

## Multiplieur analogique en technologie SOI pour le décodage de turbo-codes

Thibault KERVYN, Nicolas DONCKERS, Michel VERLEYSSEN<sup>1</sup>, Denis FLANDRE

Université catholique de Louvain  
Laboratoire de microélectronique  
3, place du Levant  
B-1348 Louvain-la-Neuve (Belgique)

E-mail : {donckers, verleysen, flandre}@dice.ucl.ac.be

### Résumé

*Ces dernières années, une classe d'algorithmes pour la correction d'erreurs de transmission a fait son apparition et est rapidement apparue comme étant de loin plus performante que les techniques existantes: les turbo-codes. L'utilisation de ces techniques dans les réseaux mobiles est conditionnée à la possibilité de pouvoir effectuer le décodage (complexe) dans un appareil embarqué alimenté par une batterie. Classiquement effectué par voie digitale, il a été montré récemment que le décodage d'un turbo-code pouvait également être réalisé au moyen d'un circuit analogique dont l'élément de base est le multiplieur. Dans cet article, nous mettons en évidence les avantages de l'utilisation de la technologie SOI pour la réalisation de multiplieurs analogiques basse puissance et comparons les performances du circuit conçu avec des réalisations existantes.*

### 1. Introduction

Ces dernières années, les techniques dites "turbo" ont envahi les domaines des télécommunications. Initialement destinées à améliorer les performances des codes correcteurs d'erreurs, ces techniques ont également été appliquées avec succès à d'autres problèmes tels l'égalisation ou la détection multi-utilisateurs. De nombreux articles ont démontré que les performances de ces techniques sont de loin supérieures à celles de la plupart des techniques connues à ce jour et approchent les limites prédites par Shannon. De ce fait, les techniques "turbo" constituent des éléments clés des nouveaux standards de téléphonie mobile de troisième génération.

Bien que l'encodage d'une séquence au moyen d'un code turbo ne soit pas d'une grande complexité, il n'en va pas de même pour le décodage. Un décodage optimal nécessite plusieurs itérations d'un algorithme (l'algorithme MAP), au départ déjà complexe. Bien que réalisable au moyen d'un circuit digitale, ce décodage pourrait se réaliser de manière avantageuse au moyen de circuits analogiques.

Un turbo codeur est constitué de l'association de codeurs élémentaires, par exemple des codeurs convolutionnels. A

l'image de l'encodeur, le décodeur est également constitué de plusieurs décodeurs élémentaires échangeant de l'information au cours d'un processus itératif permettant d'améliorer progressivement la fiabilité des décisions prises (Figure 1). L'objectif de cet article n'étant pas de présenter l'algorithme de turbo-décodage, on se référera à [1] pour plus de détails. Il faut en retenir que l'information échangée par les décodeurs est de l'information douce (c'est-à-dire des valeurs continues), qu'il est plus facile de véhiculer au moyen d'un signal analogique que digitale. L'algorithme MAP, implémenté dans chacun des décodeurs élémentaires, travaille également sur des valeurs continues [2].

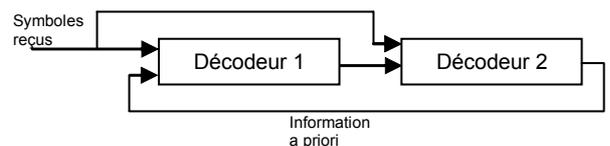


Figure 1. Structure générale d'un turbo-décodeur

On peut bien entendu opter pour une solution digitale: les signaux reçus sont échantillonnés, encodés en format binaire, le reste du traitement se faisant de manière digitale. Il n'est néanmoins pas évident que cette solution permette de réaliser un décodeur qui satisfera aux contraintes de vitesse et de consommation imposées par son utilisation dans un appareil alimenté sur batterie.

L'algorithme MAP [2] constitue un cas particulier de l'algorithme "sommes et produits". Il peut être implémenté au moyen d'additions et de multiplications. Un décodeur MAP hardware complet peut être réalisé par la combinaison d'un grand nombre de cellules, directement dérivées d'un multiplieur [3]. L'optimisation de cette cellule de base devient donc un enjeu primordial.

### 2. Structure d'un décodeur MAP analogique

Un code convolutionnel systématique peut être décrit sous la forme d'un treillis [4]. Ce dernier décrit complètement le code et le décodage de la séquence reçue se réalise au moyen d'un circuit qui est directement dérivé de ce treillis.

<sup>1</sup> Michel Verleysen est Maître de Recherche du Fonds National de la Recherche Scientifique Belge.

Soit l'encodeur de la Figure 2. La séquence codée  $x_{p,i}$  est calculée à partir de l'état du codeur (le contenu des registres à décalage D) et de l'entrée courante  $u_i$ . On peut représenter ce codeur sous la forme du treillis de la Figure 3. L'axe horizontal représente l'indice temporel  $i$ . Verticalement, on a représenté tous les états possibles  $S_j$  du codeur (4 car codés sur 2 bits). Les branches du treillis représentent toutes les transitions possibles en fonction des symboles reçus. Au départ de chaque état, 2 branches partent et aboutissent à l'état suivant en fonction de la valeur de l'entrée.

Ce treillis est directement transposable en circuit électronique. En effet, on commencera par noter que toutes les sections du treillis sont identiques (à l'exception de celles aux extrémités). Si, au temps  $i$ , on connaît les probabilités associées aux différents états ainsi que celles associées au symbole reçu, on peut déterminer celles associées aux états possibles pour l'instant  $i+1$ .

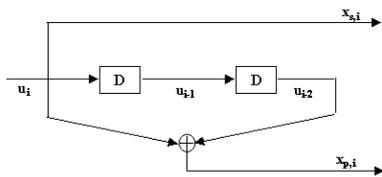


Figure 2. Encodeur convolucional.

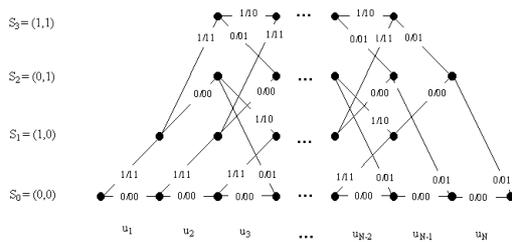


Figure 3. Treillis relatif au codeur de la Figure 2.

Par exemple, à partir de la Figure 3, on peut établir la probabilité de se trouver dans l'état  $S_0$  à l'instant  $i+1$  à partir des probabilités associées à l'instant précédent comme suit:

$$P(S_0, i+1) = P(S_0, i) \cdot P(x=0) + P(S_2, i) \cdot P(x=0) \quad (1)$$

où,  $P(S_j, i)$  est la probabilité d'être dans l'état  $j$  à l'instant  $i$ ,  $P(x=0)$  est la probabilité que le symbole reçu soit 0.

On peut évidemment, par un raisonnement identique, déterminer les probabilités associées à chaque état. On retiendra de cette présentation, dont une analyse fouillée peut être obtenue en consultant [3], qu'un décodeur est constitué de la mise en série de cellules identiques. Chaque cellule admet pour entrées les probabilités associées aux états précédents ainsi que celles associées aux signaux reçus. Chaque cellule calcule les probabilités associées aux états suivants. Etant donné le grand nombre de cellules élémentaires à utiliser (autant que de sections dans le treillis, c'est-à-dire que de symboles dans une trame), leur optimisation devient un enjeu majeur.

Il n'est a priori pas évident que ces cellules dont nous avons besoin dérivent directement d'une structure bien connue: le multiplieur de Gilbert. C'est ce que nous nous proposons de développer dans la section suivante.

### 3. Cellule de Gilbert en technologie MOS

Notre cellule de base est une extension de la cellule initialement présentée par Gilbert [5]. Dans sa version originale, cette cellule est réalisée au moyen de transistors bipolaires. Afin de conserver les propriétés de cette cellule et de rencontrer nos objectifs en termes de consommation et de performances, nous utiliserons des transistors MOS fonctionnant en faible inversion et réalisés en technologie SOI. Cette dernière possède, par rapport à une technologie Bulk classique, l'avantage d'une plus faible tension de seuil ainsi que de capacités parasites réduites. Notre objectif de dessiner un circuit rapide et basse puissance ne peut que profiter de ces caractéristiques.

#### 3.1 Multiplieur en mode tension

Le noyau de base du multiplieur est présentée à la Figure 4. On suppose que tous les transistors sont saturés. Il est alors simple d'établir les relations suivantes:

$$I_{D1} = \frac{I_0}{1 + e^{-\frac{V_Y}{nU_T}}} \quad \text{et} \quad I_{D2} = \frac{I_0}{1 + e^{\frac{V_Y}{nU_T}}}$$

où,  $n$  est le facteur d'effet de substrat  
 $U_T$  est la tension thermique:  $k.T/q$   
 $I_0$  est le courant de polarisation

qui permettent finalement d'obtenir:

$$I_{OUT} = I_{OUT1} - I_{OUT2} = I_0 \cdot \tanh\left(\frac{V_X}{nU_T}\right) \cdot \tanh\left(\frac{V_Y}{nU_T}\right)$$

où,  $V_X = V_{X1} - V_{X2}$  et  $V_Y = V_{Y1} - V_{Y2}$

L'opération de multiplication s'obtient en considérant que les signaux d'entrée ne sont pas trop grands. On peut alors utiliser l'approximation:  $\tanh(x) \approx x$

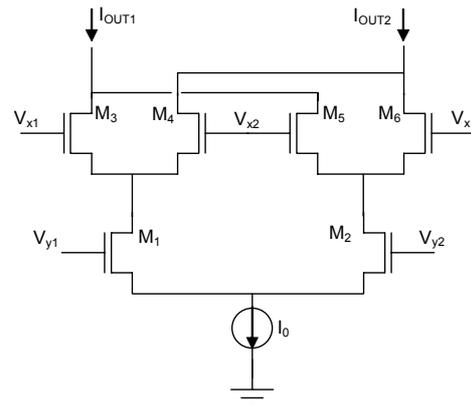


Figure 4. Version MOS du multiplieur de Gilbert.

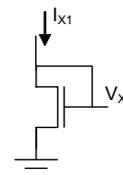


Figure 5. Convertisseur courant - tension.

### 3.2 Multiplieur en mode courant

Comme nous l'avons souligné dans l'introduction, notre circuit devra effectuer de nombreuses additions et multiplications. Cette contrainte nous amène à nous diriger vers un circuit qui fonctionne en mode courant et non en mode tension. Au départ de la cellule de Gilbert, on peut réaliser un multiplieur en mode courant en ajoutant à l'entrée du circuit, un convertisseur courant – tension. Le convertisseur le plus simple est réalisé au moyen d'un transistor monté en diode (Figure 5). En imposant à ce transistor de travailler en faible inversion, on obtient, avec les notations des Figures 4 et 5, les relations:

$$V_x = V_{x1} - V_{x2} = n.U_T \cdot \ln\left(\frac{I_{x1}}{I_{x2}}\right)$$

En introduisant ces relations dans celle ci-dessus liant le courant de sortie aux tensions d'entrée, on obtient la relation suivante dans laquelle la multiplication est cette fois exacte et ne résulte plus d'une approximation de la tangente hyperbolique:

$$I_{OUT} = I_0 \cdot \frac{I_{x1} - I_{x2}}{I_{x1} + I_{x2}} \cdot \frac{I_{y1} - I_{y2}}{I_{y1} + I_{y2}}$$

Le circuit ainsi réalisé est un circuit trans-linéaire. Un schéma de principe de ce type de circuit est représenté à la figure 6. Un avantage de ce genre de circuit est d'augmenter la dynamique; en effet, les signaux utiles sont "compressés" par le passage dans une fonction logarithmique avant d'être appliqués au circuit qui effectue les opérations utiles.

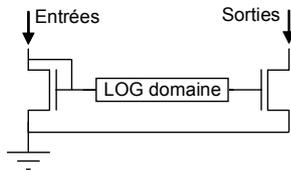


Figure 6: Circuits trans-linéaires.

### 3.3 Une cellule de base

A partir de ce qui précède, on peut établir le modèle de la cellule de base dont nous avons besoin pour réaliser notre décodeur analogique. Si on note  $I_{X,i}$  et  $I_{Y,j}$  les signaux d'entrée à multiplier, le circuit de la Figure 7 effectue tous les produits partiels  $I_{i,j} = I_{X,i} \cdot I_{Y,j}$  qu'il suffit d'additionner pour obtenir, conformément à l'équation (1), les probabilités cherchées. Sur base de la Figure 7, on peut écrire:

$$\frac{I_{i,j}}{I_{X,i}} = \frac{I_{i,j}}{\sum_k I_{i,k}} = \frac{I_0 \cdot e^{\frac{V_{Y,j} - n \cdot V_{X,i}}{n \cdot U_T}}}{\sum_k I_0 \cdot e^{\frac{V_{Y,k} - n \cdot V_{X,i}}{n \cdot U_T}}} = \frac{e^{V_{Y,j}}}{\sum_k e^{V_{Y,k}}}$$

où,  $V_{i,j}$  est le potentiel au nœud d'entrée du courant  $I_{i,j}$

En adoptant une démarche similaire pour calculer le rapport  $I_{y,j}/I_y$ , on obtient finalement:

$$I_{i,j} = I_z \cdot \frac{I_{X,i}}{I_X} \cdot \frac{I_{Y,j}}{I_Y}$$

où  $I_X = \sum I_{X,i}$ ,  $I_Y = \sum I_{Y,j}$  et  $I_z = \sum \sum I_{i,j}$

## 4. Dimensionnement

Notre cellule de base a été dimensionnée grâce à la méthode  $g_m/I_D$  [6]. Cette dernière permet d'assurer, pour chaque transistor, le mode d'opération approprié et, après avoir fixé le courant de polarisation, de déterminer la taille des transistors.

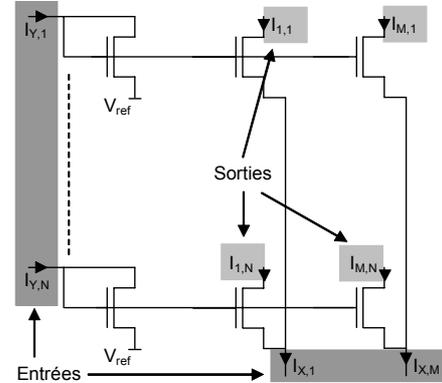


Figure 7: Généralisation de la cellule de Gilbert

### 4.1 Principes

Le paramètre  $g_m/I_D$ , rapport de la transconductance  $g_m$  au courant de drain  $I_D$ , est utilisé conjointement au courant normalisé  $I$ , défini par la relation ci-dessous. Il possède la propriété de ne dépendre que du degré d'inversion du transistor. Typiquement, il prend des valeurs allant de 3 ... 5 pour la forte inversion jusqu'à des valeurs de 30 ... 35 (resp. 20 ... 25) pour la faible inversion en technologie SOI complètement déserté (resp. en bulk CMOS conventionnel).

$$I = \frac{I_D}{\mu \cdot C_{ox} \cdot \frac{W}{L}}$$

En imposant, au paramètre  $g_m/I_D$  des valeurs pour chaque transistor, on peut aisément le forcer en faible inversion. Une fois le courant de drain fixé par des considérations de limitation de la puissance consommée, on peut déterminer les dimensions à donner à chaque transistor. La Figure 8 illustre la relation entre le  $g_m/I_D$  et  $I$  dans notre technologie SOI. A partir des dimensions obtenues, il est ensuite possible d'estimer les performances en fréquence du circuit en calculant les capacités parasites associées aux différents transistors.

### 4.2 Dimensionnement

Sur base des considérations ci-dessus, il devient alors aisé de dimensionner l'ensemble des transistors du circuit. Le résultat de cette procédure fournit  $g_m/I = 29 \text{ V}^{-1}$  pour tous les transistors – valeur arrondie pour obtenir des  $W$  et  $L$  entiers,  $(W/L)_{1,2} = 24/3$  et  $(W/L)_{3,6} = 12/3$ . Les diodes ajoutées à l'entrée du multiplieur et assurant la conversion tension – courant possèdent les mêmes caractéristiques que les transistors  $M_3$  à  $M_6$ .

## 5. Performances

Le circuit a été dessiné et ses performances simulées à partir du layout réalisé. La Figure 9 les illustre. La Table 1 compare notre circuit avec d'autres réalisations tirées de la littérature. Elle reprend, pour chaque réalisation, la tension d'alimentation (Vdd), la puissance consommée (Puis.) et la bande passante (B.P.).

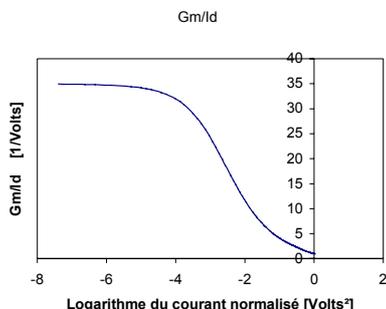


Figure 8: Courbe caractéristique d'un transistor MOS dans notre technologie.

| Réf.         | Vdd [V] | Puis. [mW] | B.P. [MHz] |
|--------------|---------|------------|------------|
| [8]          | 10      | 4          | 20         |
| [9]          | 10      | 8          | 30         |
| [10]         | 4.6     | 2.1        | 10.5       |
| [5]          | 15      | 75         | 500        |
| notre design | 1.5     | 0.0011     | 2          |

Table 1: Comparaison des performances.

Le comportement du circuit a été analysé grâce à une série de simulations sur Eldo 5.4 utilisant un modèle EKV et une technologie MOS SOI 2 $\mu$ m. L'alimentation et le courant de polarisation ont été fixés respectivement à 1,5V et 250 nA, ce qui engendre une puissance statique consommée totale de 1,1  $\mu$ W. La figure 9 illustre la linéarité de la multiplication sur la quasi totalité de la dynamique du courant d'entrée. Il s'agit bien entendu ici d'un gage de précision. Pour cette simulation, le premier courant d'entrée  $I_x$  varie de manière continue, le second prend ses valeurs entre 0 et 250 nA par pas de 25 nA.

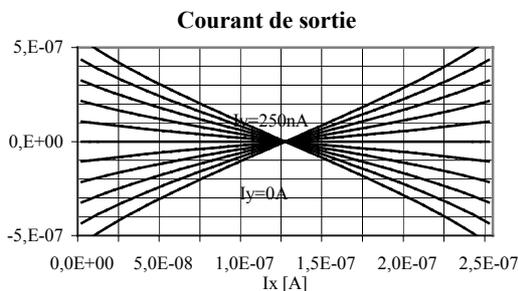


Figure 9: Simulation de la linéarité du circuit

La réponse transitoire du circuit a également été simulée. Pour une entrée constante, l'autre variant à la fréquence de 1 MHz, elle confirme le bon fonctionnement de notre montage. La vitesse maximale de fonctionnement est ici limitée par la longueur minimale de 2  $\mu$ m de notre

technologie. Nous avons estimé qu'en technologie SOI 0.25 $\mu$ m, la bande passante serait augmentée d'un facteur 10, à même consommation de puissance statique. Les performances seraient alors compatibles avec les besoins nécessaires à la réalisation d'un turbo décodeur inséré dans une application de télécommunication embarquée.

## 6. Conclusion

Cette étude sur l'optimisation d'un multiplieur analogique pour le décodage de codes correcteurs d'erreurs nous amène à tirer deux conclusions principales.

Tout d'abord, la structure initialement proposée par Gilbert semble fournir d'intéressantes propriétés. Elle permet, dans un circuit fonctionnant en mode courant, d'effectuer une multiplication exacte et non approchée [12]. On se rappellera que la facilité d'effectuer des additions nous a guidé vers un circuit en mode courant plutôt que tension. Cette structure permet aussi, par le passage dans le domaine logarithmique, d'étendre la dynamique des signaux d'entrée et de sortie.

Nous avons également montré que l'utilisation de la méthodologie  $g_m/I$ , combinée avec les caractéristiques de la technologie SOI, permettait d'obtenir des circuits dont les performances en termes de vitesse et de consommation sont tout à fait intéressantes, comparées aux autres réalisations trouvées dans la littérature.

Le circuit décrit dans cet article a été réalisé dans une technologie MOS SOI 2 $\mu$ m et a fait l'objet de mesures validant l'analyse et les performances simulées.

## Références

- [1] Berrou, Glavieux, et Thitimajshima, "Near Shannon-limit error-correcting coding and decoding: turbo-codes" Proc. Of the ICC93, Genève, Suisse, mai 1993, pp. 1064-1070.
- [2] Bahl, Cocke, Jelinek, et Raviv, "Optimal decoding of linear codes for minimizing symbol error rate" IEEE Trans. Inform. Theory, vol. IT-20, pp. 284-287, mars 1974.
- [3] F. Lustenberger, M. Helfenstein, H.-A. Loeliger, F. Tarköy, and G. S. Moschytz "An analog decoding technique for digital codes". Proc. ISCAS '99, Orlando, Florida, May 30 - June 2, 1999, vol. II, pp. 428-431
- [4] C. Schlegel, "Treillis Coding", IEEE Press, New York, 1997.
- [5] B. Gilbert, « A precision four-quadrant multiplier with subnanosecond response, » IEEE J. of Solid State Circuits, vol. SC-3, pp. 353-365, déc. 1968.
- [6] F. Silveira, D. Flandre et P.G.A. Jespers, "A  $g_m/I$  based methodology for the desing of CMOS analog circuits and its application to the synthesis of a SOI micropower OTA" IEEE J. of Solid State Circuits, vol. 31, n. 9, sept. 1996.
- [7] G. Han et E. Sanchez-Sinencio, "Cmos transconductance multipliers : a tutorial" IEEE Trans. on Circuits and Systems, vol. 45, n. 12, pp. 1550-1562, déc. 1998.
- [8] S. Liu et Y. Hwang, "Cmos four-quadrant multiplier using bias feedback techniques" IEEE J. of Solid State Circuits, vol. 29, pp. 750-752, juin 1994.
- [9] H. Song et C. Kim, "An MOS four-quadrant analog multiplier using simple two-input squaring circuits with source followers" IEEE J. of Solid State Circuits, vol. 25, pp. 841-848, june 1990.
- [10] Z. Wang, "A Cmos four-quadrant analog multiplier with single-ended voltage output and improved temperature performance" IEEE J. of Solid State Circuits, vol. 26, pp. 1293-1301, sept. 1991.
- [11] C. Winstead et all "Analog MAP Decoder for (8,4) Hamming Code in Subthreshold CMOS" Proc. Fo the International Symposium on Information Theory, Washington, June 2001
- [12] H.A. Loeliger and F. Tarköy, "Decoding in analog VLSI" IEEE Communication Magazine, April 1999.