Layout Fabrics" Kluwer Academic Publishers, 2001

	Activité α	Capacité C_L		Vitesse	Surface	Consommation du bus E(nJ)		
	Δ°	Δ°	Pire cas	Δ°	Δ°	Δ°	L=3mm	L=10mm
Adresses séquentielles	1/4	-	$C_e + 3.C_c$	-	-	-	37	91
TECHNIQUES AU NIVEAU TECHNOLOGIQUE								
Spacing	-	$\searrow C_c$	$C_e + 3.C_c'$	$\nearrow$	$\nearrow$ xfois surface des fils	$\searrow$	x	x
Shielding GND	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ x2	$\nearrow$	59	128
Shielding GND/Vdd	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ x2	$\nearrow$	59	128
Shielding AND	$\nearrow$	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ x2	$\nearrow$	47	90
Duplication fils	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ x2	$\nearrow$	55	117
TECHNIQUES AU NIVEAU ARCHITECTURAL								
Bus Invert	-	$\nearrow$	$C_e + 4.C_c$	$\searrow$	$\nearrow$ surface codecs + 1fil	$\nearrow$	44	112
Gray	$\searrow$	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\searrow$	29	64
XOR Code	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\searrow$	30	67
Offset Code	$\searrow$	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\searrow$	16	16
Code 0	$\nearrow$	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\nearrow$	155	465
Code 1	$\nearrow$	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\nearrow$	103	282
Code 2	-	$\searrow$	$C_e + 2.C_c$	$\nearrow$	$\nearrow$ surface codecs	$\nearrow$	47	87
T0 Code	$\searrow$	$\searrow$	$C_e + C_c$	$\nearrow$	$\nearrow$ surface codecs + 1fil	$\searrow$	17	17

TAB. 6 – Evolution des paramètres activité, capacité, vitesse, surface et énergie pour un flux d'adresses consécutives en fonction des techniques de codage.

[Komatsu, 1999] Komatsu, S. ; Ikeda, M. & Asada, K. "Low Power Chip Interface Based on Bus Data Encoding with Adaptive Code-Book Method" in the Proceedings of the 9th IEEE Great Lakes Symposium on VLSI (GLS), p. 368-371, Ann Arbor, USA (1999)

[Kretzschmar, 2004] Kretzschmar, C. ; Nieuwland, A.K. & Muller, D. "Why Transition Coding for Power Minimization of On-Chip Buses Does Not Work" in the Proceedings of the conference on Design, automation and test in Europe (DATE), p. 10512-10517, Paris, France (2004)

[Macchiarulo, 2002] Macchiarulo, L. ; Macii, E. & Poncino, M. "Wire Placement for Crosstalk Energy Minimization in Address Buses" in the Proceedings of the conference on Design, automation and test in Europe (DATE), p. 158-162, Paris, France (2002)

[Magen, 2004] Magen, N. ; Kolodny, A. ; Weiser, U. & Shamir, N. "Interconnect-power dissipation in a microprocessor" in the Proceedings of the international workshop on System level interconnect prediction (SLIP), p. 7-13, Paris, France (2004)

[Philippe, 2006] Philippe, J.M. ; Pillement, S. & Sentieys, O. "Area Efficient Temporal Coding Schemes for Reducing Crosstalk Effects" in the Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED), p. 334-339, San Jose, USA (2006)

[Rabaey, 2003] Rabaey, J.M. ; Chandrakasan, A. & Nikolic, B. "Digital Integrated Circuits : A design perspective" Pearson Education, 2003

[Ragunathan, 1998] Ragunathan, A ; Jha, N.K. & Dey, S. "'High-level Power Analysis and Optimization" Kluwer Academic Publishers, 1998

[Shang, 2003] Shang, L. ; Peh, L. & Jha, N.K. "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks" in the Proceedings of the 9th International Symposium on High-Performance Computer Architecture (HPCA), p. 91-102, Anaheim, USA (2003)

[Shin, 1998] Shin, Y. ; Chae, S.I. & Choi, K. "Reduction of bus transitions with partial bus-invert coding" Electronics Letters, Issue 7, Vol 34, p. 642-643, 1998

[Stan, 1995] Stan, M. & Burleson, W.P. "Bus-Invert Coding for Low-Power I/O" IEEE Transaction on Very Large Scale Integration Systems, Vol 3, N° 1, p. 49-58, 1995

[Su, 1994] Su, C. ; Tsu, C.Y. & Despain, A.M. "Saving power in the control path of embedded processors" Design & Test of Computers, IEEE, Vol. 11, N° 4, p. 24-31, 1994

[Su, 1995] Su, C.L. & Despain, A.M. "Cache design trade-offs for power and performance optimization : a case study" in the Proceedings of the international symposium on Low power design (ISLPED),p. 63-68, Dana Point, USA (1995)

[Taylor, 2001] Taylor, C.N. ; Dey, S. & Zhao, Y. "Modeling and minimization of interconnect energy dissipation in nanometer technologies" in the Proceedings of the 38th conference on Design automation (DAC), p. 754-757, Las Vegas, USA (2001)