THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS VI

SPÉCIALITÉ INFORMATIQUE

Présentée par Arnaud TURIER
Pour obtenir le titre de
DOCTEUR DE L'UNIVERSITÉ PARIS VI

ETUDE, CONCEPTION ET CARACTÉRISATION DE MÉMOIRES CMOS, FAIBLE CONSOMMATION, FAIBLE TENSION EN TECHNOLOGIES SUBMICRONIQUES

Présentée le 13 décembre 2000, devant le jury composé de

Christian PIGUET	Rapporteur
Michel ROBERT	Rapporteur
Amara AMARA	Examinateur
Marc BELLEVILLE	Examinateur
Lotfi BEN AMMAR	Examinateur
Alain GREINER	Examinateur
Andreï VLADIMIRESCU	Examinateur

Résumé

La basse consommation est devenue en quelques années, l'un des enjeux majeurs de la micro-électronique notamment grâce à l'émergence de l'électronique portable grand public : pagers, téléphones, ordinateurs, appareils médicaux. La tendance actuelle de Système sur une seule puce (System On Chip), conduit les concepteurs de circuits à rassembler sur une seule puce, un maximum de composants de différents types dont notamment des mémoires. Les mémoires occupent aujourd'hui une part importante du circuit tant dans sa taille que dans sa consommation totale. Aussi, réduire la consommation des mémoires permettrait de réduire la consommation totale des circuits.

Dans cette thèse, nous présentons une architecture de mémoires de type ROM pour des applications faibles consommation. Cette architecture a été validée sur silicium à travers des technologies $0.5\mu m$ et $0.35\mu m$ pour plusieurs instances de différentes tailles. Nous montrons également comment optimiser la consommation d'une mémoire de type SRAM sans en dégrader les performances au niveau du délai.

Avec la réduction des géométries et l'abaissement des tensions d'alimentation, les courants de fuites prennent une part de plus en plus prépondérante dans la consommation des mémoires. Nous expliquons comment caractériser ces courants et nous présentons une méthode pour les réduire, notamment pour les mémoires de type ROM.

Avec des mémoires de grande capacité, nous avons rencontré le problème de leur simulation électrique. Ainsi, nous présentons une méthode basée sur des générateurs de courant de façon à modéliser les parties redondantes rencontrées dans les mémoires.

Enfin, nous exposons le développement et la réalisation d'un générateur de ROMs utilisant l'architecture présentée auparavant, en présentant les problèmes et les solutions liés à la méthodologie de conception et de validation.

Abstract

In just a few years, low power consumption has become one of the greatest stakes in microelectronics in part thanks to the rise of portable electronic devices such as pagers, cellular phones, computers and medical equipment, all aimed at general public. Today's trend, System On Chip, leads circuit designers to gather as many different kinds of components as possible, memories for instance, on one and same chip. Today, memories take up much of the circuit's area and total power consumption. Also, to reduce memory power consumption will help reduce all of the circuit power consumption.

In this dissertation, we present a ROM architecture for low-power consumption applications. This architecture has been validated on silicon using $0.5\mu m$ and $0.35\mu m$ technologies for instances of different sizes. We also show how to optimize the power consumption of a SRAM without increasing the timings.

With device reduction and power supply lowering, leakage currents hold a more and more prominent place in memory power consumption. We explain how to characterize leakage currents and we present a method to reduce them, in ROMs for example.

As for large capacity memories, we faced the problem of their electrical simulation. This is why we suggest a method based on current generators so as to model the redundant parts found in the memories.

Finally, we show the development and design of a ROM compiler that uses the architecture that was previously presented. We present the problems linked to design and validation flow as well as their keys.

Table des matières

Table des figures

Liste des tableaux

Introduction

UTILISÉE jusqu'à présent avec parcimonie, la mémoire est un composant essentiel dans tous les circuits intégrés, et ce quelqu'en soit le type : ROM, SRAM, DRAM, Flash et Ferro-Magnétique. Deux exemples décrivent bien cette nouvelle tendance : les baladeurs de type MP3 qui s'appuient sur le stockage d'un album musical non pas sur un disque comme auparavant, mais sur l'utilisation de mémoire Flash d'environ 32Mo actuellement. Ceci laisse prévoir une utilisation de ce principe pour des applications encore plus gourmandes en terme de capacité de stockage comme la mémorisation de séquences vidéo par exemple. L'autre exemple est celui de la nouvelle génération de processeurs qui géreront la mémoire d'une manière complètement différente puisqu'elle sera au cœur du système. C'est ce que proposent les projets IRAM (Intelligent RAM) [IRA2000] et SSMP (Stanford Smart Memories Project) [SSM2000] définis respectivement à Berkeley et Stanford. Dans ces projets, la mémoire est intégré sur la même puce que le microprocesseur.

Cette tendance appelée Système sur une Puce (Système on Chip), s'est généralisée sur bon nombre d'ASICs (Application Specific Integrated Circuit), ce qui nécessite de rendre rapidement accessible aux concepteurs de ces circuits, des mémoires clés-en-main qui sont en fait gérées commes des macros avec toutes les vues nécessaires à leur intégration dans le flot de conception. Le temps de mise sur le marché (Time to market) tend à se réduire chaque année un peu plus, il est donc nécessaire de pouvoir adapter rapidement la configuration des mémoires aux nouvelles spécification imposées dans la mise à jour d'un produit. C'est ainsi que les générateurs de mémoires et la capacité d'un circuit mémoire à pouvoir être réutilisé (Design reusability) prend toute son importance. Dans une offre complète de bibliothèque de cellules, ce qui en fait ou non sa richesse, est basée sur l'offre en terme de mémoire : type de mémoires implémentées, facilité d'intégration, densité, temps de cycle et consommation. C'est ce dernier point, la consommation qui retiendra plus particulièrement notre attention. En effet, l'acroissement de l'automonomie des sytèmes portatifs passe par la diminution de la consommation des circuits intégrés, sachant qu'il y a pour le moment, peu à attendre des améliorations sur les batteries. Puisque les mémoires sur une puce en constituent l'essentiel de la surface et du temps d'activation, on voit que si l'on parvient à réduire la consommation des mémoires, on pourra réduire la consommation totale du circuit.

Cette thèse s'inscrit dans le cadre d'une convention CIFRE en partenariat avec la société ATMEL, au sein du département ASIC et plus particulièrement dans le groupe chargé

de la conception des mémoires embarquées en technologie CMOS : ROMs, SRAMs à simple et double port. Notre travail portera sur l'estimation de la consommation, le développement de nouvelles architectures pour la faible consommation, la réduction de la consommation d'architecture existantes et enfin, sur la réalisation de générateurs de mémoires pour la basse consommation.

Dans cette thèse plusieurs aspects de la basse consommation sont abordés : la consommation dynamique à travers la conception de mémoires de type ROM et SRAM basse consommation, et la consommation statique, les courants de fuites, pour trois types de mémoires ROMs, SRAMs à simple et double port. Le but de cette thèse est de permettre le développement de mémoires faible consommation en technologie CMOS dans la perspective de la conception et de la caractérisation de générateurs de mémoires ou bien d'instances à la demande. Le caractère industriel de cette recherche a permis la réalisation sur silicium d'un certain nombre de circuits, qu'il s'agisse de mémoires complètes ou de sous-ensembles conçus pour des caractérisations spécifiques. La thèse ayant duré un peu plus de trois ans, trois technologies CMOS ont pu ainsi être étudiées : $0.5\mu m$, $0.35\mu m$ et $0.25\mu m$.

Dans le chapitre 1, nous présentons les techniques employées actuellement pour la réduction de la consommation dynamique puis pour réduction de la consommation statique dans les mémoires.

Ensuite, nous établissons dans le chapitre 2 une architecture pour la basse consommation des ROMs et nous commentons les résultats obtenus sur silicium. Dans ce chapitre nous montrons les améliorations à apporter afin d'abaisser la consommation d'une architecture de SRAM existante et nous discutons de l'efficacité des méthodes retenues.

Dans le chapitre 3, nous mettons en œuvre une technique de réduction des courants de fuites dans les ROMs à comparer avec les méthodes de polarisation de substrat, puis nous l'appliquons à d'autres types de mémoires.

De manière à étudier précisément et à caractériser les délais et la consommation dynamiques dans les architectures détaillées précédemment, nous présentons dans le chapitre 4 une méthode de caractérisation basée sur l'utilisation de générateurs de courant. Nous montrerons comment l'utiliser sur nos mémoires et nous la comparons aux simulateurs électriques de référence.

Enfin, dans l'annexe nous présentons les différents aspects de la réalisation d'un générateur de mémoire ainsi que la méthodologie retenue pour son utilisation et la validation des instances générées.

CHAPITRE 1

État de l'art

QUELLES sont les sources de consommation statiques et dynamiques ? Comment l'introduction de nouvelles technologies conduit à la diminution de la consommation dynamique mais contribue à l'augmentation de la consommation statique ? Quelles techniques adopter pour réduire la consommation et à quel prix ? Des réponses à ces questions seront apportées au cours de ce chapitre.

1.1 Consommation dynamique

1.1.1 Consommation liée à la charge et à la décharge d'une capacité

La consommation dynamique dans les technologie CMOS apparaît à chaque commutation d'au moins une des entrées d'une porte. Pour la porte représentée à la figure 1.1, l'activation d'une des entrées A_i provoque le passage d'un courant i_c de l'alimentation V_{dd} vers la charge de sortie C_L.

Pendant la charge de la capacité de sortie C_L, l'énergie tirée de l'alimentation est :

$$E_s = \int_{t_0}^{t_1} V i_c(t) dt \tag{1.1}$$

avec $i_c(t) = C_L \frac{dv_c(t)}{dt}$.

Si comme condition initiale on impose $v_c(t_0) = 0$, et qu'à la fin de la charge $v_c(t_1) = V_{dd}$, alors 1.1 devient :

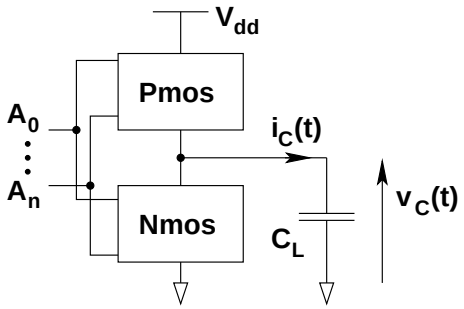


FIG. 1.1 – Modélisation de la consommation dynamique pour les portes CMOS.

$$E_s = C_L V \int_{t_0}^{t_1} \frac{dv_c(t)}{dt} dt = C_L V \int_0^{Vdd} dv_c = C_L V_{dd}^2 \tag{1.2}$$

Une partie de l'énergie est dissipée dans le réseau de PMOS, alors que l'autre est utilisée pour charger la capacité de sortie. A la fin de la charge, l'énergie stockée dans la capacité s'écrit :

$$E_{cap} = \int_{t_0}^{t_1} v_c(t) i_c(t) dt = C_L \int_{t_0}^{t_1} v_c(t) \frac{dv_c(t)}{dt} dt = C_L \int_0^V v_c dv_c = \frac{1}{2} C_L V_{dd}^2 \tag{1.3}$$

Si lors de l'utilisation du circuit, la capacité est chargée puis déchargée successivement à la fréquence f, d'après l'équation 1.2, la puissance dissipée par le circuit s'écrit :

$$P = E_s f = C_L V_{dd}^2 f \tag{1.4}$$

1.1.2 Consommation liée au courant de court-circuit

Lors de la commutation d'une porte CMOS, il arrive un moment où les transistors des réseaux PMOS et NMOS (Figure 1.1) sont passant en même temps ce qui crée un courant de court-circuit entre l'alimentation et la masse. Au premier ordre, la variation du courant de court-circuit dans le temps, pendant la transition du signal en entrée, est donnée, pour un inverseur, à la figure 1.2.

Lorsque le signal d'entrée est inférieur à V_{tn} ou bien supérieur à V_{tp}, le courant de court-circuit est nul. Il augmente lorsque la tension d'entrée dépasse V_{tn} et diminue au fur et à mesure que la tension d'entrée se rapproche de V_{tp}.

Dans ce cas, l'énergie consommée par ce courant de court-circuit est égale à [Vee1984] :

$$E_{cc} = \frac{\beta}{12} \tau (V_{tp} - V_{tn})^3 \tag{1.5}$$

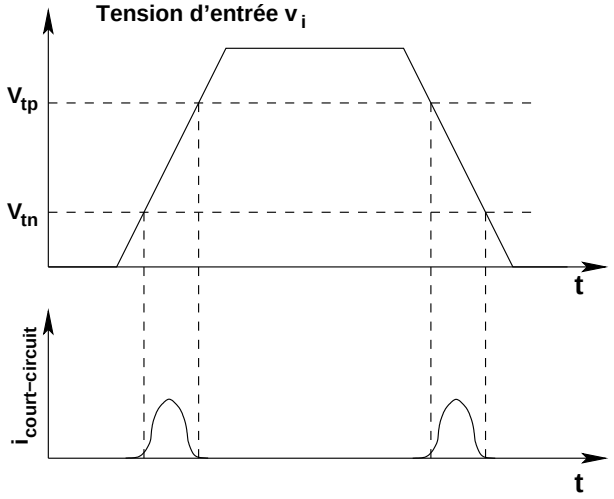


FIG. 1.2 – Courant de court-circuit d'un inverseur pendant la transistion du signal d'entrée.

Avec β, la taille des transistors considérée comme identique, τ, les temps de montée et de descente eux aussi considérés comme égaux et V_{tp} et V_{tn} les tensions de seuil respectives des transistors PMOS et NMOS. En réalité, l'équation 1.5 est plus complexe puisqu'elle est donnée ici, pour une charge nulle en sortie : le courant de court-circuit dépendat en effet de la durée et de la pente du signal d'entrée, des caractéristiques des transistors et de la charge de sortie. De manière à pouvoir négliger l'énergie de court-circuit il est recommandé d'avoir des temps de montée et de descente rapides pour les entrées et les sorties. Cependant, si la sortie est fortement chargée, le temps de montée ou de descente du signal de sortie sera grand devant celui du signal d'entrée : ainsi, les entrées ont le temps de changer d'état avant que la sortie commute totalement, ce qui rend le courant de court-circuit négligeable.

1.1.3 Réduction de la consommation dynamique

Cette réduction passe par :

- L'abaissement de la tension d'alimentation : étant donné que la tension est un terme quadratique dans l'expression de la puissance (Eq. 1.4), il est important de la diminuer. Cependant, cela a pour effet de ralentir la vitesse du circuit pour une technologie donnée. Si en revanche, cette réduction de tension s'accompagne d'une diminution des géométries (Changement de technologie), les délais ne seront pas réduits. Les tableaux 1.1 et 1.2 montrent l'abaissement de la tension d'alimentation des circuits respectivement pour les technologies actuelles et pour les technologies à venir.
- La diminution de la fréquence d'activation : au niveau architectural, on réduit la fréquence d'accès en partitionnant le circuit en blocs et n'activant que ceux nécessaires à l'opération que l'on veut réaliser. Cette réduction passe aussi par le conditionne-

Technologie (μm)	0.7	0.5	0.35	0.25	0.18
Tension d'alimentation (V)	5.0	3.3	3.3	2.5	1.8

TAB. 1.1 – Tensions d'alimentation usuelles en fonction de la technologie.

Technologie (μm)	0.13	0.10	0.07	0.05	0.035
Année	2002	2005	2008	2011	2014
Tension d'alimentation (V)	1.5	1.2	0.9	0.6	0.6

TAB. 1.2 – Tensions d'alimentation pour les technologies à venir [ITR1999].

ment de portes (Gated clock).
– La diminution des capacités de sortie : au niveau du layout, on dessine des drains les plus petits possible pour une largeur de transistor donnée. Au niveau du dimensionnement des portes, on choisit des tailles petites pour la vitesse souhaitée : on réduit ainsi les capacités de grille et de drain. Enfin, au niveau architectural, la hiérarchisation permet de réduire les capacités qui commutent lorsque le circuit est activé.

1.2 Sources de la consommation statique

Jusqu'à présent la puissance consommée liée aux courants de fuite a été négligée dans les études globales de puissance puisque elle était quantitativement faible vis-à-vis des autres sources de consommation. Avec l'apparition conjointe des nouvelles technologies submicroniques, et des dispositifs embarqués, alimentés sur batteries, les courants de fuite deviennent une source de préoccupations majeures. Il s'agit là d'un nouveau défi, et ce particulièrement pour le design des mémoires [Ito1999]. Les courants de fuite deviennent critiques quand le circuit passe beaucoup de temps en mode repos ou bien lorsque son activité dynamique est faible. Si en revanche, si toutes les parties d'un circuit sont constamment activées, la consommation statique reste faible devant la consommation dynamique.

Les courants de fuite peuvent se décomposer en cinq catégories [Neb1997, page 97] :

– Le courant de polarisation de diode en inverse (Reverse biased pn junction current).
– Le courant sous le seuil (Subthreshold current).
– Courant de Drain Induit par la Grille (Gate Induced Drain Leakage (GIDL)).
– Le courant de perçage (Drain source punch through current).
– Le courant à travers la grille (Gate tunelling current).

Cette taxinomie n'est pas définitive car certains effets électriques submicroniques sont découverts au fur et à mesure de l'apparition de nouvelles technologies aux dimensions encore plus réduites.

1.2.1 Courant de polarisation de diode en inverse

Il existe dans les diodes formées entre le substrat et l'implant, un courant de polarisation inverse, comme le montre le schéma de la figure 1.3.

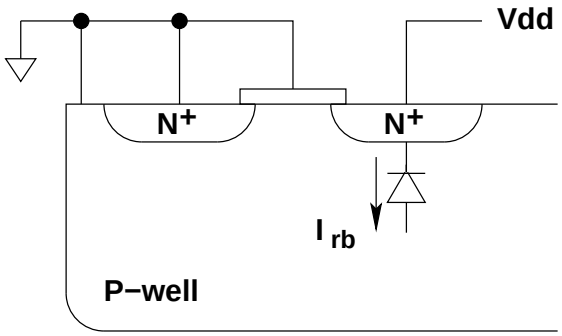


FIG. 1.3 – Courants de polarisation inverse à travers un transistor N.

L'expression du courant direct, à travers une diode est donné, au premier ordre, par [Cha1995, page 98] :

$$I_D = I_S \left(\exp^{\frac{V_{DS}}{\phi_t}} - 1 \right) \tag{1.6}$$

avec

I_S le courant de saturation.

V_{DS} la tension de polarisation.

ϕ_t la tension thermique.

On a $\phi_t = \frac{kT}{q}$, avec la constante de Boltzmann $k = 1.38 \times 10^{-23}\, J/°K$, la température de jonction $T\,[K]$, et la charge d'un électron $q = 1.6 \times 10^{-19}\, C$. Si les diodes sont polarisées en inverse, la tension de polarisation V_{DS} est négative. D'après les valeurs prises par V_{DS} et par T (tableau 1.3), le terme exponentiel devient négligeable devant 1.

Temp.$[°C]$	-55	0	25	40	85	100	125
$\phi_t\,[mV]$	18.82	23.56	25.72	27.01	30.89	32.18	34.34

TAB. 1.3 – Valeurs prises par ϕ_t dans une gamme de températures usuelles.

Ainsi, l'équation 1.6 devient :

$$I_{rb} = -I_S \tag{1.7}$$

Le courant de polarisation inverse peut aussi s'exprimer de la façon suivante :

$$I_{rb} = A_{D/S} \bullet J_{S_{D/S}} + P_{D/S} \bullet I_{P_{D/S}} \tag{1.8}$$

Avec $A_{D/S}$ l'aire de drain/source du transistor, $P_{D/S}$ le périmètre du transistor et J_S la densité de courant par unité de surface et I_P la contribution périmétrique. D'après [Cha1995, page 99], pour un process $1.2\mu m$, cas typique, à $25°C$, J_S a une valeur d'environ $1 - 5\, pA/\mu m^2$, cette valeur doublant à chaque fois que la température augmente de $9°C$.

En conclusion, le courant de polarisation inverse est indépendant de la tension d'alimentation V_{DS}. En revanche, il dépend des géométries des transistors.

1.2.2 Courant sous le seuil

Le courant sous le seuil est un courant qui circule entre la source et le drain du transistor alors que la tension V_{GS} est inférieure à la tension de seuil V_t, comme décrit sur la figure 1.4.

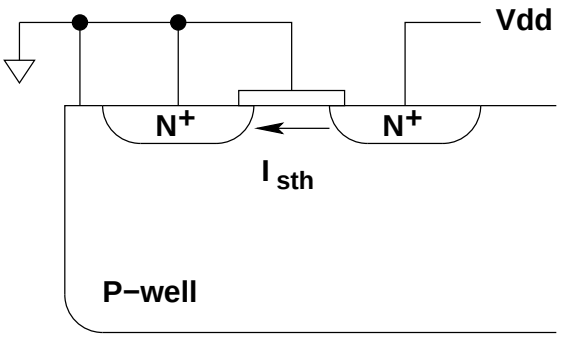


FIG. 1.4 – Courant sous le seuil dans un transistor N.

Le comportement de ce courant a évolué au fur et à mesure de l'apparition des technologies submicroniques, c'est pourquoi, il convient de distinguer 2 cas selon la largeur du canal du transistor.

Canaux longs (Approximation au 1er ordre)

Dans une approximation au premier ordre, le courant sous le seuil nous est donné par l'équation suivante [Cha1995, p 99] :

$$I_{DS} = K \exp^{\left(\frac{V_{GS}-V_t}{n\phi_t}\right)} \left(1 - \exp^{-\frac{V_{DS}}{\phi_t}}\right) \tag{1.9}$$

avec $K = \mu_0 C_{ox} \frac{W}{L} \phi_t^2$, V_t la tension de seuil et n paramètre technologique : $n = 1 + \epsilon_{si}/\epsilon_{ox} \times t_{ox}/D$. t_{ox} étant l'épaisseur d'oxyde et D la largeur de déplétion du canal, ce

qui revient à écrire : $n = 1 + C_D/C_{ox}$, avec C_D, la capacité de la couche déplétée et C_{ox}, la capacité d'oxyde.

Comme pour le courant de polarisation inverse, V_{DS} est grand devant ϕ_t, ainsi, $\exp^{-\frac{V_{DS}}{\phi_t}} \simeq 0$. Ainsi 1.9 s'écrit :

$$I_{sth} = \mu_0 C_{ox} \frac{W}{L} \phi_t^2 \exp^{\frac{V_{GS}-V_t}{n\phi_t}} \tag{1.10}$$

De plus, le transistor n'est pas passant et $V_{GS} = 0$, 1.10 devient :

$$I_{sth} = \mu_0 C_{ox} \frac{W}{L} \left(\frac{kT}{q}\right)^2 \exp^{-\frac{qV_t}{nkT}}$$

La caractérisation du courant sous le seuil est représentée par l'expression de $\log(I_{sth})$ en fonction de V_{GS} (figure 1.5).

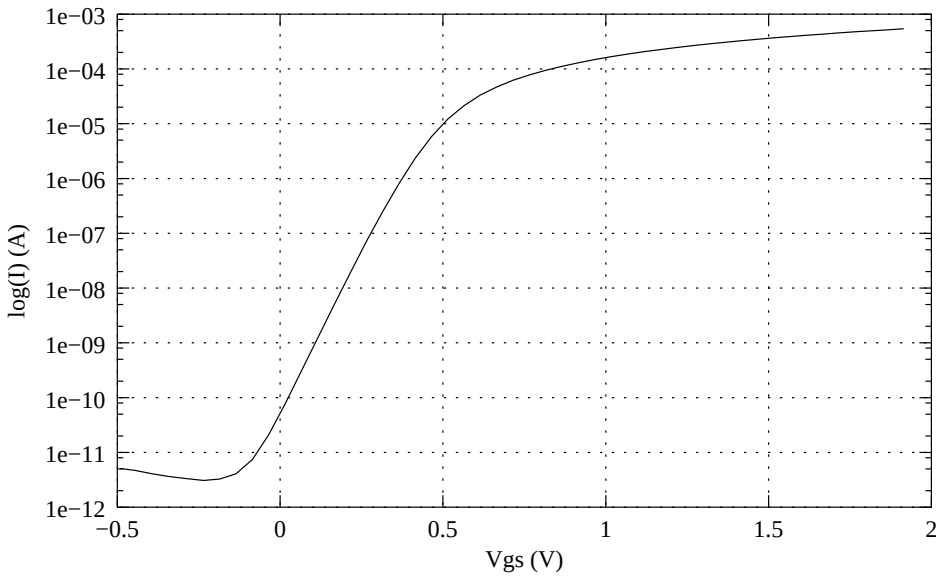


FIG. 1.5 – Caractéristique du courant $I(V_{gs})$.

On a l'habitude de mesurer la valeur de la pente (S_{sth}) de la partie linéaire de la courbe, correspondant à la conduction sous la tension de seuil.

Si on exprime le rapport de 2 intensités prises dans la partie linéaire, on a :

$$S_{sth} = n\ \phi_t\ \ln(10) \tag{1.11}$$

Ou encore [Bel1995, page 86] :

$$S_{sth} = \frac{kT}{q} \ln(10) \cdot \left(1 + \frac{C_D}{C_{ox}}\right)$$

D'après [Cha1995, page 100], S_{sth} prends des valeurs comprises entre 60 et $90\ mV/dec$. Plus S_{sth} est faible, plus le courant de fuite sera faible. La valeur limite de la pente peut être déterminée en considérant un procédé de fabrication SOI, pour lequel le rapport est proche de 0. Ainsi, la valeur minimale de S_{sth} est égale à $\frac{kT}{q}\ln(10) \simeq 60mV/dec$ à température ambiante.

En conclusion, le courant sous le seuil est indépendant de la tension d'alimentation V_{DS} et la pente S_{sth}qui le caractérise possède une valeur minimale qu'il est impossible de faire varier pour diminuer la valeur du courant de fuite. De plus, il faut noter qu'à W et L égaux, les transistors PMOS présentent une pente plus importante que les transistors NMOS.

Pour résumer, au premier ordre, l'évolution du courant sous le seuil varie :

- Linéairement avec W.
- Exponentiellement avec T.
- De manière inversement proportionnelle quand L diminue.
- De manière inversement proportionnelle quand l'épaisseur d'oxyde diminue.
- Exponentiellement quand V_t diminue.

Canaux courts (Short Channel Effect)

Pour les géométries à canaux courts (géométries submicroniques), l'effet de Drain Induced Barrier Lowering (DIBL) intervient. L'équation 1.9, devient :

$$I_{DS} = K \exp^{\left(\frac{V_{GS}-V_t+\eta V_{DS}}{n\phi_t}\right)} \left(1 - \exp^{-\frac{V_{DS}}{\phi_t}}\right) \tag{1.12}$$

Pour des considérations identiques de la valeur de V_{DS} par rapport à ϕ_t, 1.12, s'écrit :

$$I_{DS} = K \exp^{\left(\frac{V_{GS}-V_t+\eta V_{DS}}{n\phi_t}\right)}$$

Quand $V_{GS} = 0$, l'équation devient :

$$I_{DS} = K \exp^{\left(\frac{\eta V_{DS}-V_t}{n\phi_t}\right)} \tag{1.13}$$

Le facteur ηV_{DS} a pour effet de diminuer la valeur de $-V_t$, ce qui est équivalent à une réduction de V_t dans l'expression du courant de fuite sous le seuil [Dav1995], ce qui revient à décaler la courbe 1.5 vers la gauche (figure 1.6).

En réalité l'équation 1.13 est beaucoup plus complexe, puisque la tension de seuil V_t dépend notamment des paramètres W, L et température. Aussi, la dépendance linéaire du courant sous le seuil en fonction de W, évoquée à la section précédente, est remise

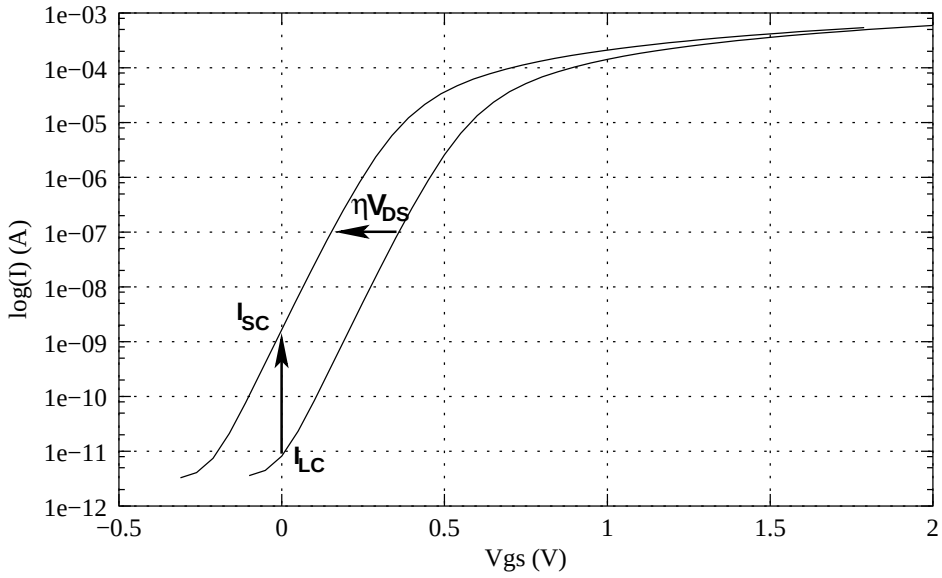


FIG. 1.6 – Effet canal court sur le courant sous le seuil

en cause. Pour les technologies submicroniques et notamment à partir du $0.25\mu m$, à W minimum, la fuite n'est pas minimale comme le montre la figure 1.7 : au fur et à mesure que l'on se rapproche de la largeur minimale du transistor, le courant de fuite s'éloigne de l'asymptote.

1.2.3 Autres courants

- Courant de perçage. Ce courant apparaît entre la source et le drain du transistor notamment pour les technologies submicroniques puisque la distance entre le drain et la source diminue. Le courant de perçage est dépendant de L et pour reduire son effet, on augmente le dopage du canal, ce qui augmente la valeur de V_t.
- Courant de drain induit par la grille (Gate Induced Drain Leakage).
  C'est un courant de fuite provoqué par un champ électrique de valeur élevée à l'endroit où la grille recouvre le drain.

1.2.4 Réduction des courants de fuite

La diminution de la tension d'alimentation, qui conduit à la réduction de la consommation dynamique, a obligé les ingénieurs de procédés de fabrication à créer des technologies avec une tension de seuil plus basse que d'ordinaire ($0.7V$) afin de ne pas limiter en terme de vitesse des circuits alimentés par une faible tension d'alimentation. Cet effort est cependant néfaste sur les courants de fuite, notamment pour le courant sous

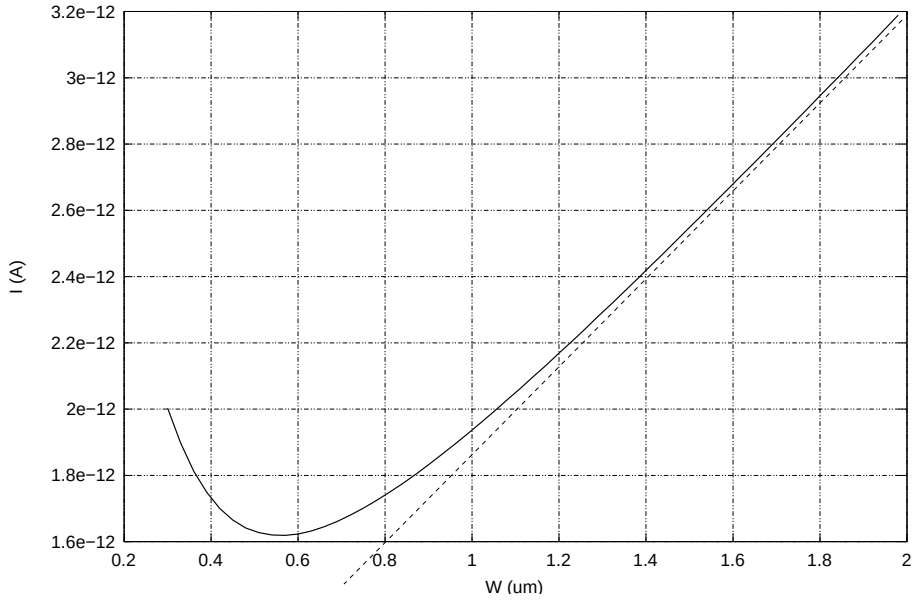


FIG. 1.7 – Dépendance de $V_t(W)$ sur la courbe $I(W)$ pour des technologies fortement submicroniques

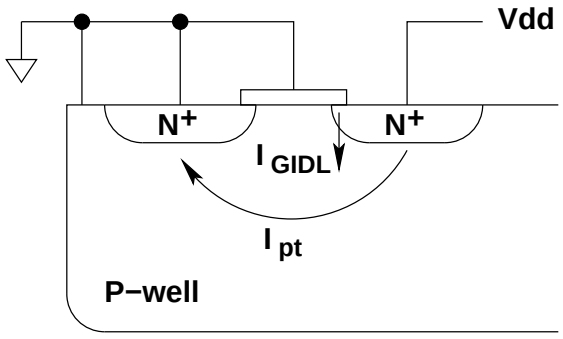


FIG. 1.8 – Courants de perçage et de drain induit par la grille (GIDL)

le seuil, puisque, comme nous l'avons vu à la section précédente, il dépend exponentiellement de la valeur de V_t et de manière inversement proportionnelle à la longueur du canal. L'apparition des nouvelles technologies submicroniques est défavorable aux courants de fuite. Dans cette section nous présentons différentes techniques pour diminuer les courants de fuite et notamment le plus important en valeur, le courant sous le seuil. L'équation 1.10 nous montre les paramètres sur lesquels on peut agir pour diminuer le courant sous le seuil : réduction de W, augmentation de L, mais ce, au détriment de la vitesse et de la surface, ce qui n'est donc pas forcément intéressant. Sachant que la pente sous le seuil connaît un minimum il apparaît que le meilleur moyen pour diminuer le courant sous le seuil est de décaler la courbe 1.5 vers la droite du graphique. Pour cela, les moyens d'action reposent sur la tension V_{GS} et sur la tension de seuil V_t qui, au premier ordre, s'écrit [Rab1996, page 43] :

$$V_T = V_{T0} + \gamma \left(\sqrt{|-2\phi_F + V_{SB}|} - \sqrt{|-2\phi_F|} \right) \tag{1.14}$$

Si dans le cas classique on a $V_{GS} = 0$ et $V_{SB} = 0$, nous verrons que les solutions mises en œuvre, reposent sur une polarisation non classique de la source et du substrat ce qui conduit à une modification du courant sous le seuil.

1.2.5 Polarisation inverse de V_{gs}

Cette approche consiste à modifier, en mode repos, la tension V_{gs} des transistors bloqués et ne peut donc être appliquée qu'à des circuits dont les sorties ont un état prédictible en mode repos. Pour les transistors NMOS de la figure 1.9, la tension V_{gs} est négative ($V_{gs} = -\delta$) ce qui polarise de manière inverse la grille du transistor puisque d'habitude, cette tension est toujours supérieure ou égale à zéro. De même pour les transistors PMOS, la tension V_{gs} est positive ($V_{gs} = +\delta$) alors qu'elle est toujours négative ou nulle.

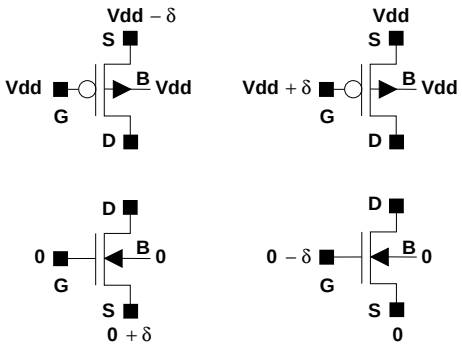


FIG. 1.9 – Polarisation inverse de V_{GS}. [Ito1999]

Cette polarisation inverse possède une double action sur le courant sous le seuil des transistors comme l'indique la figure 1.10 pour un transistor de type NMOS. Dans un

premier temps, $V_{SB} \neq 0$, ce qui, d'après 1.14, augmente la valeur de V_T. La courbe représentant le courant sous le seuil se déplace vers la droite et la valeur du courant passe de I_{L0} à I_{L1}. Enfin, V_{GS} étant négative, cela revient à se déplacer sur la pente sous le seuil vers la gauche du graphe de I_{L1} vers I_{L2}.

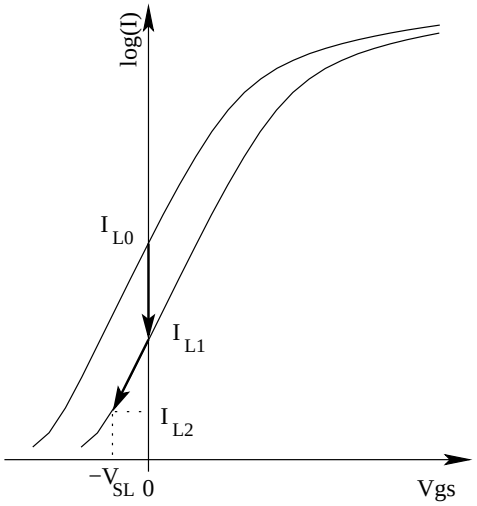


FIG. 1.10 – Évolution du courant de fuite en polarisation inverse de V_{GS}.

Impédance de Source Commutée (Switched-Source-Impedance)

Cette technique [Hor1993] consiste à modifier la tension de source des transistors pendant le mode de repos des mémoires en rajoutant une résistance entre la source du transistor qui fuit et l'alimentation. Cette résistance provoque un changement de la tension sur la source du transistor. Afin de ne pas pénaliser le circuit pendant son mode de fonctionnement, cette résistance peut être court-circuitée, ce qui place le circuit dans un cas classique. Cette technique est utilisée dans des parties logiques dont les états sont prédictibles. Elle est représentée à la figure 1.11 pour un inverseur dont la sortie est au niveau haut, le transistor PMOS étant saturé et le transistor NMOS bloqué et fuyant. La résistance est composée de polysilicium et le switch par un transistor possédant une tension de seuil élevée de manière à ne pas court-circuiter la résistance en mode repos par des courants de fuite éventuels à travers ce transistor. Cette méthode implique l'utilisation d'un procédé de fabrication avec double V_T. Bien que les auteurs s'attachent à montrer que leur idée est applicable à toutes les portes logiques, il faut noter que le surcoût en terme de taille et de circuiterie de commande pour le switch n'est vraisemblablement intéressant que pour de gros transistors (Le courant sous le seuil étant directement proportionnel à W) dont l'état de sortie est prédictible, c'est-à-dire principalement des inverseurs de grande taille utilisés pour piloter les lignes de mots sur toute la longueur d'une mémoire.

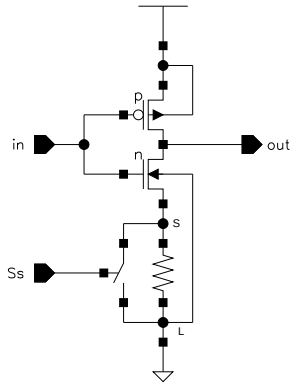


FIG. 1.11 – Polarisation de V_{GS} par une impédance de source commutée.

Polarisation auto-inversée (Self-Reverse Biasing)

En s'inspirant de la méthode précédente, on peut imaginer construire la résistance et le switch avec un seul transistor possédant une tension de seuil identique aux autres transistors du circuit. Lorsqu'il n'est pas passant, en mode repos, le courant qui le traverse peut être assimilé à celui qui traversait la résistance dans le paragraphe précédent. Cette technique est appelée polarisation auto-inversée [Kaw1993] et est décrite à la figure 1.12. Le fait que le transistor M_C soit coupé, une tension inférieure à la tension d'alimentation V_{dd} apparaît sur le nœud V_{C1} à travers les courants de fuite traversant les transistors M_C et M_D. Dans cet exemple, les transistors M_D fuient car un niveau haut est positionné sur leur grille, avec une tension de source inférieure à V_{dd}. On se retrouve alors dans le premier cas de la figure 1.9, le courant sous le seuil est donc réduit par cette technique. Le dimensionnement du transistor M_C doit être soigneusement réalisé de manière à ce qu'il laisse passer suffisamment de courant lors du passage du mode repos au mode actif de façon à ce que le nœud V_{C1} soit de nouveau à V_{dd} le plus rapidement possible. Cependant, plus W_C est petit, plus la réduction du courant de fuite sera importante, il y a donc un compromis à trouver comme l'expliquent les auteurs de cette idée.

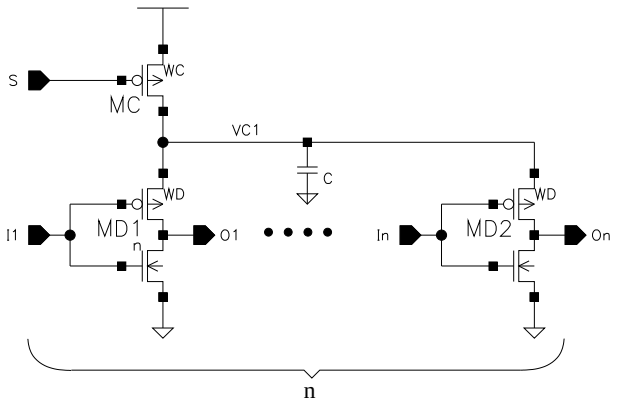


FIG. 1.12 – Polarisation auto-inversée

Alimentation hiérarchique (hierarchical power-line scheme)

Les architectures de mémoires de tailles importantes font appel à la division de la mémoire en blocs, où un bloc à la fois est activé pendant que les autres sont inactifs. Ainsi, alors que la mémoire est activée, on a quand même une consommation statique, par un courant de fuite dans les blocs non activés. L'idée présentée dans le paragraphe précédent peut être améliorée [Sak1994] en introduisant une hiérarchie dans les transistors qui servent à contrôler les courants de fuite.

Inverseur à alimentation commutée avec compensateur de niveau (Switched-Power-Supply inverter with Level Holder)

Des circuits à la complexité plus grande, permettent de réduire les courants de fuite toujours pour de gros drivers dont, cette fois-ci, la sortie ne pourrait plus être prédéterminée (Dans les parties liées au décodage notamment). Ces circuits [Sak1994] font appel à des transistors à V_T élevé comme le montre le schéma de la figure 1.13. En mode repos, les deux transistors d'alimentation commandés par Sp et Sn sont coupés, le niveau logique en sortie de l'inverseur est donc flottant. Pour compenser ce problème, un latch composé de transistors à V_T élevés est utilisé pour garantir un niveau logique franc sur la sortie de l'inverseur.

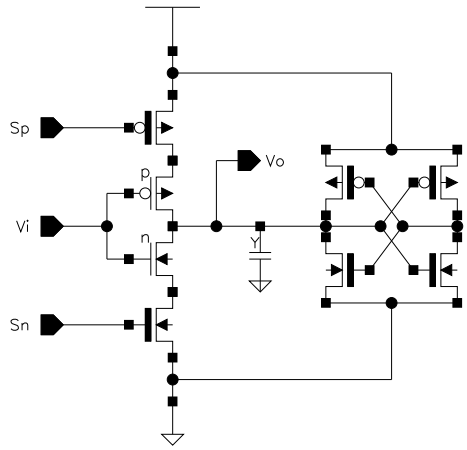


FIG. 1.13 – Inverseur à alimentation commutée avec un compensateur de niveau en sortie.

Super cut-off CMOS (SCCMOS)

Le problème majeur de la polarisation inverse de V_{gs} reste la rétention de l'information pour des portes dont on ne peut prévoir l'état de sortie. Pour cela, une solution

plutôt surprenante est proposée par Kawaguchi et al,[Kaw1998b] : un chemin de scan à travers un circuit combinatoire est utilisé pour stocker le niveau des nœuds dans une mémoire SRAM avec transistor à V_T élevés, avant le passage en mode repos. On notera que le testchip qui a servi aux mesures de courant ne comporte que des bascules, et que la faisabilité d'un circuit complet avec cette technique n'est pas évoquée.

1.2.6 Polarisation inverse de V_{bs}

Cette approche consiste à modifier, en mode repos, la tension V_{bs} des transistors bloqués tout en maintenant à zéro la tension V_{gs}. Par rapport à l'approche précédente, cela revient à se déplacer uniquement du point I_{L0} au point I_{L1} de la figure 1.10, puisque par rapport à l'équation 1.14, la valeur de V_T augmente et la valeur de V_{gs} reste nulle. Les différentes configurations de polarisation d'un transistor dans ce cas, sont résumées à la figure 1.14. Les transistors présentés dans la partie droite de la figure (c et d), peuvent à la fois , être appliqués sur des circuits en mode actif, ainsi que sur des circuits en mode repos dont la sortie est indéterminée, ce qui constitue un avantage par rapport à la polarisation inverse de V_{gs}. La mise en œuvre de cette technique repose sur l'utilisation de technologies triple-puits (triple-well) pour la polarisation du substrat des transistors NMOS.

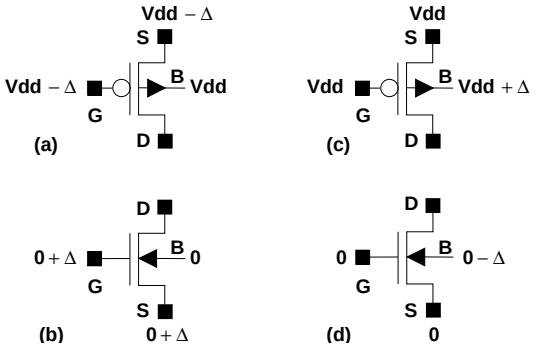


FIG. 1.14 – Polarisation du substrat[Ito1999]

Système de suppression des courants de fuite (Subthreshold leakage Current Supression System (SCSS))

Afin de réduire la consommation en mode actif, Hasegawa et al. [Has1998], utilisent une polarisation du substrat en choisissant les schémas c et d de la figure 1.14. Il faut noter que la vitesse du circuit est pénalisée puisque la tension V_T des transistors est ainsi augmentée. Cependant, si cette technique est suffisante pour réduire les courants de fuite des transistors NMOS à la fois en mode actif et repos, elle ne l'est pas pour les

transistors PMOS, dont la longueur de grille cumulée est supérieure à celle des NMOS dans cet exemple. De plus, la pente sous le seuil des transistors PMOS est supérieure à celle des transistors NMOS. (Cf 1.2.2 page 10). La technique d'impédance de source commutée (SSI) est alors utilisée en complément en mode repos, ce qui a pour effet d'accentuer l'augmentation de V_T en élevant la valeur de V_{SB} (Cf. 1.14). Le coût de cette méthode est l'utilisation de plusieurs lignes d'alimentation, ce qui rend le circuit assez complexe à réaliser, comme le montre la figure 1.15.

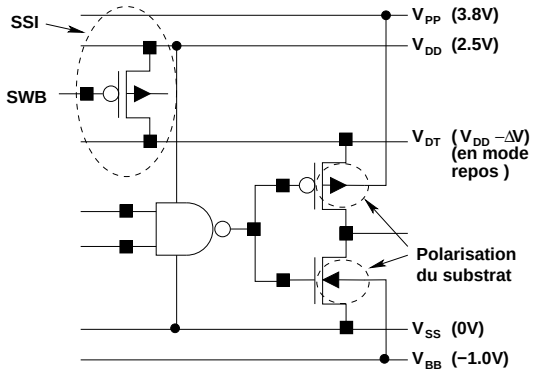


FIG. 1.15 – Système de suppression des courants de fuite (SCSS)

Impédance de substrat commutée (Switched substrate-impedance)

Mizuno et al. [Miz1999] introduisent une nouvelle méthode de polarisation du substrat de manière à éviter que le bruit de l'alimentation et celui du substrat, ne viennent perturber le fonctionnement des transistors. La tension source-substrat, qui est une source de bruit pour le substrat, est réduite pendant le mode actif en rendant passant les transistors à haut V_t des cellules de commutation (Fig. 1.16). En mode repos, ces cellules sont désactivée, ce qui crée une différence de potentiel entre la source et le substrat. Dans cet article on remarque que les temps de passage d'un mode à l'autre (Entre mode repos et mode actif) sont relativement long : le passage du mode actif au mode repos se fait en $50\mu s$, et le retour au mode actif s'effectue en $370ns$. Cette technique nécessite un procédé de fabrication triple-puits et la génération de tension multiples stables ($-2.3V$, $-1.5V$, $-1.8V$, $-1.0V$, $1.8V$, $3.3V$) ce qui paraît difficilement réalisable pour des mémoires embarquées pour ASICs.

Application aux points mémoire de SRAMs

Afin de limiter les courants de fuites dans les points mémoire de type SRAM, Kawaguchi et al. [Kaw1998a] proposent de polariser le substrat des transistors du point mémoire quand ceux-ci ne sont pas activés. Au lieu d'utiliser un signal de mise au

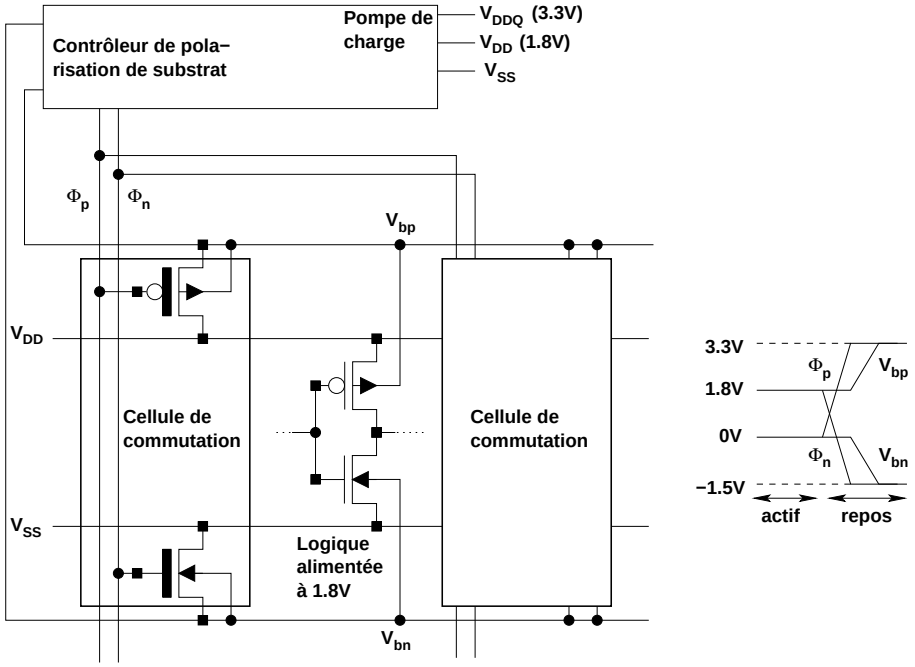


FIG. 1.16 – Impédance de substrat commutée (Switched substrate-impedance)

repos de la mémoire qui commanderait la polarisation, les signaux de ligne de mot, issus du décodeur d'adresse, sont mis à contribution pour détecter les lignes de mots qui ne sont pas activées (Fig. 1.17). Ici, le nombre de tensions à générer est moindre : $-V_{dd}$, V_{dd} et $2V_{dd}$, ce qui rend cette technique plus facile à mettre en œuvre que la précédente. Cependant, les auteurs signalent qu'un basculement inopiné du point mémoire peut être observé lors du passage d'un mode à l'autre (Actif / Repos).

1.2.7 Techniques multi-V_t

Tensions de seuil multiples (MTCMOS : Multiple Threshold CMOS)

La technique des tensions de seuil multiples consiste à disposer de deux tensions de seuil pour chaque type de transistor NMOS et PMOS. Les transistors à V_t élevé sont insérés entre les alimentations et le circuit (Fig. 1.18). En mode repos, ces transistors sont coupés. Comme leur tension de seuil est élevée, le courant de fuite qui circule à l'intérieur est très faible comparé à celui qui circulerait si les transistors du circuit, à V_t standard étaient directement connectés aux alimentations. Les principaux inconvénients sont : la place occupée par les transistors à V_t élevé (Ils doivent être suffisamment gros pour permettre l'alimentation du circuit), le temps de passage du mode repos au mode actif à cause des capacités importantes sur les lignes d'alimentation virtuelles et enfin, en mode repos, l'information stockée dans les nœuds est

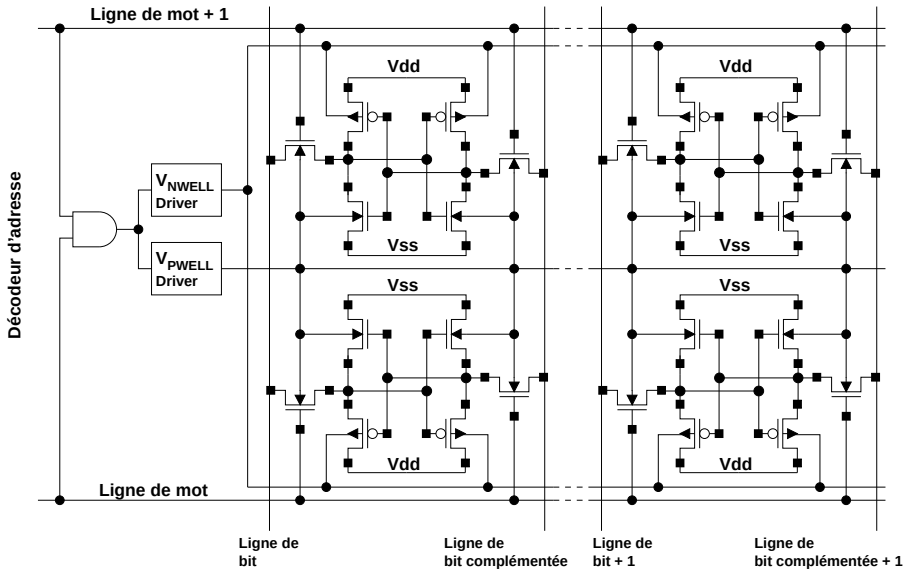


FIG. 1.17 – Diminution dynamique du courant de fuite dans les SRAMs

perdue. Le principe est exposé par Mutoh et al. [Mut1995] et appliqué notamment à un DSP [Mut1996]. Le problème de la conservation de l'information est discuté dans [Aka1996] : un circuit est ajouté de manière à rafraîchir périodiquement les lignes d'alimentation virtuelles ce qui évite la perte d'information. Cependant, rien n'est dit sur la façon de déterminer automatiquement la bonne fréquence de rafraîchissement. Enfin, des considérations d'ordre technologiques sur l'utilisation du SOI pour cette technique sont exposées dans [Dou1997] [Shi1998].

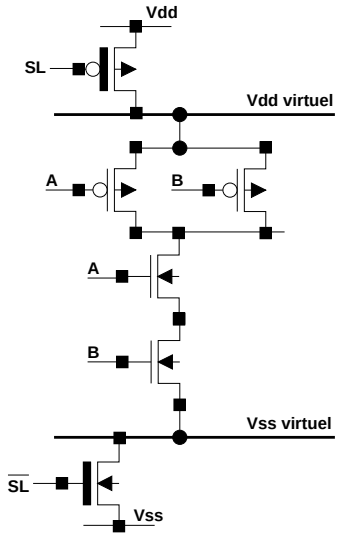


FIG. 1.18 – Tensions de seuil multiples (MTCMOS) appliquées à une porte ET

Tensions de seuil variables (VTCMOS : Variable Threshold CMOS)

L'un des problème de la basse tension est la sensibilité du délai par rapport à la tension de seuil des transistors [Sai1996]. Afin de compenser les fluctuation de V_t, on en fait varier dynamiquement la valeur en polarisant le substrat. De plus, en mode repos, on diminue fortement le courant de fuite sous le seuil. Cette technique à notamment été appliquée à un microprocesseur [Kur1996]. Son inconvénient réside dans la nécessité de technologies triple puits et dans la génération de tensions multiples (Fig. 1.19).

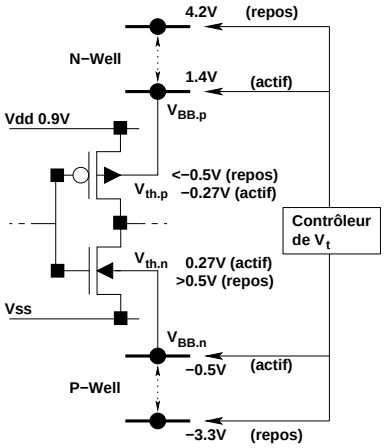


FIG. 1.19 – Tensions de seuil variables VTCMOS appliquées à un inverseur

1.3 La consommation dans les mémoires

1.3.1 Généralités

L'architecture basique d'une mémoire (Figure 1.20) se compose d'un bloc de contrôle pour la synchronisation des signaux de commande, d'un décodeur de lignes, d'un plan mémoire qui contient les points mémoire où est stocké le contenu de chaque bit, et d'un chemin de données qui comprend un circuit de précharge, un multiplexeur, un amplificateur de lecture, parfois un deuxième étage de multiplexeur, un latch pour mémoriser les données lues et un buffer de sortie [Rab1996, page 552], [Bel1995, page 314]. On trouve aussi quelque fois un partage de la mémoire en plusieurs blocs [Sug1993].

Ligne de mot divisée

Pour des mémoires de grande taille, le nombre de points mémoire à activer devient important et augmente le délai et la consommation. Ainsi, une technique consiste à

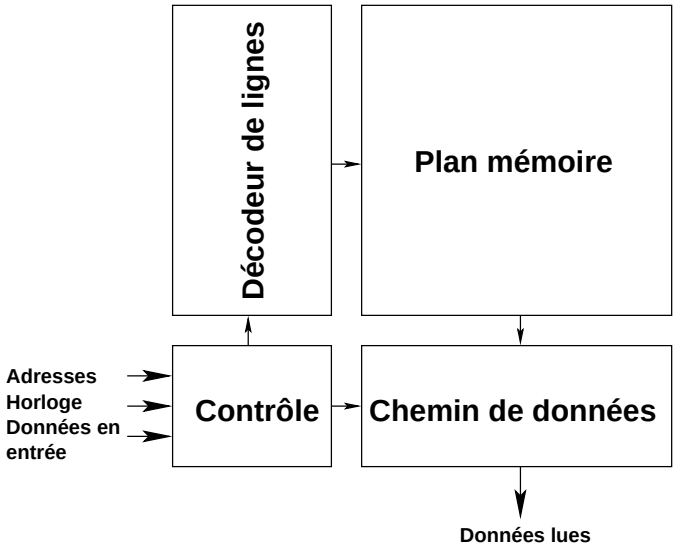


FIG. 1.20 – Schéma bloc d'une mémoire

diviser la mémoire en blocs de façon à découper la ligne de mot en lignes de mot locales commandées par une ligne de mot globale et un signal d'activation de bloc (Figure 1.21) [Yos1983].

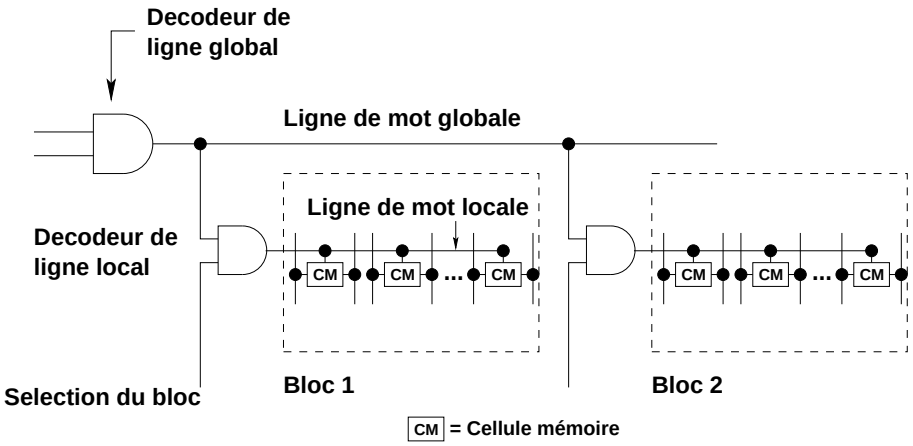


FIG. 1.21 – Ligne de mot divisée

Il faut veiller à ce que le nombre de décodeurs locaux vus par le décodeur global ne soit pas trop important sinon la division de la ligne de mot perd de son intérêt. Cette technique a ensuite été étendue au concept de ligne de mot hiérarchique [Hir1990] de façon à réduire davantage de délai et la consommation pour un plus grand nombre de blocs. Cependant, avec les nouvelles technologies $0.25\mu m$ et $0.18\mu m$ notamment, les capacités de drain et de grille deviennent petites devant les capacités de routage. Cette technique est-elle obsolète ? Non, mais elle n'est efficace, dans la réduction de la capacité de la ligne de mot, qu'à partir d'un nombre de blocs plus grand que pour les technologies passées. Mais là où cette technique reste très intéressante, c'est dans

la réduction du nombre de lignes de bit qui commutent au moment de l'activation du signal de sélection de bloc. Ainsi, il est probable que cette méthode sera encore utilisée à l'avenir, non pas pour réduire la consommation du décodeur de ligne, mais pour réduire le nombre de lignes de bit activées.

Dispositifs de précharge

On distingue principalement deux dispositifs de précharge :

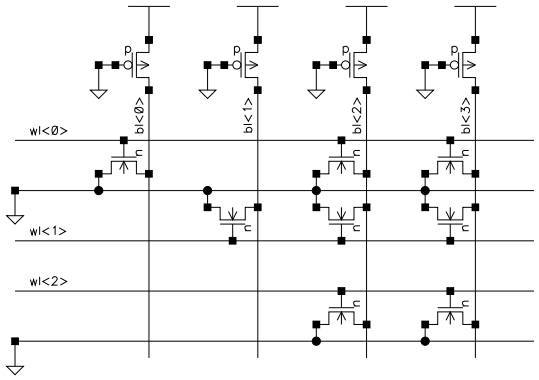
- La précharge par un pull-up : un transistor maintient un niveau sur la ligne de bit en permanence. Le temps pour atteindre la tension d'alimentation, Vdd, après une lecture est réduit, puisque l'excursion de tension sur la ligne de bit est faible. L'inconvénient de ce dispositif est le courant qui circule entre l'alimentation et la masse pendant la lecture. Il est évident que ce type de circuiterie ne convient pas aux applications basse consommation [Rab1996, page 562].
- La précharge activée de manière partielle : les lignes de bits sont préchargées en dehors des phases de lecture et d'écriture après la fin de chaque opération. Ici, il n'y a donc pas de courant qui circule entre l'alimentation et la masse pendant la lecture ou l'écriture puisque la précharge est coupée [Duh1995]. Dans la plupart des cas, on utilise un transistor PMOS pour la précharge. Cependant, on peut aussi utiliser un transistor NMOS [Cha1995, page 336]. Ainsi, l'excursion en tension sur les lignes de bit se fera entre 0 et $V_{dd} - V_t$ et non plus entre 0 et V_{dd}, comme dans le cas précédent.

Enfin, dans le cas des ROMs, on rencontre le principe de précharge sélective qui pourrait aussi être utilisé pour des SRAMs : la précharge est réalisée avant toute opération de lecture, sur les lignes de bit qui vont être lues. Ainsi, seul un nombre limité de lignes de bit est chargé ce qui divise, par rapport au cas précédent, l'énergie consacrée à la précharge par le nombre lignes de bit chargées. Cette approche est souvent utilisée dans les ROMs basse consommation [Wes1994, page 586], [Kab1996], [DeA1997], [Tak1998].

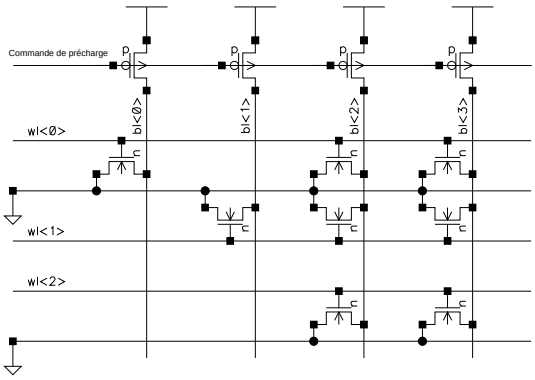
Les figures 1.22 reprennent les techniques exposées ici en les applicant à une mémoire de type ROM (Figure 1.22 a : précharge par pull-up, figure 1.22 b : précharge activée de manière partielle, figure 1.22 c : précharge sélective).

Ligne de mot pulsée

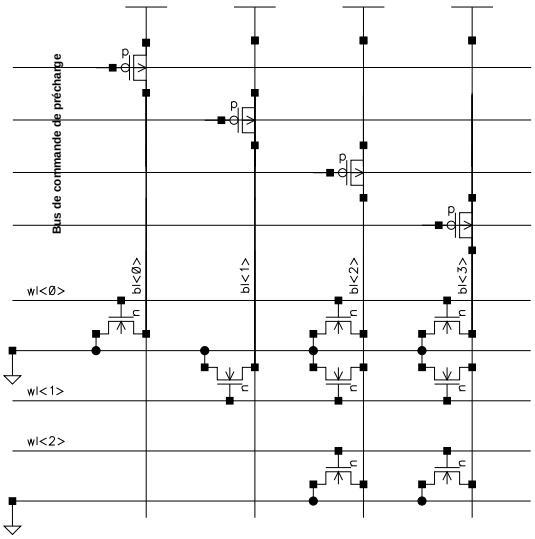
La ligne de mot pulsée (Pulsed word-line) est une technique qui, au départ, a été couramment utilisée pour limiter l'inconvénient lié à la précharge par pull-up. Elle consiste à limiter l'activation de la ligne de mot juste le temps nécessaire à l'opération de lecture ou d'écriture, de manière à écourter au maximum le temps pendant lequel l'alimentation et la masse seront en court-circuit [Kus1995]. Cependant, cette technique continue d'être utilisée même avec des circuits de précharge activés en dehors des



(a) Précharge par pull-up



(b) Précharge en dehors de la lecture



(c) Précharge sélective

FIG. 1.22 – Trois dispositifs de précharge pour les ROMs

phases de lecture et d'écriture, de façon à limiter l'excursion en tension sur les lignes de bit. Ainsi, lors de la phase de précharge, une énergie moindre sera dépensée pour recharger les lignes de bit [Amr1994].

Pour générer les signaux de commande des ligne de mot, on peut utiliser un détecteur de transition d'adresse (ATD :Address Transition Dectection) à base de chaînes de retard [Ito1995]. Cette technique nous paraît peu fiable quand au délai qui peut varier en fonction du procédé de fabrication et des paramètres électriques. Aussi, nous pensons qu'il est préférable d'utiliser des lignes de bit et des lignes de mot factices qui serviront de référence pire cas pour synchroniser les signaux [Amr1994], [Tsa1998], [Amr1999, page 55].

Isolation des lignes de bit

Lors de la lecture, l'amplificateur de lecture accentue la différence de potentiel entre la ligne de bit et son complément, de manière à accélérer la lecture. Ainsi, à la fin de la lecture et même avec l'utilisation d'une ligne de mot pulsée, l'excursion en tension pour l'une des 2 lignes de bit est proche de V_{dd} (Entre les nœuds LB et LBB sur la figure 1.23).

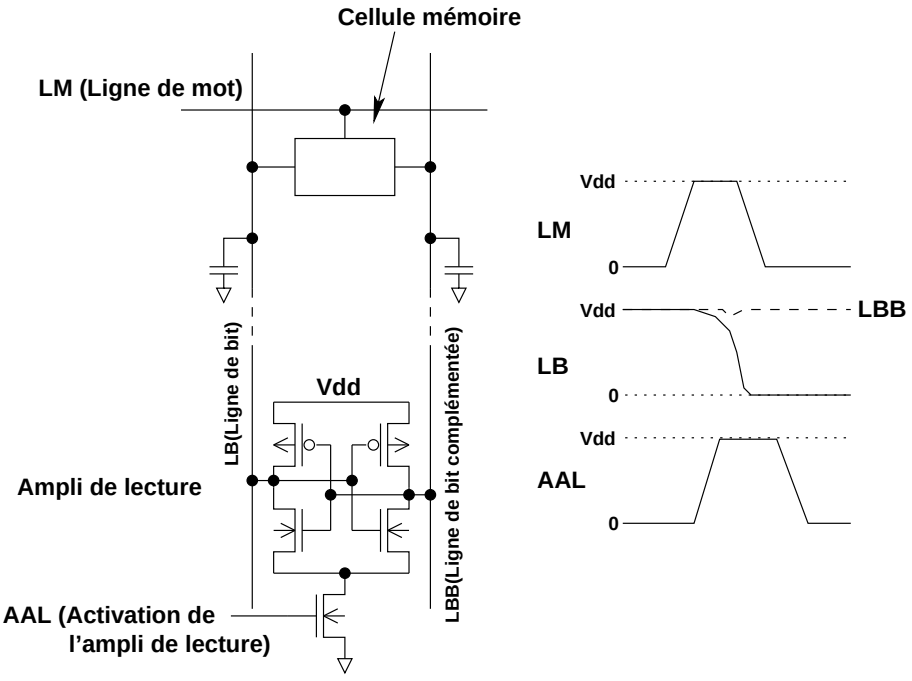


FIG. 1.23 – Lecture sur une SRAM sans isolation des lignes de bit

Pour remédier à ce problème, il est possible d'isoler la ligne de bit de l'amplificateur de lecture [Shi1995] : une fois que l'opération d'amplification a été amorcée, un switch coupe la liaison entre la partie supérieure de la ligne de bit reliée aux points mémoire

et la partie inférieure reliée à l'amplificateur de lecture. De cette façon, l'excursion en tension sur les lignes de bit est réduite (Figure 1.24). Il faut cependant veiller à ce que les signaux LM et ILB soient bien synchronisés avec AAL pour que cette technique soit efficace.

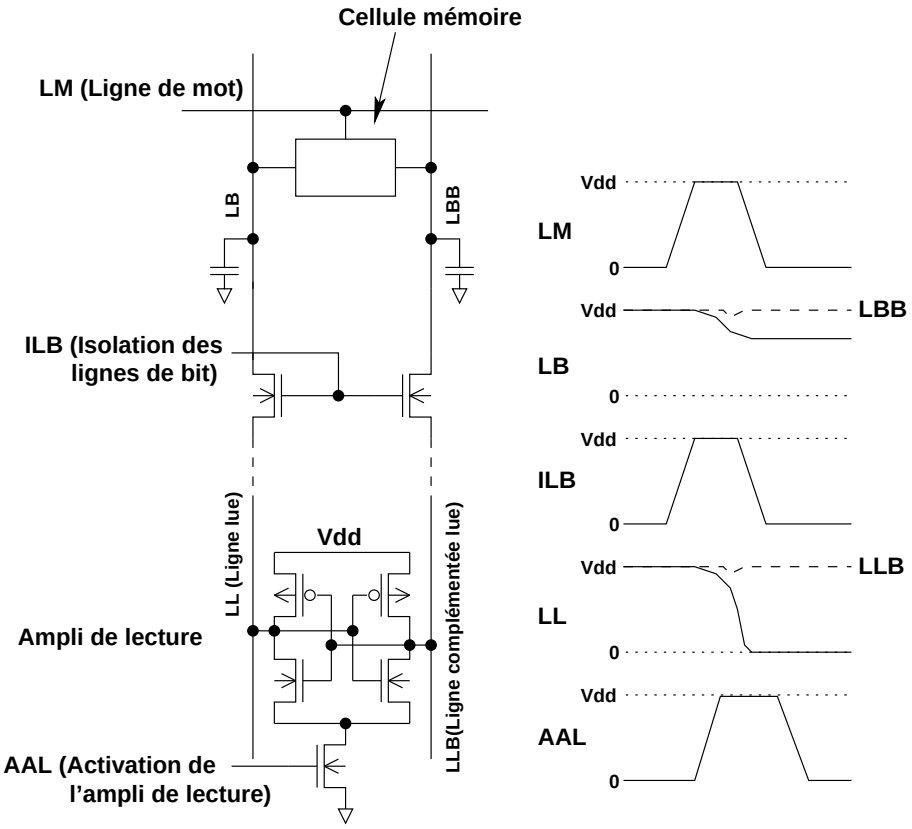


FIG. 1.24 – Lecture sur une SRAM avec isolation des lignes de bit

1.3.2 Particularités des SRAMs

Point mémoire

Dans les premières mémoires de type SRAM, les points mémoires étaient réalisés avec des transistors NMOS et une charge créée par une partie de polysilicium résistif comme le montre la figure 1.25. L'apparition des technologies CMOS complémentaires a permis de remplacer la résistance de polysilicium par un transistor PMOS (Figure 1.26), ce qui supprime le courant statique qui s'écoulait entre l'alimentation et la masse [Sas1990], [Oot1990].

Certain concepteurs de SRAMs, ont proposé de diminuer le nombre de transistors dans un point mémoire, de façon à augmenter la densité des mémoires ainsi réalisées. On

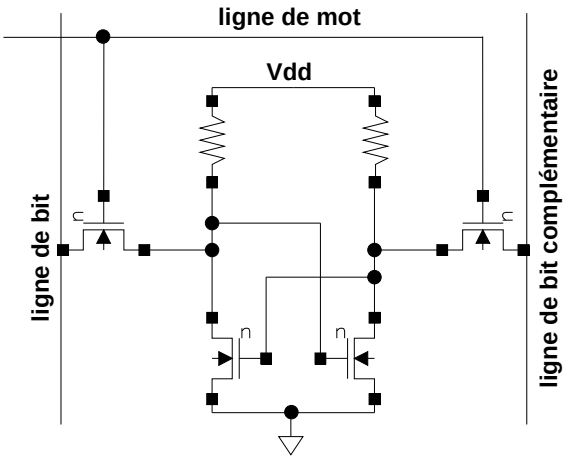


FIG. 1.25 – Point mémoire SRAM avec une charge en polysilicium

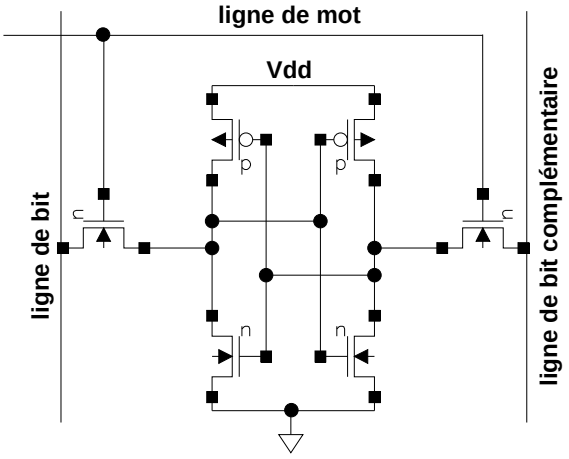


FIG. 1.26 – Point mémoire SRAM en technologie CMOS complémentaire

trouve ainsi des points mémoire à 5 transistors, en supprimant l'un des transistors de passage [Tra1996], ou encore avec seulement 4 transistors en supprimant cette fois, les 2 transistors PMOS et en remplaçant les transistors de passage NMOS par des PMOS [Tak2000], [Mas2000]. Une circuiterie assez complexe tente de minimiser les fuites à l'intérieur de la cellule, ce qui ne prédispose pas, pour le moment, ces points mémoire à la basse consommation.

Enfin, pour limiter les courants de fuites, on peut remplacer la connexion habituelle des transistors NMOS du point mémoire à la masse, par une connexion à signal (Ligne de source) dont la tension varie en fonction du mode : repos, écriture ou lecture (Figure 1.27) [Miz1996]. Cette technique présente aussi l'avantage de limiter l'excursion en tension des lignes de bit pendant l'écriture. La difficulté de cette technique est la génération d'une tension stable négative pendant la lecture.

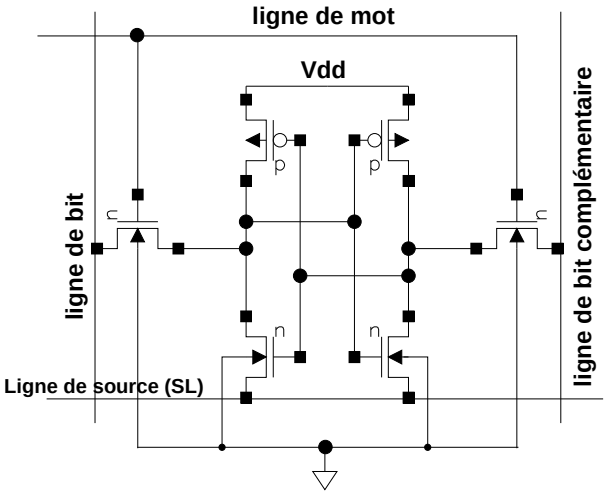


FIG. 1.27 – Point mémoire Sram avec ligne de source

Écriture faible consommation

Lors de l'écriture, il faut imposer sur toutes les lignes de bit d'un mot un niveau pendant le temps nécessaire au basculement des points mémoires correspondants. Lors de cette opération, chaque ligne de bit d'un mot effectue une excursion maximale en tension entre 0 et V_{dd}.

Afin de remédier à ce problème, une technique est proposée en réduisant la tension d'alimentation [Alo1995] (Tension de $1V$ pour une tension d'alimentation de $5V$). Pendant la lecture, la tension sur la ligne de mot est aussi réduite ($3V$) de manière à éviter des problèmes d'instabilité dans le point mémoire. Cependant, cela a un impact sur le temps de lecture puisque le courant qui traverse les transistors de passage est réduit.

De façon à accélérer le temps de lecture, il est possible de précharger les lignes de bit à une tension égale à $V_{dd}/2$ [Mor1998] ce qui permet quand même de diminuer la

consommation. Pendant l'écriture, une des 2 lignes de bit sera déchargée à 0, ce qui réduit par 4 l'énergie dépensée par rapport à une architecture conventionnelle. Il est cependant nécessaire d'utiliser une tension plus élevée que V_{dd} pour activer la ligne de mot notamment pendant la lecture.

Une approche différente consiste a utiliser le point mémoire comme un amplificateur de lecture pendant la phase de d'écriture [Amr1999, page 84].

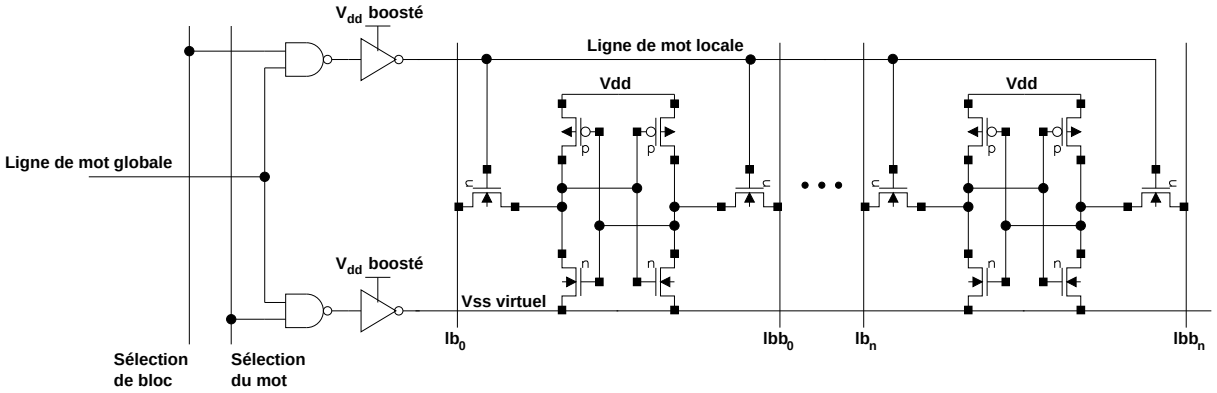


FIG. 1.28 – Utilisation du point mémoire comme amplificateur pendant l'écriture

Pendant la lecture la ligne "Vss virtuel" est maintenue à 0, ce qui place le circuit dans un cas classique. Lors de l'écriture, le niveau sur la ligne "Vss virtuel" passe à V_{dd}, ce qui efface le contenu des points mémoires sélectionnés par le signal "Sélection de mot". La ligne de mot devenant active, une faible différence de potentiel est appliquée sur les lignes de bit qui est amplifiée à l'intérieur du latch rebouclé quand la ligne "Vss virtuel" repasse à 0. Cette technique comporte cependant un bon nombre de risques : problèmes de marge de bruit, synchronisation des signaux, augmentation de la taille de la mémoire.

1.4 Conclusion

La consommation dynamique étant liée aux paramètres tension d'alimentation, fréquence de fonctionnement et capacité commutée, ce sont ces paramètres qu'il va falloir réduire principalement de manière architecturale : abaissement local de la tension d'alimentation, conditionnement des horloges pour n'activer que la partie nécessaire d'un circuit et réduction des capacités en utilisant des transistors de taille juste nécessaire pour la vitesse requise.

En voulant à la fois diminuer la consommation dynamique et augmenter la densité des circuits, on a créé des transistors avec un faible V_T de manière à pouvoir faire fonctionner les circuits à faible tension d'alimentation et à des fréquences élevées. L'apparition

de ces technologies faible V_T accentue le problème des courants de fuite qui, auparavant, était souvent négligé. La diminution de la consommation dynamique passe souvent par un mécanisme d'activation partielle des composantes d'un circuit. Aussi, dans une mémoire, quand un bloc est actif (la consommation dynamique est alors dominante pour ce bloc), tous les autres blocs sont inactifs et font apparaître un courant de fuite.

CHAPITRE 2

Conception de mémoires basse consommation

Dans ce chapitre, nous allons présenter à travers deux type de mémoires, ROM et SRAM, des techniques de réduction de la consommation dynamique. Pour le cas des ROMs nous présenterons une architecture utilisée pour le développement d'un compilateur. Pour le cas des SRAMs, nous partirons d'une instance existante que nous modifierons afin d'en abaisser la consommation sans en affecter les performances en terme de délai.

2.1 Générateur de ROM

2.1.1 Introduction

L'offre d'ATMEL comprenait avant le démarrage du projet, un générateur de ROMs basé sur une technologie 2 niveaux de métal avec une capacité maximale de 128 Kb. Cette taille ayant été jugée insuffisante pour les applications actuelles, il a été décidé de développer une nouvelle architecture pouvant atteindre une taille maximale de 4Mb pour des technologies $0.5\mu m$ et $0.35\mu m$ 3 niveaux de métal. De plus, la réduction de la consommation a été prise en compte dès le départ de manière à remplir les objectifs de basse consommation imposés par les applications embarquées. Nous allons présenter l'architecture utilisée et nous commenterons les résultats obtenus sur silicium.

2.1.2 Architecture

Partitionnement en blocs / ligne de mot divisée

L'architecture classique d'une ROM est bien connue [Rab1996, page 552]. Elle se compose de quatre parties : le plan mémoire, le décodeur de lignes, le décodeur de colonnes et un bloc pour le contrôle. Cependant, cette organisation convient mal à des mémoires de grande capacité : en effet, les charges dynamiques (charges liées aux grilles des transistors) et de routage, deviennent trop importantes, notamment pour les portes du décodeur de ligne. Si l'on considère l'exemple d'une mémoire d'une capacité totale d'un mégabit avec un facteur de forme carré (1024 lignes $\times$ 1024 colonnes), le nombre de grilles vues par le décodeur de ligne est 1024, ce qui donne en technologie $0.5\mu m$, une longueur d'environ $2200\mu m$ pour la ligne de mot. Cette charge importante contribue à augmenter à la fois le temps de propagation des signaux sur la ligne de mot et à accroître la puissance consommée lors de l'activation de ces signaux. Pour remédier à ces problèmes, nous avons opté pour une architecture de type multi-blocs en utilisant le principe de ligne de mot divisée [Yos1983]. Étant donné que le nombre maximal de blocs contigus pour un "buffer" de ligne reste inférieur à quatre (voir Fig 2.1), nous avons utilisé ce concept plutôt que celui de ligne de mot hiérarchique [Hir1990], plus adapté à un grand nombre de bloc contigus.

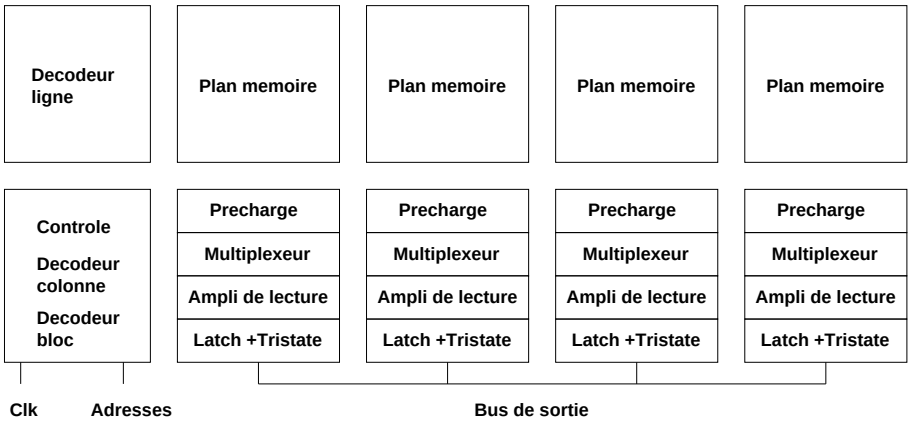


FIG. 2.1 – Architecture multi-blocs

Avec le principe de la ligne de mot divisée (Fig. 2.2), la capacité sur une ligne de mot vue par la sortie du décodeur de ligne est considérablement diminuée : la capacité totale, due aux grilles d'une ligne, est sectionnée en ligne de mot locales. Chaque décodeur local voit un nombre de grilles réduit et le décodeur de ligne global attaque un nombre de portes égal au nombre de blocs contigus.

En revanche, la capacité liée au routage peut augmenter, compte-tenu qu'à la capacité de routage de ligne de mot globale, on ajoute celle de la ligne de mot locale. Il faut s'assurer que quelque soient la technologie et la topologie, le gain en terme de capacité est toujours efficace avec cette technique.

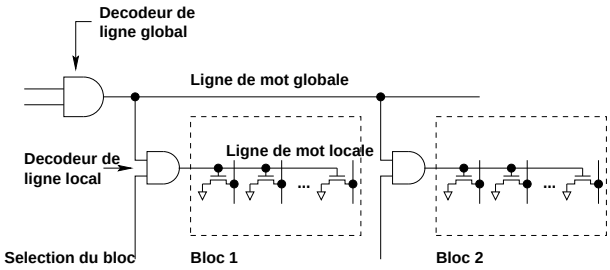


FIG. 2.2 – Le principe de ligne de mot divisée

Nous avons comparé l'approche classique avec la ligne de mot divisée (Figure 2.3) en réalisant une activation suivie d'une désactivation d'une ligne de mot pour un nombre croissant de blocs. Compte-tenu des capacités parasites liées aux géométries de notre circuit, nous voyons que le principe de ligne de mot divisée n'est intéressant qu'au delà de 2 blocs. En changeant de technologie ($0.35\mu m$) et en diminuant linéairement les tailles des transistors avec un rapport W/L constant par rapport à la technologie $0.5\mu m$, nous obtenons les courbes de la figure 2.4. Cette fois-ci, la technique de la ligne de mot divisée n'est intéressante qu'à partir de 3 blocs. Avec la réduction des géométries, les valeurs des capacités diminuent, c'est pourquoi le circuit, à nombre de blocs constant, consomme plus en $0.5\mu m$ qu'en $0.35\mu m$. En revanche, la répartition entre capacités d'interconnexion (capacités de routage et de couplage) et les capacités dynamiques de drain et de source est modifiée entre les deux technologies : avec la réduction des géométries, la contribution des capacités de drain et de grille diminue par rapport à celle des capacités d'interconnexion.

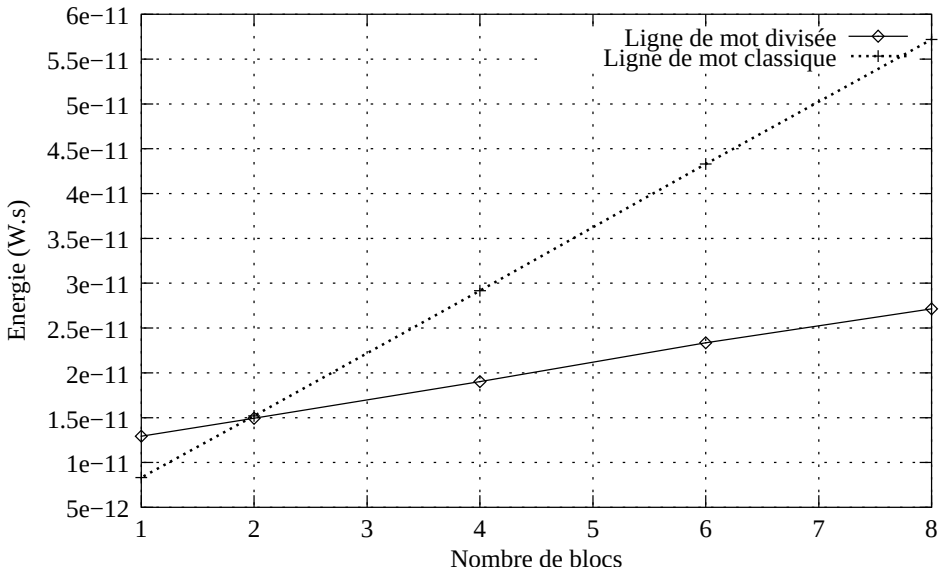


FIG. 2.3 – Comparaison de la puissance consommée par le décodeur de ligne mot entre l'utilisation de la ligne de mot divisée et une architecture classique pour une technologie $0.5\mu m$.

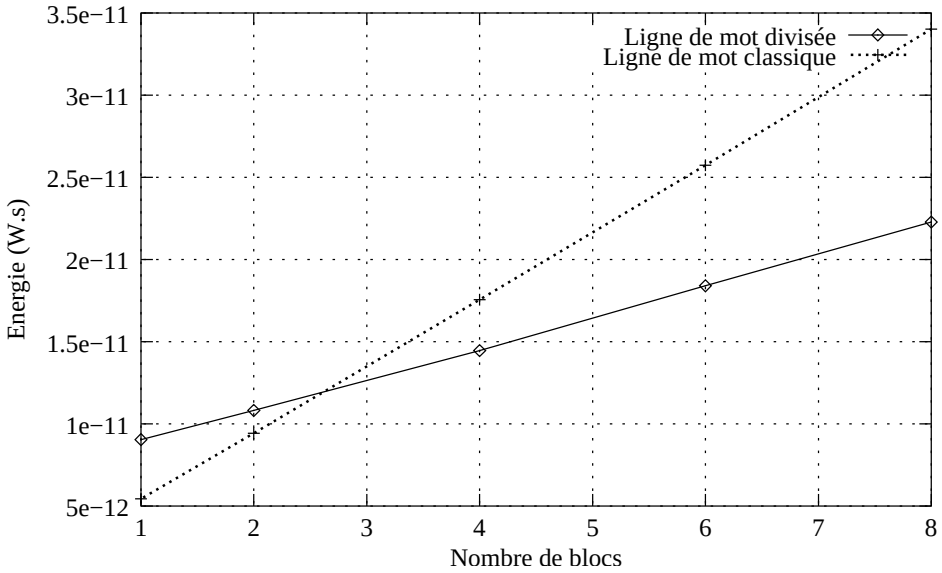


FIG. 2.4 – Comparaison de la puissance consommée par le décodeur de ligne mot entre l'utilisation de la ligne de mot divisée et une architecture classique pour une technologie $0.35\mu m$.

Le concept de ligne de mot divisée a été étendu à tous les autres signaux qui commandent les blocs de précharge, de multiplexage et l'étage de sortie de manière à diminuer le temps d'accès et à réduire la consommation totale. c'est-à-dire que tous les signaux ayant un sens de propagation horizontal dans la mémoire ont été hiérarchisés. Aussi, la mémoire n'est plus organisée en deux dimensions mais en trois : une dimension X pour le décodeur de lignes, une dimension Y pour le décodeur de colonnes et une dimension Z pour le décodeur de blocs. Afin de réduire encore la puissance consommée, un seul bloc est activé lors d'un accès en lecture, les autres blocs n'ayant qu'une consommation statique. Ainsi, tous les bits d'un même mot sont stockés dans le même bloc.

Dans l'utilisation du principe de la ligne de mot divisée, nous avons limité le nombre de blocs à quatre de manière à trouver un compromis entre la taille de la mémoire et la largeur du bus qui commandera l'activation des blocs : plus il y a de blocs à commander, plus il y a de signaux qui partent du bloc de contrôle vers les plans mémoires.

Précharge sélective

Dans les circuits de précharge classique (Figure 1.22 b), toute lecture est précédée d'une phase de précharge de toutes les lignes de bit à "vdd". Ainsi, au moment de la lecture, à l'intérieur d'un même bloc, plusieurs lignes de bit peuvent être déchargées sans pour autant être utilisées. Sur la figure 2.6, les points noirs représentent des transistors NMOS connectant une ligne de mot aux lignes de bit. L'activation de la ligne de mot

dans cet exemple, montre que la moitié (4 sur 8) des lignes préchargées à “1” seraient déchargées quelque soit la colonne lue. Afin d'économiser cette énergie perdue, nous utilisons le principe de précharge sélective [Wes1994, page 586], [Kab1996], [DeA1997], [Tak1998].

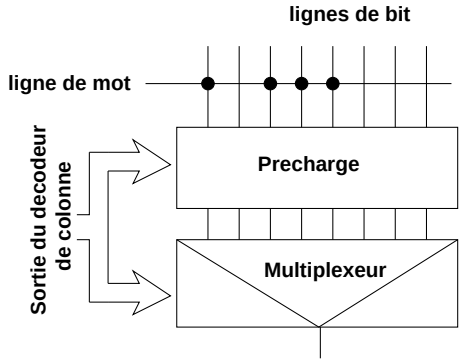


FIG. 2.5 – Lignes de bit et précharge sélective

La précharge sélective peut être définie à 2 niveaux : au niveau bloc, où l'on ne précharge que les lignes de bit du bloc qui va être lu et au niveau des lignes de bit, où seules les lignes de bit qui seront lues vont être préchargées. Nous avons choisi d'appliquer la précharge sélective à la fois au niveau bloc et au niveau des lignes de bit de manière à tirer le meilleur parti de cette technique. La précharge est conditionnée par un signal interne au bloc (Figure 2.6), ce qui permet de ne la déclencher que pendant une durée juste suffisante avant de démarrer la lecture. Ce conditionnement est nécessaire pour notre approche basse consommation, ce qui n'est pas toujours le cas dans les architectures traditionnelles [Pri1991, page 8] où la précharge est réalisée en permanence même pendant la phase de lecture. En plus de ce signal de conditionnement, un bus parcourt la mémoire afin de déterminer les lignes de bit à précharger. Étant donné que cette information est la même que celle dans les multiplexeurs de sortie, on utilise le même décodeur. On utilise aussi le même bus pour le multiplexage et la précharge en plaçant le bloc de précharge juste au dessus du multiplexeur (Fig. 2.6) et non pas tout en haut de la mémoire comme c'est le cas traditionnellement [Hir1990]. Un autre avantage de la précharge sélective est la réduction notable des courants de fuites dans les blocs au repos (Section 3.1.3, page 64). En effet, le courant de fuite est, dans ce cas, imposé par les transistors de précharge et non plus par l'ensemble des transistors du plan mémoire.

Les transistors de précharge sont de type PMOS dans la mesure où l'on souhaite opérer à des basses tensions, proches de la tension de seuil, pour lesquelles on ne veut subir aucune dégradation des signaux.

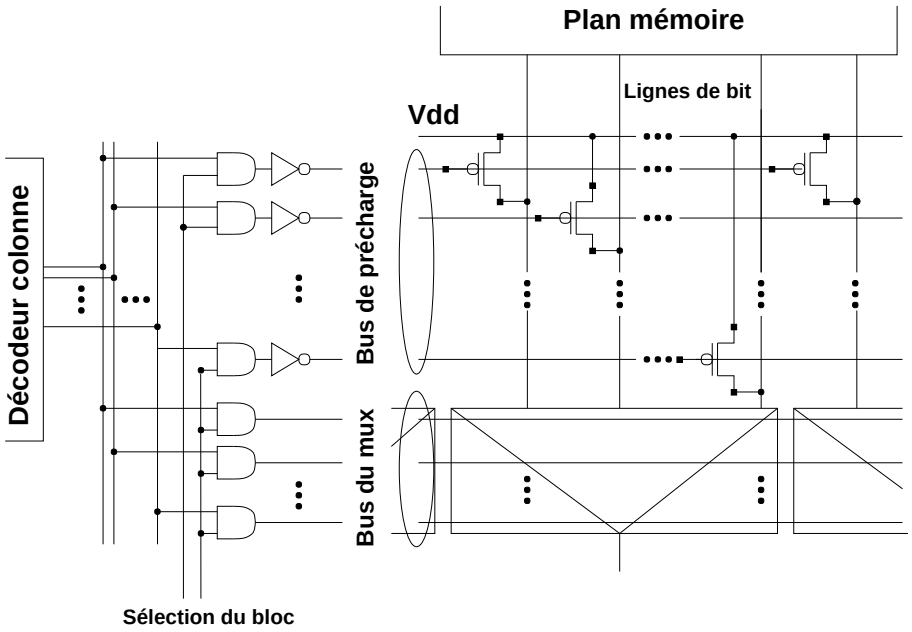


FIG. 2.6 – Implémentation de la précharge sélective

Partage de l'étage de sortie et chemin de données

Comme dans le cas des DRAMs multi-blocs [Ito1995], l'étage de sortie est partagé entre deux blocs face à face (Figure 2.7). En utilisant les signaux de sélection des blocs, un switch détermine de quel bloc il faut sélectionner la ligne de mot à lire. L'amplificateur de lecture (Sense Amplifier), noté A.L. sur le schéma, a pour but d'amplifier la décharge de la ligne de bit quand la ligne de mot est connectée à la ligne de bit par un transistor (Lecture d'un 0). En revanche dans le cas de la lecture d'un 1, la sortie de l'amplificateur doit rester au niveau haut. En dehors de la phase de lecture dans un bloc, l'amplificateur de lecture est systématiquement ré-initialisé. Ainsi la décharge des lignes de bit, par les courants de fuite, n'a aucune incidence sur la sortie de l'amplificateur de lecture.

Un second étage de multiplexeur utilisé pour des mots de 8 ou 16 bits, sélectionne la sortie d'un amplificateur de lecture parmi 4 ou 2 respectivement. Le fait de partager l'étage de sortie permet de réduire la taille du layout.

A partir des différentes parties de la mémoire exposées précédemment, nous pouvons construire un schéma du chemin de données (Fig. 2.8). Des bus globaux parcourent la mémoire (lignes de mots, bus de précharge et de multiplexage) et leurs données sont prises en compte localement dans un bloc si un signal de sélection générés par le bloc de contrôle les y autorisent.

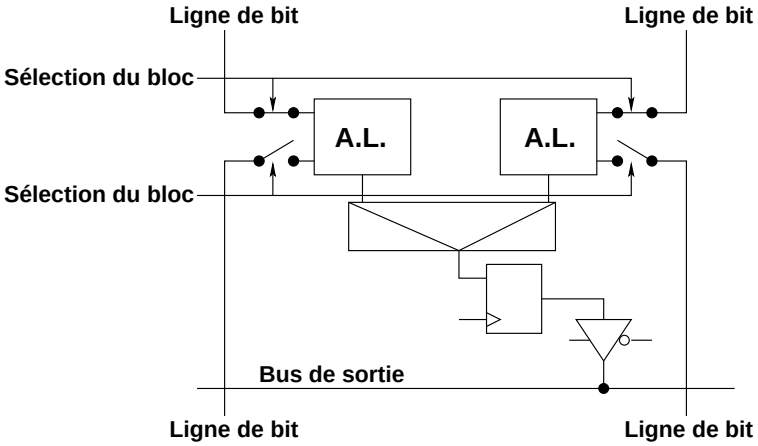


FIG. 2.7 – Partage de l'étage de sortie.

Pipe-line

Dans le cas d'une architecture traditionnelle (Figure 2.9 a), le temps d'accès est fixé par les temps de décodage et de lecture tandis qu'avec la précharge sélective, le temps d'accès inclus les temps de décodage, de précharge et de lecture puisqu'il faut décoder les adresses des colonnes à précharger, ce qui allonge le temps d'accès (Figure 2.9 b). Afin de remédier à cet inconvénient, on utilise un pipe-line à deux étages : le premier étage réalise le décodage et le second la précharge et la lecture. Ainsi, lorsqu'on active la mémoire, l'information sur les blocs et les colonnes à précharger est déjà connue et la précharge peut alors démarrer immédiatement. Le schéma du pipe-line est donné à la figure 2.10. Quand le signal CK est à l'état bas, les adresses traversent le premier étage de "latches" et le décodage est réalisé. Quand le signal CK passe à l'état haut, l'information issue des décodeurs traverse le second étage de "latches" et rend possible la précharge sur les colonnes désignées. Les adresses en entrée de la mémoire peuvent alors changer car le premier étage de "latches" est verrouillé. Ainsi, on anticipe le décodage de l'adresse suivante pendant la phase de lecture (Figure 2.9 c).

"Auto-timing"

Afin de générer les signaux pour l'activation de la précharge et de la lecture, une structure auto-contrôlée des signaux, "auto-timing", est utilisée (Fig. 2.11). Outre la réduction de la consommation statique, cette structure permet aussi de s'affranchir des contraintes de rapport cyclique du signal d'activation de la mémoire. De plus, les ROMs compilées par le générateur varient beaucoup en terme de taille (nombre de blocs, nombre de lignes) et ce pour plusieurs technologies différentes. Pour adapter la génération interne des signaux de précharge et de lecture à toutes ces variations, cette structure détermine automatiquement les temps de précharge et de lecture en utilisant

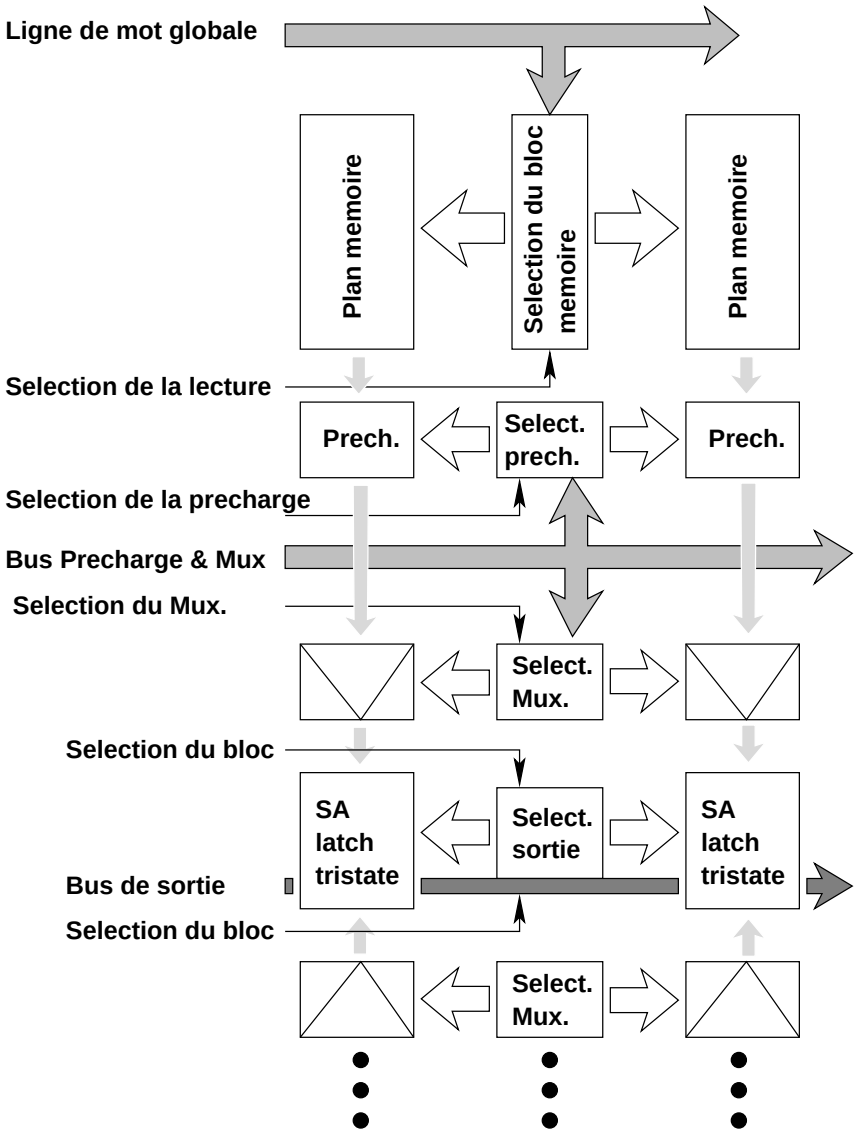
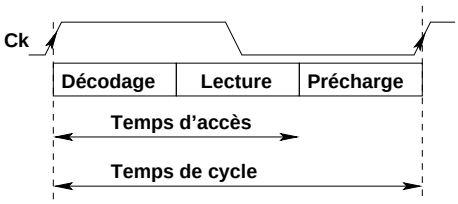
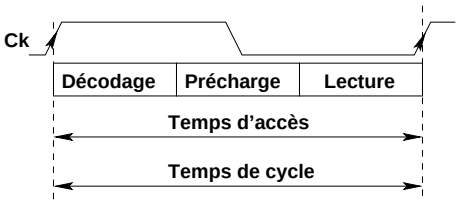


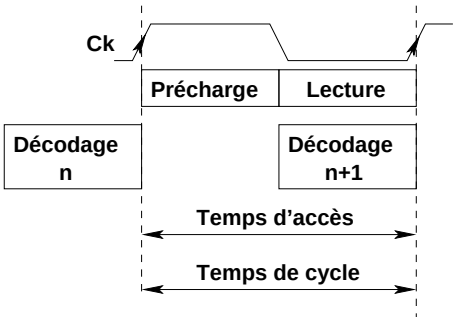
FIG. 2.8 – Chemin de données.



(a) Cas classique



(b) Precharge selective sans pipeline



(c) Precharge selective avec pipeline

FIG. 2.9 – Ordonnancement des cycles de décodage, lecture et précharge selon l'architecture

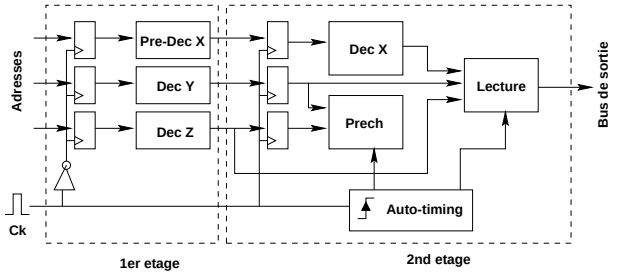


FIG. 2.10 – Le pipe-line

une ligne de bit factice qui représente un pire cas où tous les bits de la colonne seraient codés par un transistor.

Un schéma de l'auto-timing est présenté à la figure 2.11. La figure 2.12 permet de suivre l'évolution des signaux : le front montant de CK déclenche le début de la précharge sur la ligne de bit factice et aussi dans le bloc à précharger en utilisant l'information issue du décodeur de bloc. Lorsque le niveau requis est atteint sur la ligne de bit factice, le signal fin de précharge est actif et pris en compte par l'auto-timing qui stoppe la précharge et rend actif le signal de lecture pour la ligne de bit factice et pour le bloc à lire en utilisant encore l'information provenant du décodeur de blocs. La fin de lecture est détectée en sortie d'un amplificateur de lecture connecté à la bit-line factice. Étant donné que cette ligne de bit factice constitue un pire cas au niveau de la charge, on est certain que lorsque celle-ci est suffisamment déchargée pour activer l'amplificateur de lecture, toutes les autres lignes de bit de la mémoire auront déjà réagi. La fin de la lecture replace la mémoire dans un état stable, le signal de précharge n'étant pas actif.

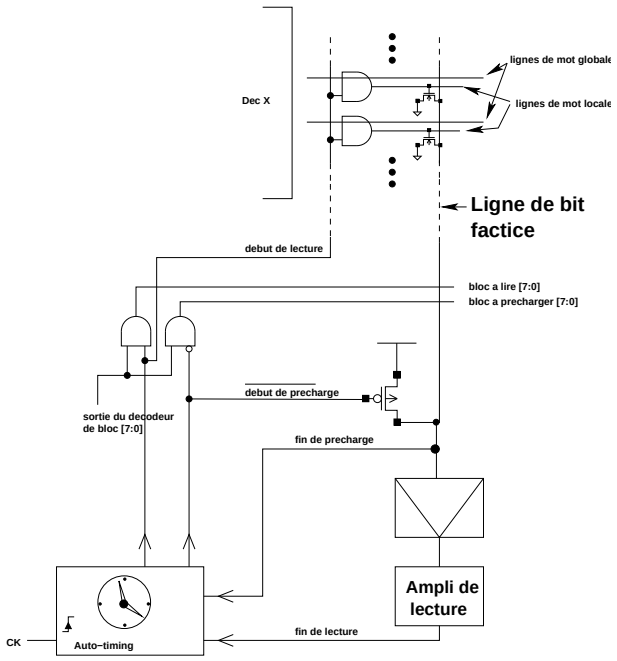


FIG. 2.11 – auto-timing

2.1.3 Résultats

En se basant sur l'architecture présentée, nous avons développé une instance 1Mb organisée en 64K-mot de 16 bits. Cette instance a été fabriquée en technologie CMOS $0.5\mu m$ puis testée, avec succès, sur silicium. De cette première version, nous avons conçu un générateur de ROMs (Cf chapitre A), disponible aujourd'hui pour 4 technologies : CMOS $0.5\mu m$, CMOS/FLASH $0.5\mu m$, CMOS $0.35\mu m$ et CMOS/FLASH $0.35\mu m$.

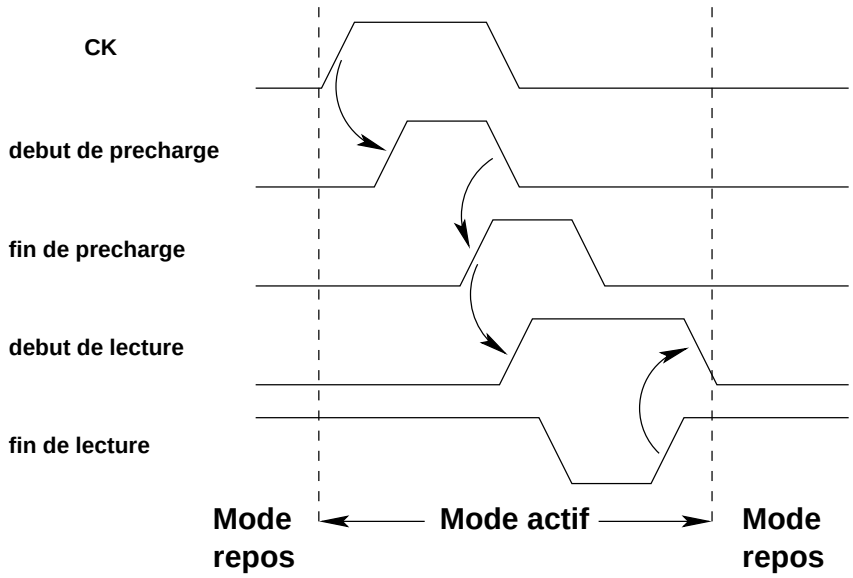


FIG. 2.12 – Chronogramme des signaux de l'auto-timing

De manière à valider notre travail, nous avons réalisé un circuit de test (testchip) (Figure 2.13) comprenant 8 instances (Tableau 2.1) jugées critiques et représentatives des extrêmes parmi celles disponibles dans le générateur et ce, dans chacune des 4 technologies citées précédemment.

Config.	Capacité (Kb)	Organisation
1	64	8Kx8
2	128	8Kx16
3	64	2Kx32
4	384	48Kx8
5	192	24Kx8
6	512	16Kx32
7	1024	64Kx16
8	4096	512Kx8

TAB. 2.1 – Liste des instances placées sur le circuit de test

Les vecteurs de test permettent d'accéder à toutes les adresses de chaque mémoire : chaque mémoire est lue de manière exhaustive. Sur testeur nous avons mesuré les temps d'accès des mémoires ainsi que leur consommation dynamique. Les délais sont mesurés à l'aide d'un chemin différentiel : on mesure le délai entre l'activation du signal "Ck" et sa sortie "DiffOut". Ensuite, on retranche ce délai aux temps d'accès mesurés sur la mémoire de manière à s'affranchir des délais liés aux plots d'entrées/sorties et au routage des signaux de façon à extraire les délais intrinsèques. Dans la réalité, il est très difficile d'obtenir des chemins identiques : les signaux d'entrée et de sortie de

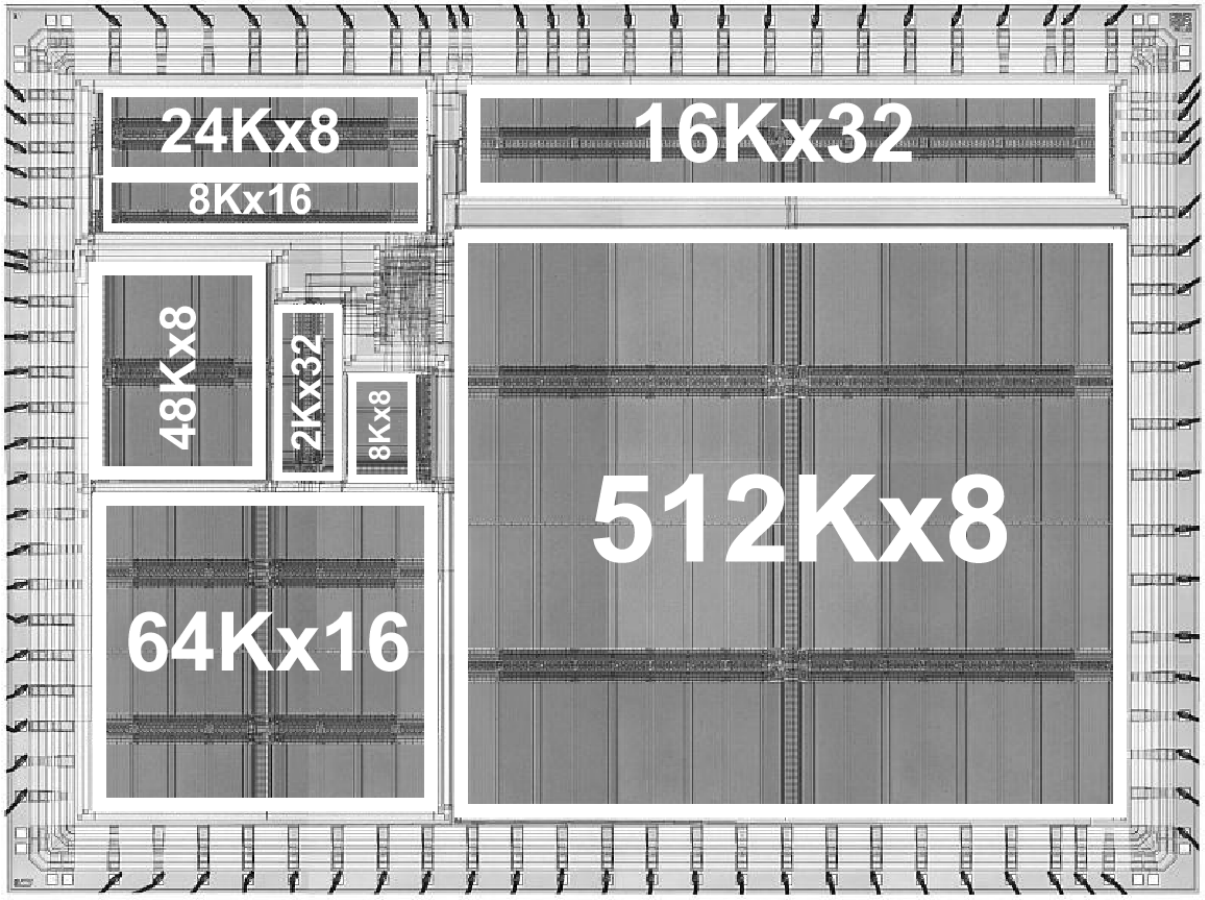


FIG. 2.13 – Testchip avec 8 instances

la mémoire sont notamment regroupés sur les largeurs très petites devant celles des plots mis côte à côte.

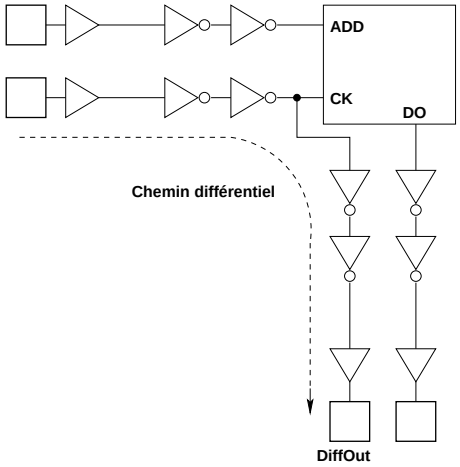


FIG. 2.14 – Chemin différentiel pour la mesure du délai

Dans le tableau 2.2, nous présentons les résultats du temps d'accès mesuré (M) sur le circuit de test. Cette mesure est comparée à la valeur simulée (S) sur le chemin critique de caractérisation. L'écart (E) entre la simulation et la mesure est donné en pourcentage. Sur de petites configurations, l'écart peut être assez grand. En revanche, c'est sur la plus grosse configuration que l'on obtient la meilleure correspondance entre la mesure et les résultats simulés car compte-tenu de la taille de cette instance, on s'affranchit le mieux des différences dans le routage entre le chemin diférentiel et la connexion de la mémoire aux plots d'entrée/sortie.

Config.	$-55°C$			$25°C$			$85°C$			$125°C$		
	M	S	E	M	S	E	M	S	E	M	S	E
1	7.1	5.9	-17%	8.5	7.5	-12%	10.9	9.1	-17%	11.6	9.5	-18%
2	8.7	5.8	-33%	10.8	7.4	-31%	14.2	9.0	-37%	16.3	9.4	-42%
3	14.7	4.6	-69%	17.4	5.9	-66%	20.4	7.2	-65%	21.9	7.6	-65%
4	8.6	6.4	-26%	9.0	8.1	-10%	11.5	9.7	-16%	12.4	10.2	-18%
5	6.2	4.9	-21%	6.8	6.2	-9%	8.2	7.5	-9%	8.8	8.0	-9%
6	16.2	6.0	-63%	15.0	7.7	-49%	21.6	9.3	-57%	23.0	9.8	-57%
7	7.9	6.4	-19%	9.2	8.1	-12%	10.7	9.7	-9%	11.4	10.2	-11%
8	8.8	8.7	-1%	10.5	11.0	+5%	13.0	13.2	+2%	13.9	13.7	+1%

TAB. 2.2 – Temps d'accès en nanosecondes ($0.35\mu m$ process typique, tension d'alimentation de 3.3V)

Pour compenser cette incertitude, nous envisageons à l'avenir, de réaliser les mesures sous micro-pointes : une sonde avec un effet capacitif très faible vient mesurer les fluctuations de tension sur un nœud du circuit, ce qui évite les problèmes liés au routage.

Par rapport à l'ancien générateur d'Atmel (Tableau 2.3), nous obtenons, avec le nouveau (Tableau 2.4), une densité meilleure notamment grâce à l'utilisation de 3 niveaux de métal et de contacts et de vias empilés. Toutes les cellules ont été dessinées aux règles de dessin minimales, de manière à obtenir un layout très compact. Le gain en densité est compris entre 20% et 92%. La puissance consommée a été fortement réduite puisque l'instance de 4Mb consomme presque autant que l'instance de 32Kb de l'ancien générateur, ce qui à tailles égales représente une division par 32 de la consommation. D'une configuration à l'autre, on a des valeurs assez différentes compte-tenu des facteurs de forme utilisés (Voir chapitre 5). La vitesse n'a pas été améliorée, ce qui constitue le prix à payer pour l'abaissement de la consommation.

Capacité (Kb)	Taille	Densité (Kb/mm^2)	Energie (mW/MHz)	Temps d'accès (ns)
32	4Kx8	133	0.79	3.79
64	8Kx8	168	1.36	4.69
128	16K8	191	2.49	4.94

TAB. 2.3 – Performances obtenues avec un ancien générateur Atmel limité à 128Kb ($0,35\mu m$ process typique, tension d'alimentation de 3.3V, $25°C$)

Capacité (Kb)	Taille	Densité (Kb/mm^2)	Energie (mW/MHz)	Temps d'accès (ns)
64	8Kx8	229	0.08	7.09
128	8Kx16	237	0.11	5.87
192	24Kx8	234	0.19	5.84
384	48Kx8	328	0.22	7.83
512	32Kx16	320	0.43	8.92
1024	64Kx16	320	0.64	7.64
4096	512Kx8	368	0.81	10.21

TAB. 2.4 – Performances obtenues avec le générateur ($0,35\mu m$ process typique, tension d'alimentation de 3.3V, $25°C$)

2.1.4 Conclusion

Un générateur de ROMs de grande capacité (jusqu'à 4Mb), destiné à des applications basse consommation, a été développé en utilisant plusieurs techniques indépendantes de la technologie. Parmi ces techniques on trouve : la partition en blocs de la mémoire, la hiérarchisation de tous les signaux internes qu'il s'agisse de signaux de commande ou de bus, la précharge sélective des lignes de bit, l'utilisation d'une ligne de bit factice pour synchroniser les signaux internes pendant le temps nécessaire afin de limiter la consommation, une minimisation des tailles des transistors et une gestion non

classique du protocole d'accès à la ROM avec l'utilisation d'un pipe-line. Les choix effectués pour ce prototype se sont révélés fructueux à travers l'analyse de circuits de test réalisés dans des technologies 3 niveaux de métal en $0.5\mu m$, $0.35\mu m$ et des technologies mixtes CMOS/Flash $0.5\mu m$ et $0.35\mu m$.

2.2 Réduction de la consommation dynamique sur une SRAM

2.2.1 Introduction

En partant d'une SRAM 16Kx16 réalisée chez Atmel en technologie $0.25\mu m$, 4 niveaux de métal, nous allons en améliorer la consommation dynamique sans modifier, ni les temps caractéristiques (Temps d'accès, temps de cycle, temps de setup, etc...), ni le protocole d'accès. Nous présenterons l'architecture de départ dont nous analyserons la consommation, puis nous décrirons les améliorations apportées pour la basse consommation.

2.2.2 Architecture et sources de la consommation

L'architecture de départ est assez classique : la mémoire est divisée en 16 blocs auxquels viennent se rajouter, le contrôle et les décodeurs de bloc et de colonne et, un décodeur de ligne comme le montre la figure 2.15. Chaque bloc se décompose en : un plan mémoire de 256 lignes par 128 colonnes et un chemin de données composé d'un bloc de précharge, d'un amplificateur de lecture, et d'un étage de sortie trois-état qui permet d'écrire sur le bus de données interne. La composition interne d'un bloc de chemin de données est détaillé à la figure 2.18.

Les signaux d'entrée et de sortie de la mémoire sont :

- `add<13 :0>` : le bus d'adresses.
- `di<15 :0>` : les données en entrée.
- `do<15 :0>` : les données en sortie.
- `ms` : un signal de mise au repos de la mémoire (Les latches d'entrée sont fermés quand ce signal est désactivé).
- `clk` : ce signal indique qu'une opération de lecture ou d'écriture est demandée.
- `wen` : ce signal, à l'état bas, indique qu'une opération d'écriture est demandée. Il doit être activé avec le signal `clk`.

Le protocole d'accès en lecture est décrit à la figure 2.16. La lecture se fait sur le front montant de `clk` quand les signaux `wen`, `ms` et `add` sont stabilisés.

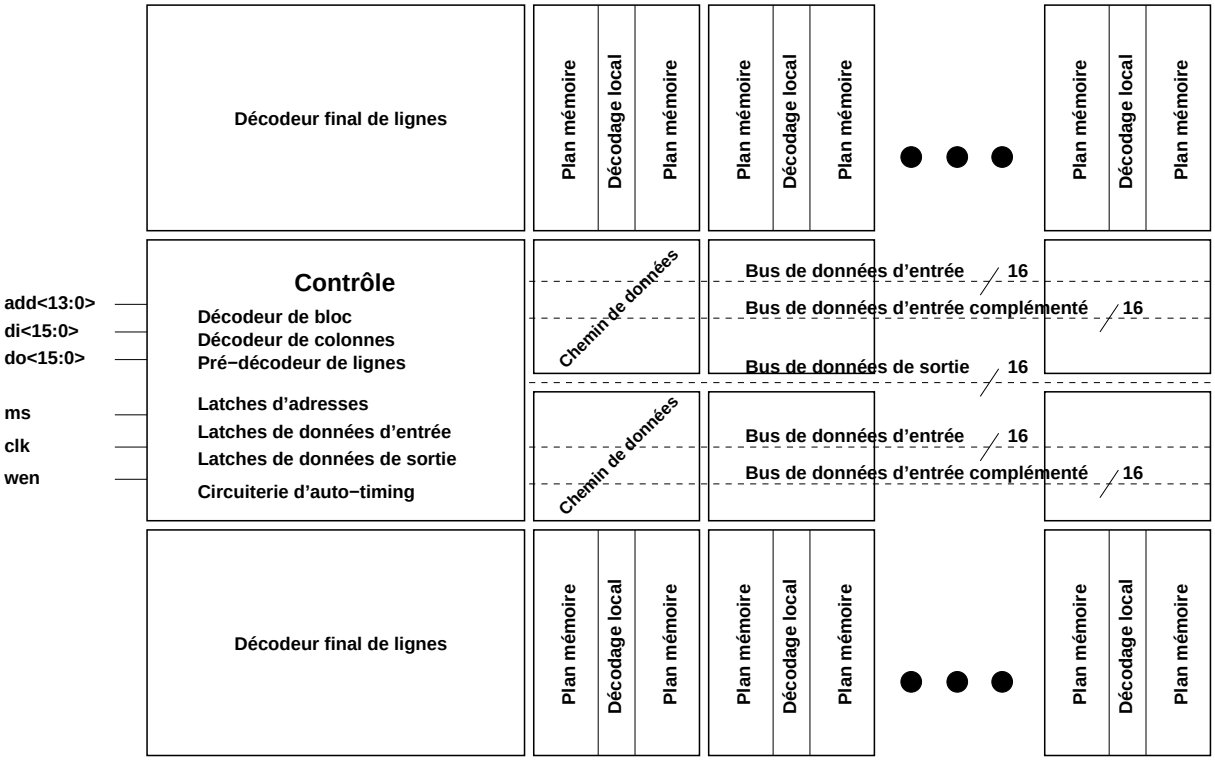


FIG. 2.15 – Architecture au niveau bloc d'une SRAM 16Kx16

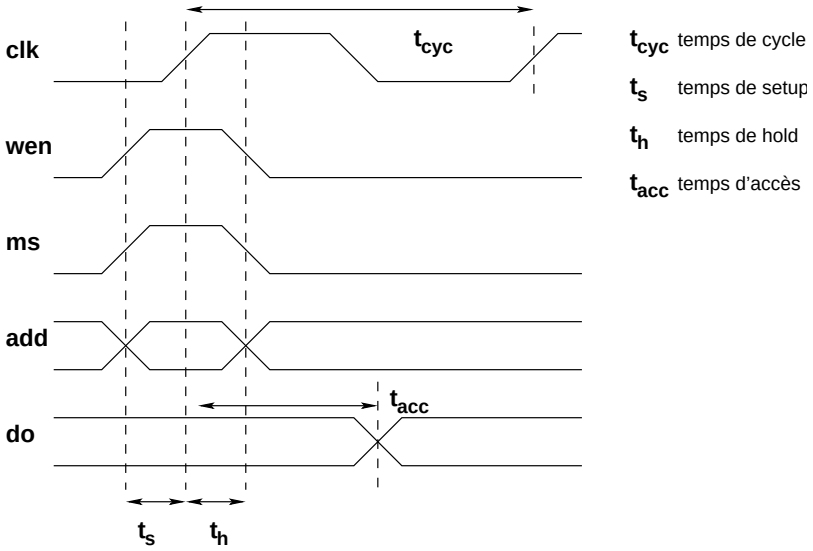


FIG. 2.16 – Protocole d'accès à la SRAM en lecture

Le protocole d'écriture est décrit à la figure 2.17. Quand `clk` passe à l'état haut, toutes les portes sur le chemin d'écriture, depuis les entrées du bus `di` jusqu'aux points mémoires, sont passantes. Ces portes sont ensuite fermées sur le front descendant de `clk` afin de verrouiller l'opération d'écriture.

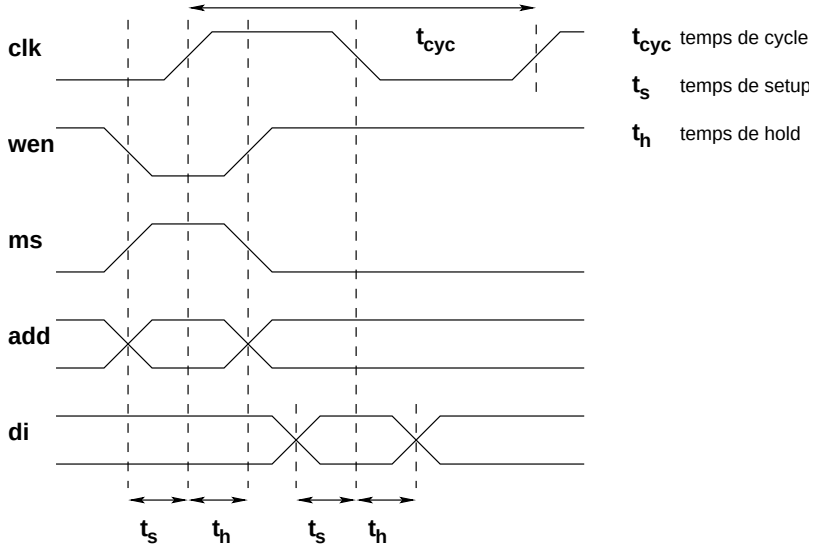


FIG. 2.17 – Protocole d'accès à la SRAM en écriture

La figure 2.19 présente la répartition de la consommation dans la mémoire, à travers des simulations effectuées sur un chemin critique utilisé à la fois pour la modélisation du délai et de la consommation.

On y trouve :

- La précharge : à chaque cycle, lors de la lecture, la moitié des lignes de bit d'un bloc sont déchargées quelque soit le contenu, puisque les lignes de bit sont complémentées. Une des lignes de bit va se décharger à travers le point mémoire, pendant que l'autre restera à V_{dd}. Il faut donc avant chaque lecture, recharger systématiquement les lignes de bit dont le niveau est inférieur à V_{dd}, sachant qu'une partie seulement des lignes de bit sera utilisée compte-tenu du multiplexage : sur 128 lignes de bit par bloc, 16 seulement sont utiles.
- Le décodeur ligne : il comprend le pré-décodage des lignes, réalisé à l'intérieur du bloc contrôle, ainsi que le décodage final des signaux des lignes de mot globales. A chaque accès, que ce soit une lecture ou une écriture, et indépendamment d'un changement d'adresses, le signal de ligne de mot globale commute en se propageant tout le long de la mémoire, ce qui constitue la première source de consommation après la précharge.
- Les données d'entrée : le bus des données d'entrée parcourt toute la mémoire. Chaque bit du bus est connecté aux 16 blocs. A l'intérieur d'un bloc, chaque bit du bus voit les 8 drains des transistors du multiplexeur d'écriture. Ainsi chaque bit est connecté à 128 lignes de bit. De plus, le bus est doublé par son complément, ce qui est équivalent

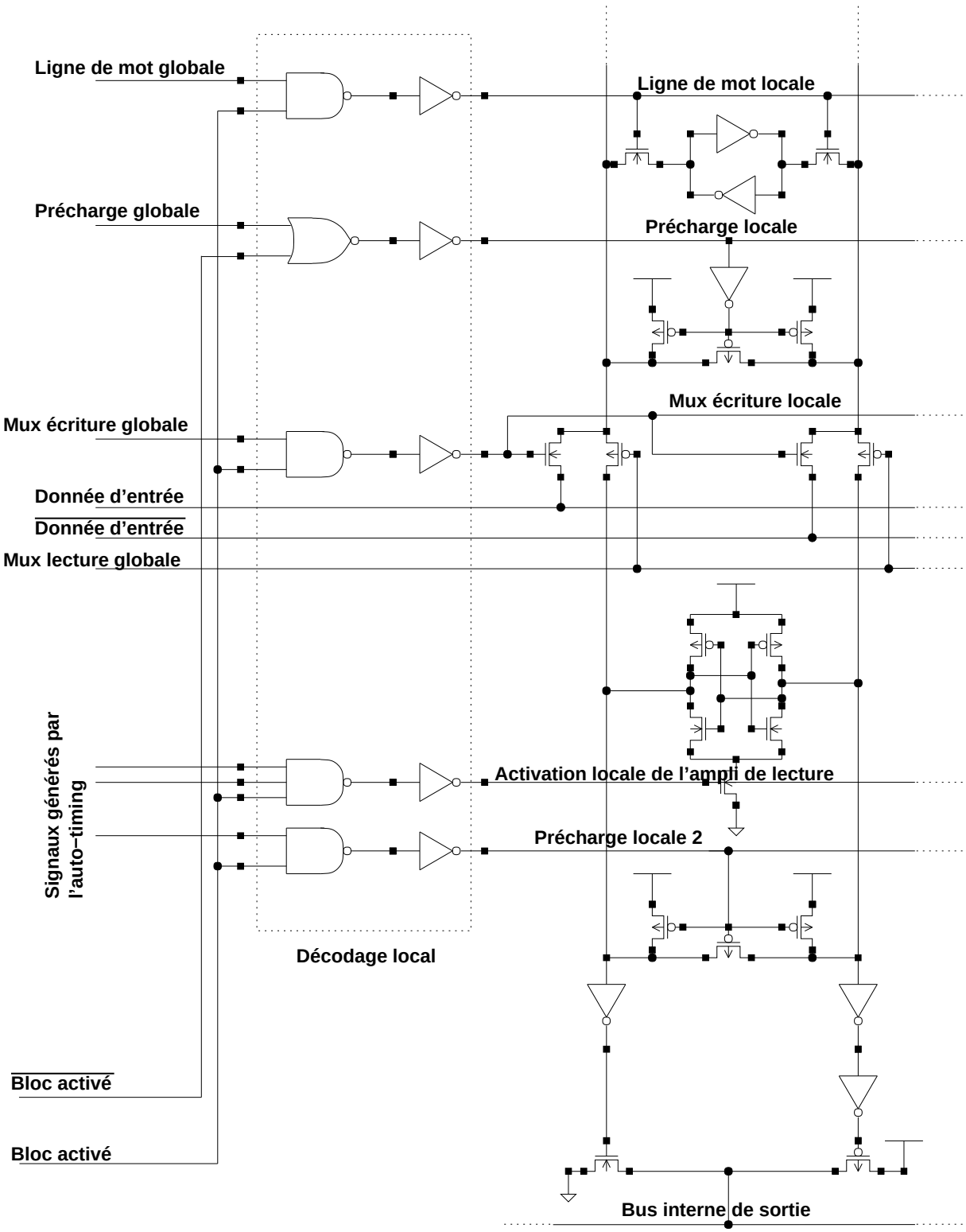


FIG. 2.18 – Chemin de données

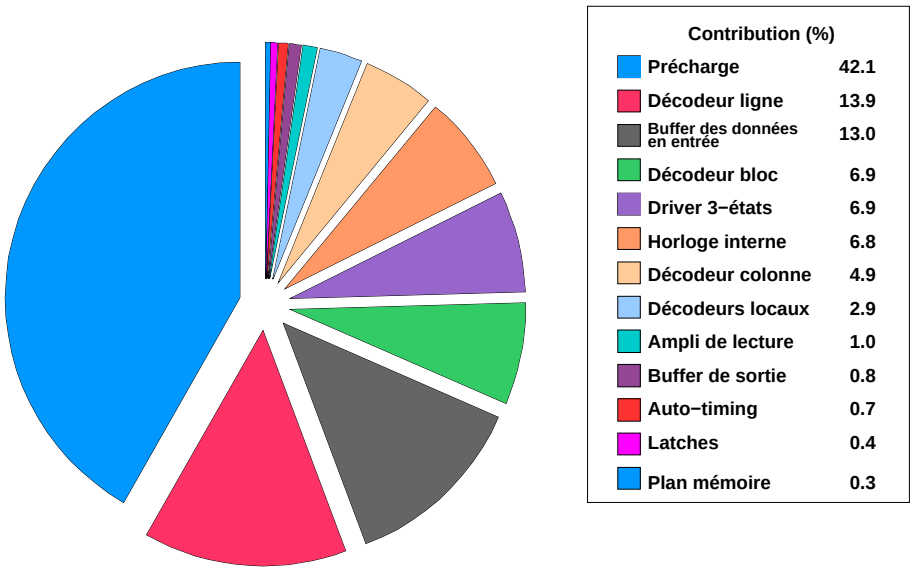


FIG. 2.19 – Répartition de la consommation

à un bus de 32 bits. A chaque fois qu'au moins une donnée d'entrée change et qu'une opération d'écriture est demandée, ce sont 2 bit du bus qui commutent.

– Le décodage bloc : il contient le pré-décodage et le décodage final des blocs ainsi que plusieurs buffers qui commandent le décodage local à l'intérieur d'un bloc. Il y a 1 signal par bloc et son complément qui partent du contrôle et qui se propagent à travers toute la mémoire. Chaque signal de commande de bloc attaque un grand nombre de grille (au moins égal au nombre de lignes) à cause de l'opération ET réalisée entre la commande de bloc et la ligne de mot globale pour l'activation de la ligne de mot locale. A chaque accès, un signal de bloc commute à travers toute la mémoire.
– Driver 3-états : la sortie du chemin de données de chaque bloc est reliée au bus interne de sortie qui parcourt toute la mémoire. La prise de contrôle et l'écriture sur ce bus se fait au moyen d'un buffer 3-états. L'écriture est réalisée avec une amplitude maximale, ce qui engendre une consommation importante.
– Le signal d'horloge interne : C'est un signal qui commande le séquencement des opérations à travers les blocs de contrôle et de décodage ligne. C'est notamment à l'intérieur du bloc de décodage de ligne que sa consommation est importante à cause de la hauteur de ce bloc liée au nombre de lignes.
– Le décodage colonne : comme les autres décodeurs, il comprend les étages de pré-décodage et de décodage final en plus de buffers qui commandent les portes de passage pour l'écriture et la lecture avec un buffer par bit de mot. Pour un accès, lecture ou écriture, on a un seul bit du bus qui commute.
– Le décodage local : il contient des portes qui réalisent une opération logique entre les signaux globaux comme les lignes de mot ou les commandes de multiplexeur et le signal de sélection de bloc. Étant donné que peu de portes sont attaquées par le décodage local, sa contribution dans la consommation totale est peu importante.

- L'amplificateur de lecture : son architecture est classique [Rab1996, page 599], [Shi1995] puisqu'il s'agit d'un inverseur rebouclé avec un transistor de commande entre les transistors NMOS de l'inverseur et la masse. L'un des ses avantages est sa faible consommation.
- Auto-timing : l'auto-timing consomme peu étant donné que le nombre de transistors est faible et les capacités qui commutent sont peu importantes.
- Le plan mémoire : sa contribution est très faible, car les transistors sont de petite taille et fournissent peu de courant. Par accès, 128 points mémoire sont activés : en lecture, la consommation est quasiment nulle car la répartition de charges se fait de la ligne de bit vers le point mémoire. C'est pendant l'écriture, d'une donnée de magnitude inverse à celle stockée précédemment dans le point mémoire, que la consommation va être la plus importante.

On remarque à travers les simulations effectuées, que la consommation n'est pas identique entre les différentes opérations d'écriture et de lecture comme le montre la figure 2.20. La consommation lors de l'écriture d'un 1 ou d'un 0 est quasiment identique car une moitié des lignes de bit du bloc sélectionné passe à 0 tandis que l'autre passe à 1. Dans les 2 cas, il y a autant de bits du bus de données d'entrée qui commutent. Dans le cas de la lecture, c'est celle d'un 1 qui est la plus importante à cause de l'écriture de la donnée sur le bus de sortie. Quand on lit un 0, il y a éventuellement une décharge des bits du bus de sortie mais cette opération se fait sans tirer de courant de l'alimentation. Le décodage colonne influe sur la consommation selon l'opération réalisée : lors d'une écriture, c'est un signal global puis local qui est généré alors que pour une lecture c'est uniquement un signal global qui commute.

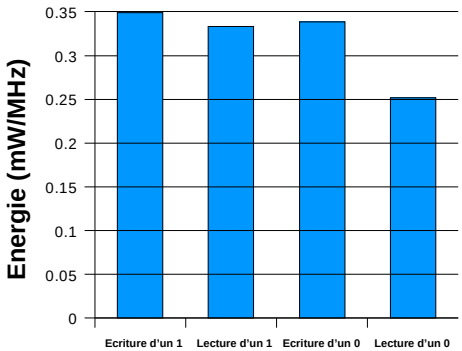


FIG. 2.20 – Contribution des différentes opérations

Désormais, nous connaissons l'architecture et la répartition de la consommation de la mémoire. Nous allons étudier la réduction globale de cette consommation en agissant en priorité sur les parties qui consomment le plus.

2.2.3 Diminution de la consommation

Nous allons proposer des améliorations pour la réduction de la consommation sans que celles-ci ne modifient les délais ou ne remettent en cause le protocole d'accès à la mémoire.

La précharge

La part la plus importante de la consommation est liée à la précharge. On s'interdit ici, à l'inverse de la ROM, présentée précédemment, de recourir à la précharge sélective dans la mesure où l'on ne souhaite apporter aucune modification aux délais. L'opération de précharge n'est utile qu'avant une lecture. Cependant, comme on ne sait pas à l'avance quelle sera la prochaine opération, la précharge est effectuée systématiquement après que la lecture ou l'écriture se soient terminées. D'après la figure 2.21, c'est après une opération d'écriture que la consommation liée à la précharge est la plus importante : en effet, une ligne de bit sur 2 d'un mot est à zéro. Ces lignes de bit doivent être rechargées à V_{dd}, l'énergie dépensée étant égale à $C_{ligne\ de\ bit} \times V_{dd}^2$ pour une colonne. En revanche, après une lecture, une ligne de bit est à V_{dd} est l'autre à une tension inférieure à V_{dd} mais non nulle, ce qui explique que l'énergie consacrée à la recharge des lignes de bit est moindre.

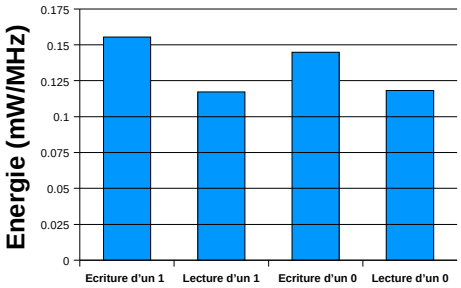
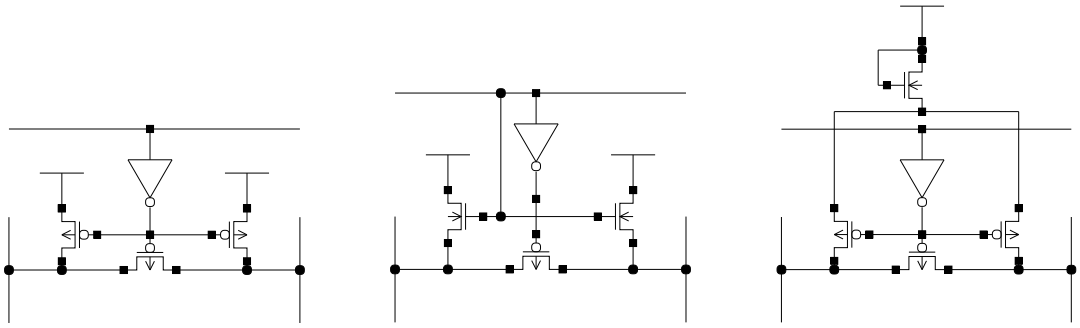


FIG. 2.21 – Consommation de la précharge durant les différentes opérations

La réduction de la consommation d'énergie peut être envisagée en réduisant la capacité de la ligne de bit en la découpant en sections plus petites [Osa1997], [Har1997] ou en réduisant l'excursion en tension sur les lignes de bit [Mor1998] en alimentant le circuit avec une tension égale à $V_{dd}/2$, générée à l'extérieur du circuit. On peut aussi utiliser le point mémoire en tant qu'amplificateur de lecture puisqu'il s'agit d'un latch rebouclé [Amr1999, page 84].

Nous allons nous intéresser à la réduction de la tension de précharge. Deux problèmes apparaissent : la génération d'une tension différente de V_{dd} et quelle est la tension optimale ? Si l'on précharge à $V_{dd}/2$, lors de la lecture, les points mémoires vont consommer de manière à rétablir un niveau haut sur une des 2 lignes de bit, à moins d'alimenter les points mémoires eux-aussi à $V_{dd}/2$. L'accès aux points mémoires se faisant par des

transistors NMOS, il apparaît, que le point mémoire ne consommera pas de courant lors de l'écriture si les lignes de bits sont préchargées à $V_{dd} - V_t$. De plus, cette tension $V_{dd} - V_t$ est très simple à générer en utilisant soit des transistors NMOS [Cha1995, page 336] pour la précharge (Figure 2.22 b), soit en alimentant la précharge au travers de transistors NMOS (Figure 2.22 c).



(a) Schema classique

(b) Precharge par Nmos

(c) Generation locale de Vdd

FIG. 2.22 – 3 dispositifs de précharge

La figure 2.23 montre les gains obtenus selon la méthode employée. Dans les 2 cas (b et c), la part de la précharge est réduite dans la même proportion. En revanche, avec la méthode b, on note une augmentation de la contribution du décodage local, à cause de l'augmentation du nombre de grilles de transistors vues par le buffer du décodeur local 2.18. Bien que le dispositif c prenne plus de place, c'est lui que nous retiendrons pour l'amélioration de la consommation.

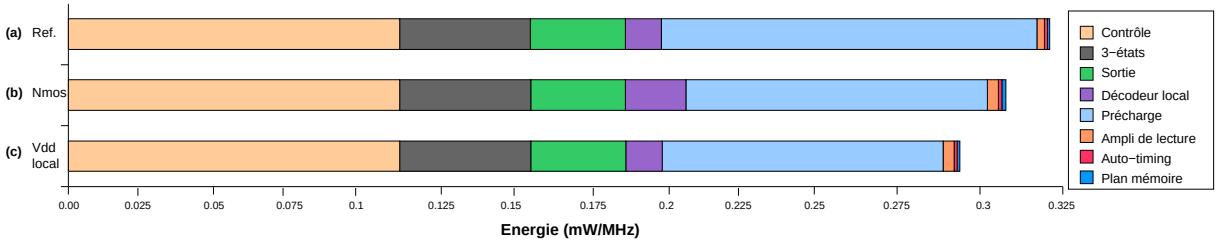


FIG. 2.23 – Comparatif des architectures pour la précharge

Hiérarchisation des signaux

En décrivant l'architecture de départ de la mémoire, nous avons vu que certains signaux voyaient un grand nombre de porte parce qu'il n'étaient pas hiérachisés. C'est notamment le cas du bus de données d'entrée qui parcourt toute la mémoire avec

sa valeur complémentée. Nous proposons une inversion locale des signaux du bus à l'intérieur de chaque bloc de manière à réduire la valeur des capacités chargées à chaque écriture d'un mot sur le bus des données d'entrée, en supprimant le bus complémenté. Le prix à payer est une augmentation de la taille du layout de chaque bloc.

De même sur le schéma 2.18, on voit que le signal qui commande l'activation d'un bloc arrive aussi sous sa forme complémentée. Nous proposons de l'inverser localement. Cela impose de rajouter un inverseur dans chaque bloc, ce qui est négligeable du point de vue de la taille du layout.

Enfin, nous pouvons agir sur la consommation liée à l'horloge interne qui se distribue depuis le contrôle dans les décodeurs de ligne, en sélectionnant l'activation de ce signal en fonction de la partie qui sera activée, haute ou basse de la mémoire.

La figure 2.24 présente les gains obtenus en hiérarchisant les signaux.

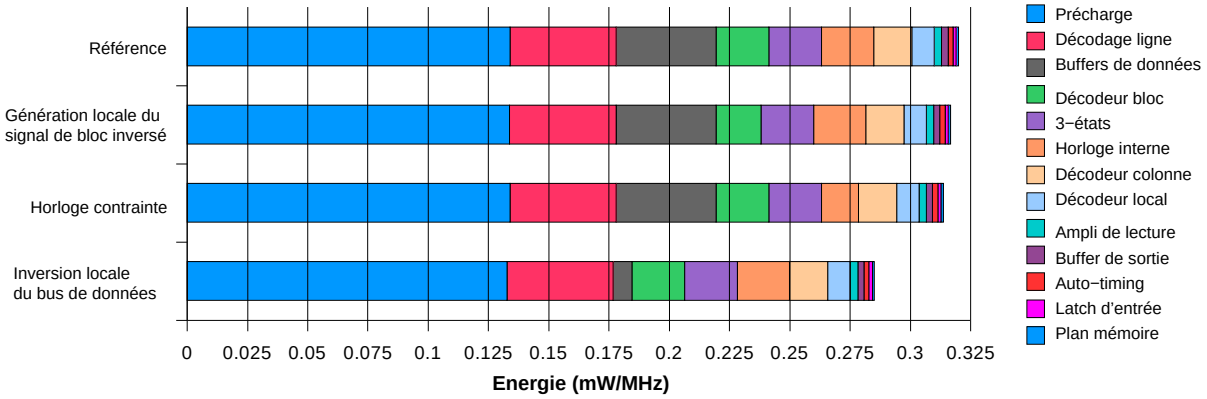


FIG. 2.24 – Gains liés à la hiérarchisation des signaux

Protocole d'accès

Dans l'architecture actuelle, l'écriture se fait sur le front descendant de `clk` (Figure 2.17) : tant que le signal `clk` est à 1, il existe un chemin entre les données d'entrée et les points mémoire. Chaque modification sur le bus d'entrée va modifier les niveaux sur la totalité du chemin d'écriture. La consommation du bus de données ayant une contribution importante dans la consommation de la mémoire, il serait judicieux de modifier le protocole pour effectuer l'écriture sur le front montant du signal `clk`. Les données d'entrées seraient alors verrouillées sur le front montant de `clk`, ce qui supprimerait une éventuelle consommation.

2.2.4 Résultat final et perspectives

Les améliorations évoquées (Précharge à $V_{dd} - V_t$, hiérarchisation des signaux) ont été intégrées dans un nouveau chemin critique afin d'estimer le gain en consommation (Figure 2.25).

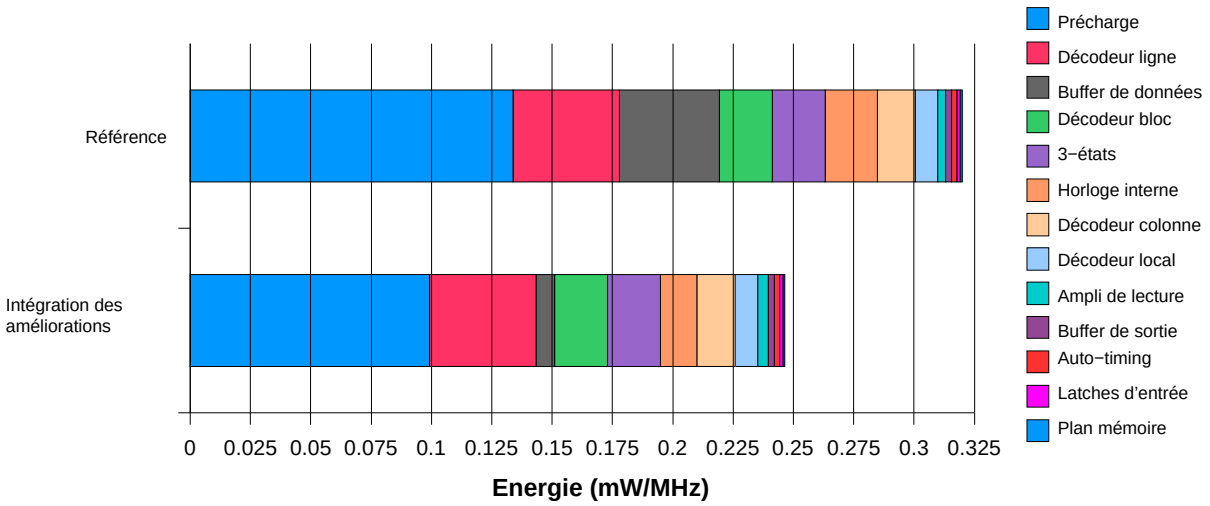


FIG. 2.25 – Comparatif entre la consommation de départ et celle avec l'implémentation des améliorations

Le gain obtenu est de 23% sans dégradation des délais 2.5. Pour diminuer davantage la consommation liée à la précharge, il faudrait recourir la précharge sélective. Une autre source importante de la consommation est le décodeur de lignes pour lequel il faudrait modifier le fonctionnement : actuellement, l'ensemble des signaux (ligne de mot globale + ligne de mot pour l'auto-timing) commutent systématique même si aucun changement d'adresse n'a lieu entre 2 opérations. L'idée d'obtenir une consommation minimale pour une succession d'opérations sur une même adresse : il suffit de détecter que les adresses n'ont pas changé entre 2 opérations pour ne pas activer la mémoire, les données en sorties étant verrouillées.

Enfin, pour réduire la consommation sur le bus interne de sortie, on envisage, pour limiter l'amplitude des signaux qui commutent, d'utiliser un second étage d'amplificateur différentiel, ce qui revient à introduire une hiérarchie dans les lignes de bit [Fre2000].

2.3 Conclusion

Dans ce chapitre, nous avons présenté une architecture de ROMs optimisée pour la basse consommation. Cette architecture a été validée sur des technologies $0.5\mu m$ et

Performances	Min	Typ	Max
temps de cycle (ns)	3.67	5.11	7.74
temps d'accès (ns)	2.26	3.18	4.91
Process	$0.25\mu m$, 4 niveaux de métal dont 3 utilisés		
Point mémoire	6 transistors		
Surface totale	$3.40\ mm^2$		

TAB. 2.5 – Caractéristiques de la SRAM 16Kx16

$0.35\mu m$ pour des instances dont la taille varie entre $64Kb$ et $4Mb$. Nous avons tiré parti des 3 niveaux de métal, de manière à augmenter la densité malgré la circuiterie ajoutée, par rapport à une architecture traditionnelle, pour la hiérarchisation des signaux notamment.

Pour la SRAM, nous avons montré qu'il était possible d'abaisser la puissance consommée jusqu'à 23%, en partant d'une architecture classique, sans changer les caractéristiques initiales de la mémoire (délai, protocole). Cette abaissement passe par la modification du circuit de précharge et par la hiérarchisation des signaux de façon à n'activer que les parties nécessaires à l'opération demandée (Lecture ou écriture).

Ces 2 architectures dont la validité n'est pas remise en cause seront portées sur des technologies plus avancées comme le $0.25\mu m$ et le $0.18\mu m$.

CHAPITRE 3

Conception de mémoires basse tension

EN réduisant la consommation dynamique par l'abaissement de la tension d'alimentation, il a fallu diminuer la valeur de la tension de seuil pour que les nouveaux procédés de fabrication ne soient pas pénalisés par une dégradation la de vitesse. Cependant, avec la diminution de la tension de seuil, les courants de fuite jusqu'alors négligés, deviennent significatifs surtout quand le nombre de transistors est important : mémoires, micro-processeur, etc... La technique CMOS complémentaire qui reposait alors sur une consommation statique nulle par rapport aux autres techniques, nécessite d'être reconsidérée pour des circuits possédant un V_t peu élevé et embarqués dans des dispositifs portables. Les courants de fuite concernent toutes les parties d'un circuit et pas seulement les mémoires comme on pourrait le croire au vu d'une littérature abondante sur le sujet. Il faut savoir que l'essentiel des courants de fuite résident dans le courant sous le seuil qui est proportionnel au W du transistor pour une technologie donnée. Ainsi, les parties les plus denses d'un circuit, sont celles ou les fuites seront les plus importantes. Elles le sont aussi pour des circuits peu actifs, dont la fréquence d'activation est faible. Les mémoires rassemblent ces deux caractéristiques : elles sont denses et la hiérarchie sous forme de partitionnement en bloc notamment, fait apparaître des zones au repos nombreuses par rapport à une partie en activité.

3.1 Consommation statique : évaluation et mesure

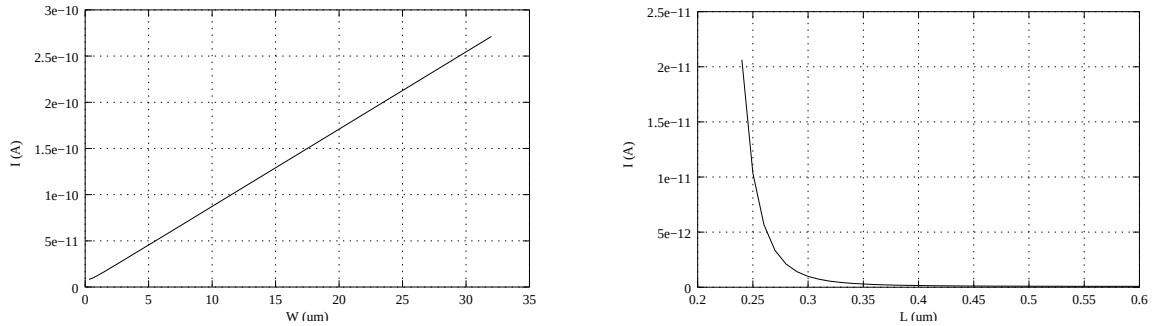
3.1.1 Simulation des courants de fuite

Pour évaluer la consommation dynamique d'un circuit, un certain nombre de méthodes de simulation sont disponibles comme l'utilisation de Spice ou de PowerMill par exemple. En revanche, la simulation des courants de fuite ne possède pas d'outil spécifique et n'est même pas évoquée dans les tutoriels Hspice [Cor1998]. On peut donc se poser la question : les courants de fuite sont-ils simulables par Spice ? Les équations du modèle BSIM3 [Liu1999] autorisent la simulation d'un circuit en faible inversion en modélisant le courant sous le seuil et le courant de polarisation de diode inverse. La nouvelle modélisation BSIM4 [Liu2000] prend en compte les effets de courant de drain induit par la grille (GIDL). Donc, intrinsèquement, la réponse est oui, encore faut-il que la carte-modèle le permette. C'est-à-dire qu'il faut s'assurer que les cartes modèles proposées par les fondeurs contiennent des valeurs correctes pour les paramètres qui permettraient de simuler ces courants de fuite. Pour s'en assurer, on peut simuler diverses caractéristiques des courants de fuite : $I(V_{gs})$, $I(W)$, $I(L)$, $I(temp)$ comme le montrent les figures 3.1. Toutes les simulations sont effectuées avec la version 1998.2 de Star-Hspice distribuée par Avant ! [Cor1998]. Ainsi seuls les courants sous le seuil et de polarisation inverse sont simulés.

A travers ces figures, on peut apprécier, non pas les valeurs simulées, mais l'allure générale des courbes :

- Caractéristique I(W) : l'équation 1.10 nous indique une dépendance linéaire entre le courant et la largeur W du transistor, ce qui correspond bien à la courbe simulée (Figure 3.1 a).
- Caractéristique I(L) : toujours d'après 1.10, on une dépendance inversement proportionnelle entre le courant et la longueur du canal L. Ce qui se traduit sur la courbe (Figure 3.1 b), où le courant diminue avec l'augmentation de la valeur de L.
- Caractéristique I(Vgs) : à la section1.2.2, nous avons vu que le courant sous le seuil est caractérisé par une pente comprise entre $60mV/dec$ et $90mV/dec$, puis il connaît une inflexion au fur et à mesure que V_{gs} se rapproche de V_t quand le transistor arrive en forte inversion. C'est ce que nous observons sur la figure 3.1 c.
- Caractéristique I(temp) : enfin, d'après 1.10, le courant dépend quadratiquement de la température, ce qui le cas sur la figure 3.1 d.

Ces observations nous permettent de relever une erreur dans l'un des modèles $0.25\mu m$ dont les simulations sont présentées à la figure 3.2. Pour la courbe du fondeur 1, la linéarité du courant de fuite par rapport à W n'est pas établie, il n'y a pas d'asymptote à la courbe qui passe par l'origine. Ainsi, on s'aperçoit visuellement que ce modèle ne peut pas être utilisé pour simuler les courants de fuites. C'est en simulant au moins,



(a) I(W)

(b) I(L)

(c) I(Vgs)

(d) I(temp)

FIG. 3.1 – Caractéristiques des courants de fuite en fonction de quatre paramètres.

les quatre caractéristiques précédentes, que l'on rapidement s'apercevoir si le modèle considéré peut être utilisé pour la simulation des courants de fuites.

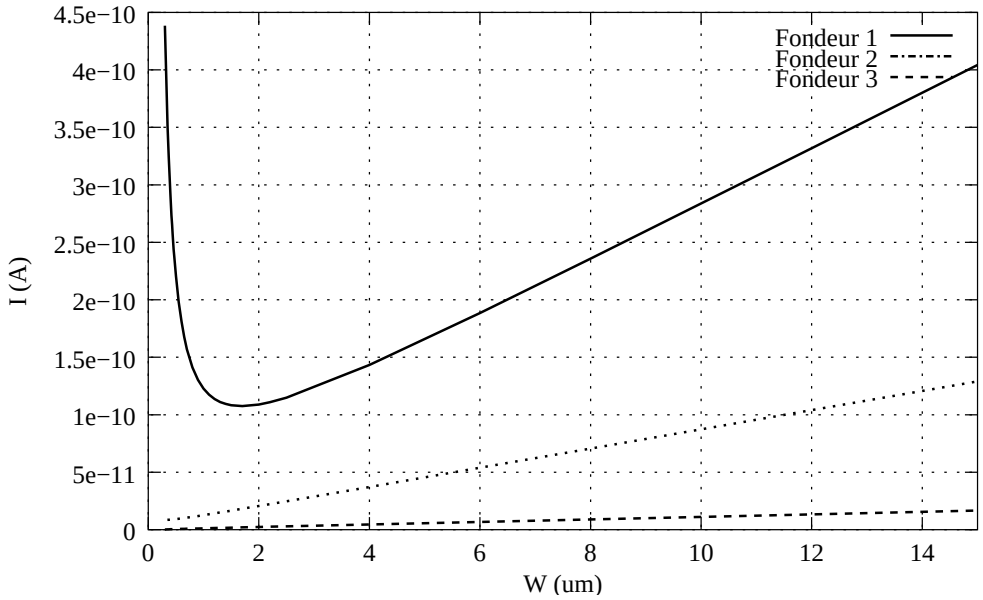


FIG. 3.2 – Caractéristique $I(V_{gs})$ montrant une non-linéarité pour l'un des modèles.

De façon à caractériser les courants de fuites pour plusieurs modèles provenant de plusieurs fondeurs et ce, pour différentes technologies, nous avons développé un script de caractérisation automatique en langage Perl.

Influence du paramètre GMINDC

La valeur du paramètre GMINDC est fondamentale pour la simulation des courants de fuite. Pour aider à la convergence d'une simulation, une conductance de valeur GMINDC est rajoutée en parallèle de chaque jonction PN. Ainsi, pour simuler la caractéristique $I(V_{gs})$, Spice simule en réalité, le schéma représenté à la figure 3.3.

Ainsi, le courant mesuré par l'ampèremètre est égal à : $I_{sth} + I_{rb} + V_{sb} \times GMINDC$. Il faut alors veiller à ce que la valeur du courant de fuite ne soit pas masquée par le produit $V_{sb} \times GMINDC$. Sur la figure 3.4, pour chaque valeur de GMINDC, la courbe marque un plateau égal à $V_{sb} \times GMINDC$. Avant de simuler les fuites sur un circuit complet, il faut s'assurer que, pour le plus petit des transistors, son courant de fuite ne sera pas inférieur à la valeur $V_{sb} \times GMINDC$. Par défaut, la valeur de $GMINDC$ est égale à $1e - 12$.

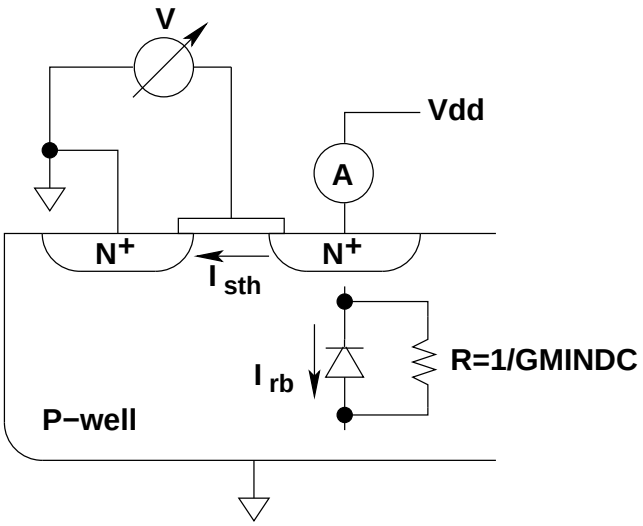


FIG. 3.3 – Schéma représentant la simulation de la caractéristique $I(V_{gs})$.

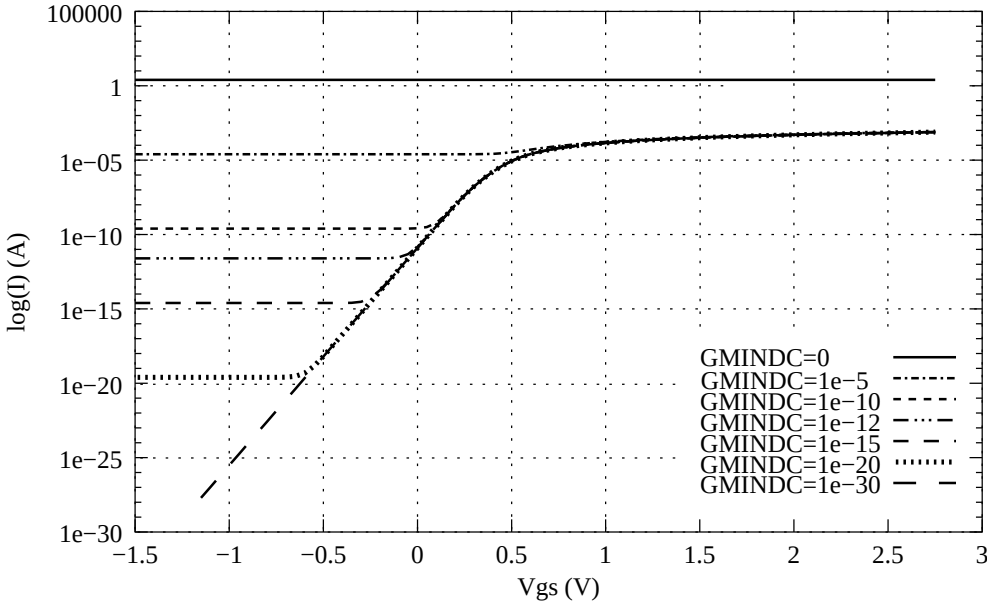


FIG. 3.4 – Influence du paramètre GMINDC sur la caractéristique $I(V_{gs})$.

Mode d'analyse (Transitoire ou DC)

Toutes les simulations présentées dans ce chapitre, sont effectuées avec le mode DC de Spice, c'est-à-dire que l'on réalise ici une analyse statique du transistor. Le mode de simulation transitoire (TRAN), n'est pas recommandé pour la simulation des courants de fuite. La figure 3.5 présente deux courbes, une réalisée en mode DC, et l'autre en mode transitoire. Pour le mode transitoire, on fait varier V_{gs} dans le temps (On applique une rampe sur la grille du transistor). On voit que les courbes sont quasiment superposées pendant le début de la forte inversion. En revanche, à $V_{gs} = 0$, la différence est importante.

La simulation d'un circuit complet est compliquée par l'utilisation du mode DC : comme, il n'est pas possible de simuler le fonctionnement d'une mémoire dans le temps et de mesurer les courants de fuite en continu, il faut recourir à une simulation en deux temps : le premier est consacré à la simulation dynamique (mode transitoire : TRAN) au cours de laquelle, on choisit un moment où la mémoire est au repos pour figer la valeur de chaque nœud du circuit dans un fichier. Dans le second temps, on effectue la simulation de la mémoire en mode statique (DC) en prenant soin d'initialiser les nœuds du circuit avec les valeurs sauvegardées précédemment dans un fichier. Il n'est pas question de se passer de la première simulation, étant donné que la valeur des nœuds est déterminante pour connaître les courants de fuite : en effet, les fuites pour un inverseur par exemple, seront différentes selon le transistor qui fuit.

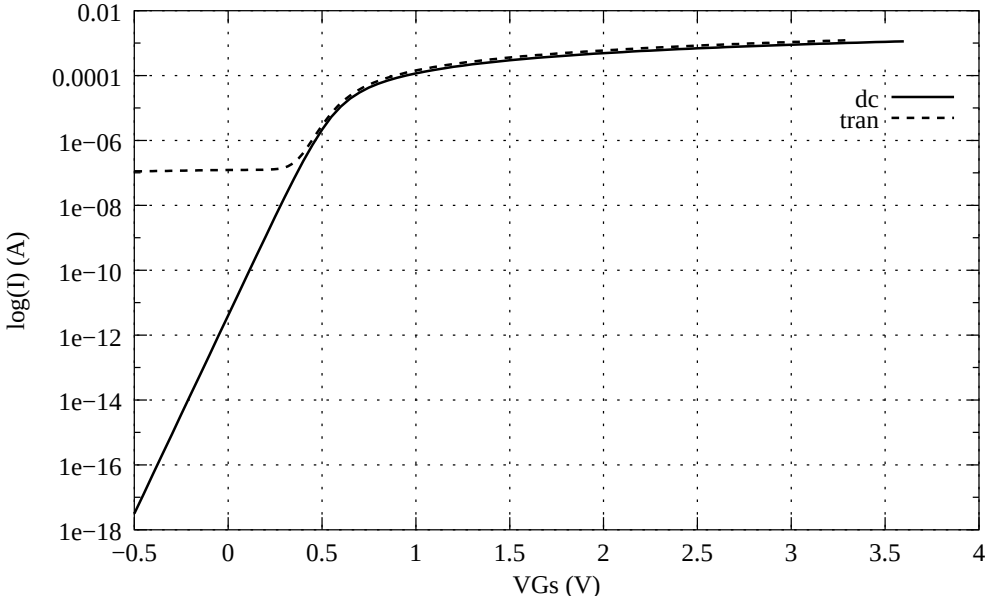


FIG. 3.5 – Influence du mode de simulation (Transitoire ou DC).

3.1.2 Mesure sur silicium des courants de fuite sur différents plan mémoires de ROM, SRAM et SRAM double port.

Étant donné que la simulation électrique des courants de fuite peut s'avérer incertaine, nous avons procédé à une mesure sur silicium de ces courants en nous intéressant plus particulièrement aux plans mémoires car ils constituent la partie la plus dense d'une mémoire. Ensuite, nous avons expérimenté des techniques pour réduire le courant sous le seuil. Cette étude a été réalisée sur trois types de plan mémoire : ROM, SRAM et SRAM à double port d'écriture et de lecture (DPRAM), sur une technologie $0.35\mu m$. Les figures 3.6 à 3.8 mettent en évidence, par des flèches, les courants sous le seuil qui traversent les transistors bloqués. Dans ces trois figures, nous nous intéressons au comportement de la mémoire en mode repos : en attendant la prochaine opération, la précharge est activée et les commandes de ligne de mots sont désactivées.

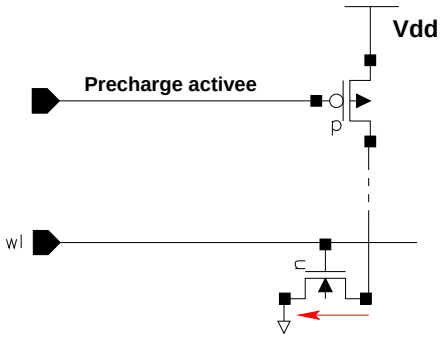


FIG. 3.6 – Courant de fuite à travers un point mémoire de type ROM

Nous avons réalisé des structures qui ne contiennent que des plans mémoires avec un circuit de précharge. Pour la ROM, la taille du plan mémoire est de 2048 bits, pour la SRAM, 512 bits, et pour la DPRAM, 256 bits. Pour les trois types de points mémoires différents, nous avons identifié et mesuré sur silicium, le courant qui circule à travers les transistors de précharge, c'est-à-dire majoritairement le courant sous le seuil (Tableaux 3.1 à 3.3). A chaque fois, la valeur du courant est rapportée à une seule cellule. Les valeurs sont mesurées à partir d'une centaine d'échantillons prélevée sur 4 plaques (wafer) différentes, le lot étant considéré comme 'typique' ($V_{tn} = 0.5V$, $V_{tp} = -0.7V$). Le courant de polarisation de diode inverse a aussi été mesuré en changeant la polarisation des caissons. Les valeurs ne sont pas présentées dans le détails puisqu'elles ont, en général, une valeur négligeable d'environ $1e-15\ A$.

Le banc de mesure est composé des instruments suivants :

- D'une de source courant / tension (HP4142)
- D'un capacimètre (HP4284)
- D'une matrice de test (HP4085)
- Prober Electroglass (HP4080)
- D'une station de travail pour piloter l'ensemble (HP725)

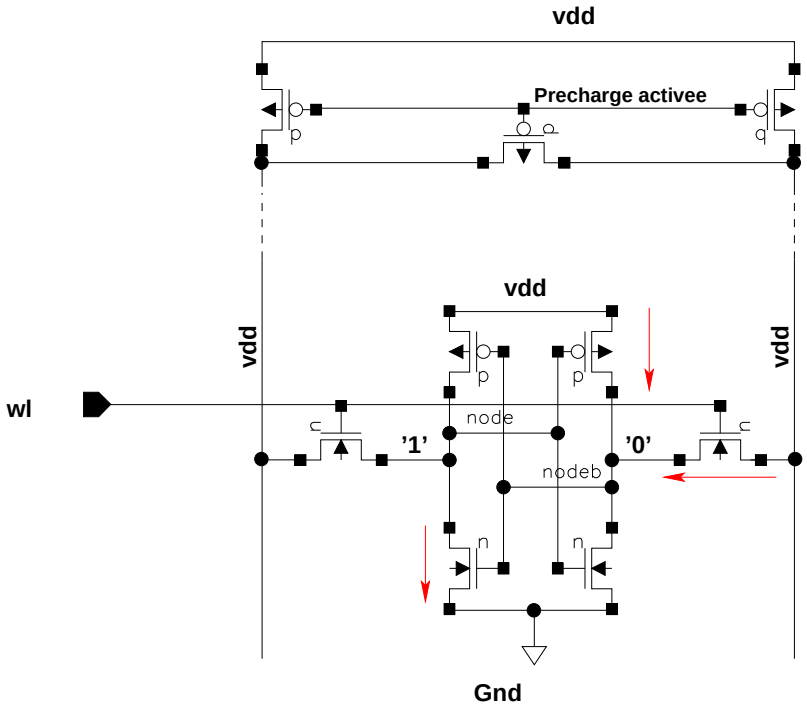


FIG. 3.7 – Courant de fuite à travers un point mémoire de type SRAM

Température	$30°C$			$85°C$		
Vdd	$3V$	$3.3V$	$3.6V$	$3V$	$3.3V$	$3.6V$
Min. (A)	$3.75e-13$	$4.16e-13$	$4.68e-13$	$1.46e-11$	$1.58e-11$	$1.74e-11$
Med. (A)	$5.45e-13$	$5.90e-13$	$6.42e-13$	$1.96e-11$	$2.11e-11$	$2.29e-11$
Max. (A)	$7.71e-13$	$8.28e-13$	$9.32e-13$	$2.52e-11$	$2.74e-11$	$2.96e-11$

TAB. 3.1 – Courants de fuite à travers un point mémoire de type ROM

Température	$30°C$			$85°C$		
Vdd	$3V$	$3.3V$	$3.6V$	$3V$	$3.3V$	$3.6V$
Min. (A)	$9.01e-12$	$9.98e-12$	$1.11e-12$	$2.96e-11$	$3.20e-11$	$3.46e-11$
Med .(A)	$1.32e-12$	$1.57e-12$	$1.97e-12$	$3.95e-11$	$4.61e-11$	$5.50e-11$
Max. (A)	$3.67e-12$	$6.82e-12$	$1.54e-12$	$8.44e-11$	$1.41e-10$	$2.67e-10$

TAB. 3.2 – Courants de fuite à travers un point mémoire de type SRAM

Température	$30°C$			$85°C$		
Vdd	$3V$	$3.3V$	$3.6V$	$3V$	$3.3V$	$3.6V$
Min. (A)	$1.26e-12$	$1.37e-12$	$1.51e-12$	$5.65e-11$	$6.03e-11$	$6.51e-11$
Med. (A)	$1.59e-12$	$1.76e-12$	$1.97e-12$	$6.64e-11$	$7.19e-11$	$7.78e-11$
Max. (A)	$3.18e-12$	$4.65e-12$	$7.30e-12$	$8.72e-11$	$9.81e-11$	$1.10e-10$

TAB. 3.3 – Courants de fuite à travers un point mémoire de type DPRAM

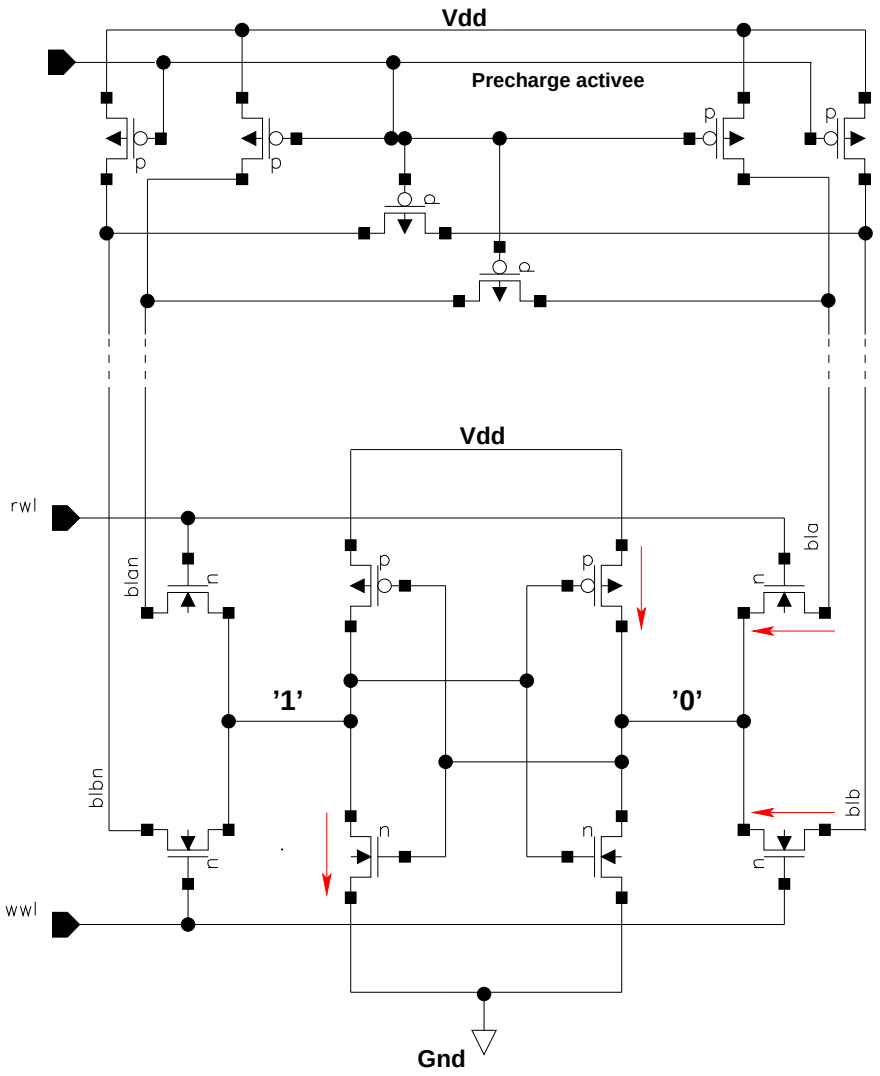


FIG. 3.8 – Courant de fuite à travers un point mémoire de type DPRAM

Les valeurs présentées dans ces tableaux vont nous servir de référence par rapport aux techniques de réduction des courants de fuite développées par la suite.

3.1.3 Effet de la précharge sélective sur les courants de fuite

Dans la partie consacrée au compilateur de ROM, nous avons vu à la section 2.1.2, page 34, les effets de la précharge sélective sur la consommation dynamique. Nous nous intéressons maintenant à son influence sur les courants de fuite.

Points mémoires de type ROM

Pour une ROM ayant un fonctionnement classique, le courant de fuite pendant la précharge est imposé par la somme des contributions de chaque transistor NMOS connecté à la ligne de bit préchargée comme l'indique la figure 3.9. Ainsi le courant de fuite pour une colonne s'écrit : $I_{CL} = m \times I_n$, m étant le nombre de lignes. Le transistor PMOS de précharge ne limite pas la fuite puisqu'il est passant ($V_{gs} > V_t$) et qu'il peut délivrer un courant de saturation bien supérieur à I_{CL}. Ainsi, c'est le nombre de transistors NMOS qui impose la valeur de la fuite dans la colonne.

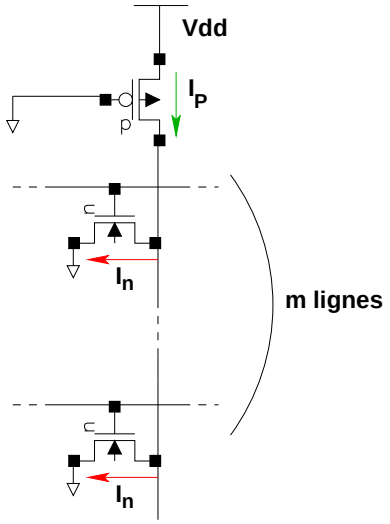


FIG. 3.9 – Courants de fuite dans un plan mémoire de ROM avec précharge par défaut

Maintenant, observons le cas de la précharge sélective, à la figure 3.10. Cette fois-ci, le transistor de précharge fuit puisque $V_{gs} = 0$, et dans chaque transistor NMOS, circule un courant I'_n. La somme des courants I'_n est égale au courant de fuite I'_p traversant le transistor PMOS, puisque c'est ce transistor qui est relié à l'alimentation. Or $m \times I_n \gg I'_p$,

c'est le transistor de précharge qui cette fois-ci impose la valeur du courant de fuite. Bien que la valeur du potentiel de la ligne de bit soit indéterminée, la fuite à travers la colonne est considérablement réduite.

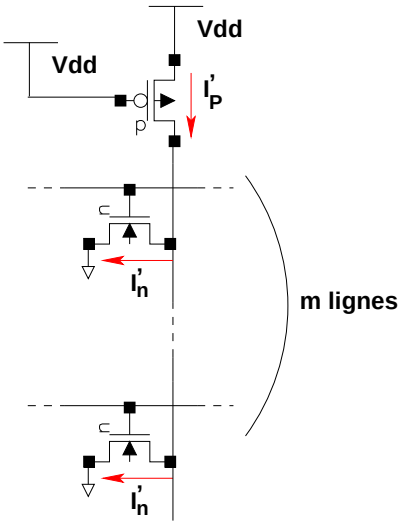


FIG. 3.10 – Courants de fuite dans un plan mémoire de ROM avec précharge sélective

Les mesures du courant de fuite dans le cas de la précharge sélective sont présentées dans le tableau 3.4 et sont à comparer avec celles de la précharge par défaut du tableau 3.1. On remarque que la diminution du courant de fuite est d'environ 99.6%, soit une diminution de 2.5 décades sur la caractéristique $I(V_{gs})$. Ceci montre que la précharge sélective possède aussi un impact sur la réduction de la consommation statique en agissant sur le courant sous le seuil en plus de son influence sur la réduction de la consommation dynamique Cette réduction apparaîtra d'autant plus significative que le nombre de transistor NMOS est grand par colonne. Cette technique s'apparente à celle présentée à la section 1.2.5 (Polarisation auto-inversée [Kaw1993]), page 15, mais présente l'avantage de ne rajouter aucun transistor supplémentaire, puisque l'on se sert du transistor de précharge.

Température	$30°C$			$85°C$		
Vdd	$3V$	$3.3V$	$3.6V$	$3V$	$3.3V$	$3.6V$
Min. (A)	$1.37e-15$	$1.81e-15$	$1.95e-15$	$1.31e-14$	$1.32e-14$	$1.33e-14$
Med. (A)	$1.6e-15$	$2.16e-15$	$2.46e-15$	$4.55e-14$	$5.16e-14$	$5.72e-14$
Max. (A)	$2.17e-15$	$2.67e-15$	$3.09e-15$	$8.73e-14$	$9.92e-14$	$1.11e-13$

TAB. 3.4 – Courants de fuite à travers un point mémoire de type ROM en utilisant la précharge sélective

La figure 3.11 résume les valeurs des tableaux précédents pour les valeurs médianes. Le gain obtenu par la précharge sélective permet de diviser le courant par un facteur compris entre 300 et 400.

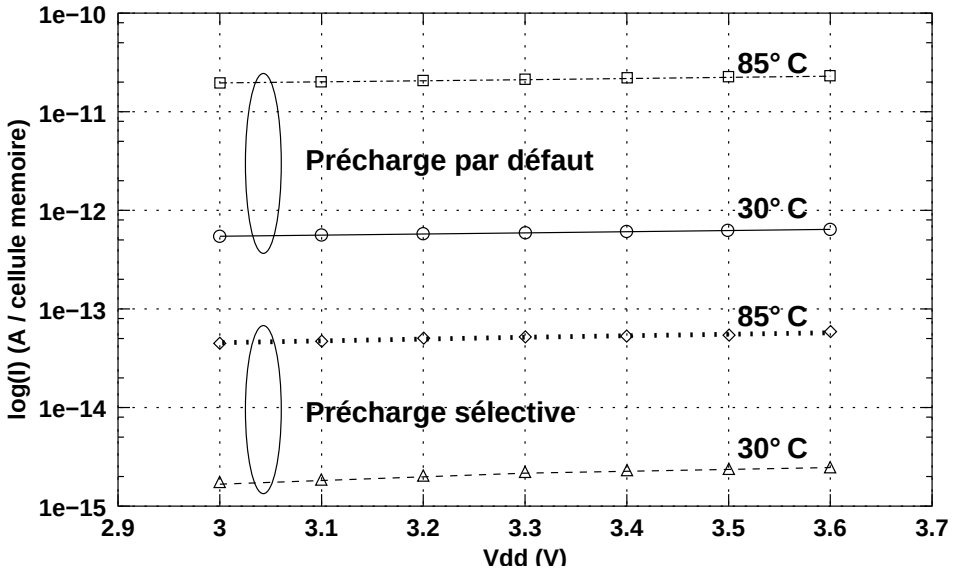


FIG. 3.11 – Comparaison entre précharge par défaut et précharge sélective dans le cas des ROMs.

Points mémoires de type SRAM

Pour les points mémoires de type SRAM, l'utilisation de la précharge sélective pour la réduction des courants de fuite s'avère moins efficace que dans le cas des ROMs. Le rapport des courants entre la précharge sélective et la précharge active en mode repos est compris entre 15% et 20% (Tableaux 3.2 et 3.5), soit une diminution maximale de 0.1 décade, l'essentiel du courant de fuite provenant de l'inverseur rebouclé du point mémoire (Voir figure 3.7) La précharge sélective jouant seulement sur le courant allant des transistors de précharge vers les transistors de passage des points mémoire, les fuites à travers l'inverseur rebouclé ne sont pas diminuées. Nous avons aussi réalisé cette mesure sur un autre plan mémoire de SRAM avec des transistors de passage possédant une largeur plus importante. Dans ce cas la précharge sélective est un plus efficace, elle est donc à mettre en relation avec la taille des transistors de passage pour des cellules de type SRAM : plus la largeur des transistors de passage est importante, plus la précharge sélective est efficace sur les SRAMs.

Température	$30°C$			$85°C$		
Vdd	$3V$	$3.3V$	$3.6V$	$3V$	$3.3V$	$3.6V$
Min. (A)	$6.48e-13$	$7.29e-13$	$8.51e-13$	$2.30e-11$	$2.49e-11$	$2.74e-11$
Med. (A)	$1.10e-12$	$1.32e-12$	$1.61e-12$	$3.39e-11$	$3.85e-11$	$4.64e-11$
Max. (A)	$2.61e-12$	$3.55e-12$	$6.42e-12$	$6.08e-11$	$8.02e-11$	$1.12e-10$

TAB. 3.5 – Courants de fuite à travers un point mémoire de type SRAM en utilisant la précharge sélective

La figure 3.12 résume les valeurs des tableaux précédents en reprenant les valeurs

médianes. Le gain obtenu par la précharge sélective permet de diviser le courant par un facteur environ égal à 1.2.

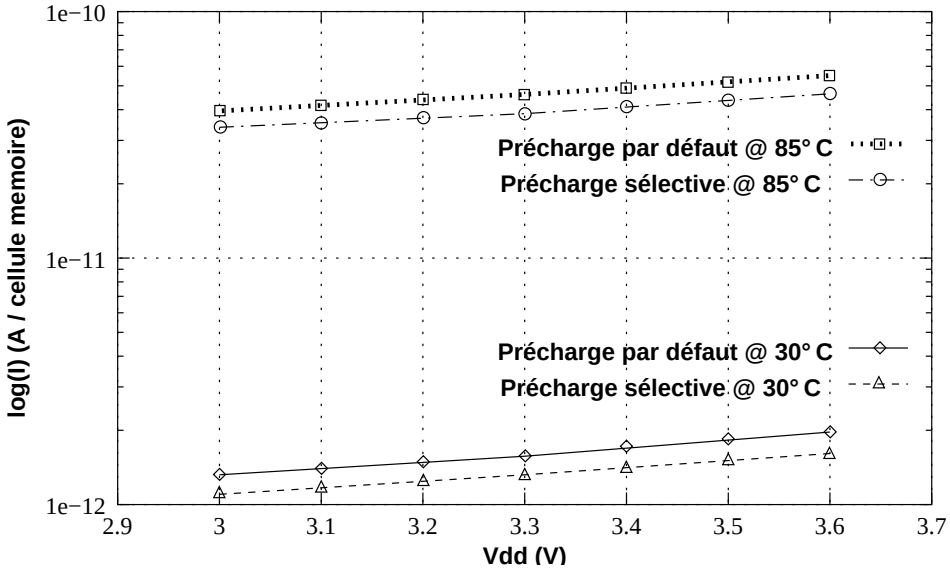


FIG. 3.12 – Comparaison entre précharge par défaut et précharge sélective dans le cas des SRAMs.

Points mémoires de type DPR

De même que pour les cellules de type SRAM dont s'inspirent celles de type DPRAM, la précharge sélective ne présente pas de réduction importante des courants de fuite puisque le rapport entre courant de fuite avec précharge sélective et précharge active n'est que de 40%, ce qui représente un gain de 0.2 décades seulement (Tableaux 3.3 et 3.6). Ce gain est un plus important que dans le cas des SRAM car le nombre de transistors de passage, sur lesquels cette technique est efficace, est ici doublé (Figure 3.8).

Température	$30°C$			$85°C$		
Vdd	$3V$	$3.3V$	$3.6V$	$3V$	$3.3V$	$3.6V$
Min. (A)	$7.64e-13$	$8.15e-13$	$8.83e-13$	$3.72e-11$	$3.95e-11$	$4.22e-11$
Med. (A)	$9.44e-13$	$1.02e-12$	$1.11e-12$	$4.28e-11$	$4.57e-11$	$4.91e-11$
Max. (A)	$1.27e-12$	$1.38e-12$	$1.56e-12$	$5.64e-11$	$6.02e-11$	$6.51e-11$

TAB. 3.6 – Courants de fuite à travers un point mémoire de type DPRAM en utilisant la précharge sélective

La figure 3.13 résume les valeurs des tableaux précédents pour les valeurs médianes. Le gain obtenu par la précharge sélective permet de diviser le courant par un facteur

compris entre 1.6 et 1.8. L'inverseur rebouclé du point fuit de la même manière qu'avec une précharge par défaut.

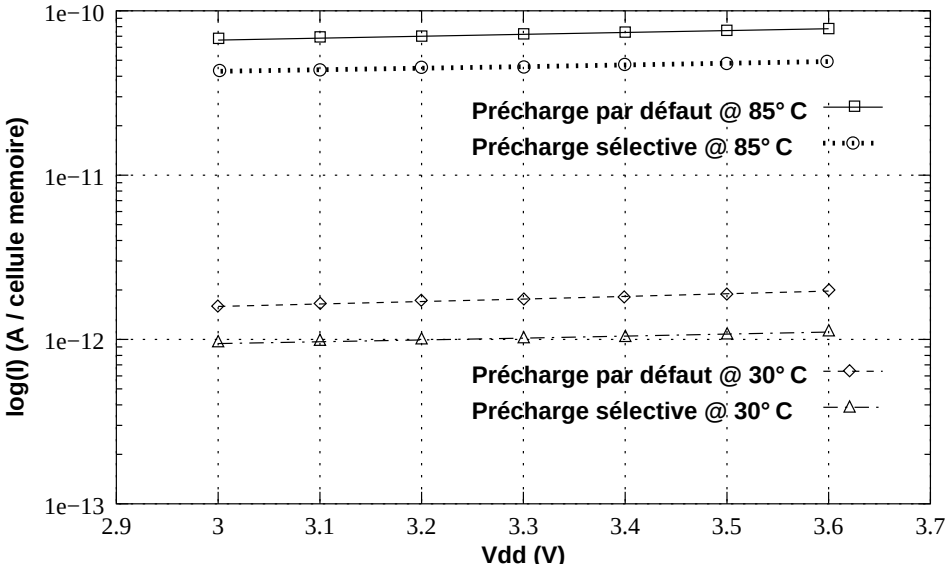


FIG. 3.13 – Comparaison entre précharge par défaut et précharge sélective dans le cas des DPRAMs.

3.1.4 Effets de la polarisation du substrat sur les courants de fuite

Une des possibilités pour limiter les courants de fuite, est de modifier la valeur du V_t des transistors, en polarisant le substrat de manière non habituelle, pendant le mode repos. Pour cela, nous avons modifié le schéma des points mémoires comme le montre la figure 3.14 pour un point mémoire de type SRAM. La polarisation des PMOS (en rouge) est dissociée de l'alimentation de la précharge et de l'inverseur rebouclé, de même que celle des NMOS est dissociée de l'alimentation de l'inverseur rebouclé. Les cinq carrés de la figure, représentent des plots sur lesquels les pointes du testeur viendront se placer. Les courants sont mesurés pour différentes valeurs de polarisation du substrat variant entre $0.4V$ et $0.8V$.

Ces modifications ont été apportées respectivement aux autres types de mémoires ROMs et DPRAMs étudiés précédemment. Dans tous les cas de figures observés, la polarisation du substrat est bénéfique pour la réduction du courant sous le seuil. D'ailleurs, c'est ainsi que l'on peut mesurer la contribution du courant de polarisation de diode inverse, puisque que le courant sous le seuil est éliminé. En revanche, le courant de polarisation de diode inverse a tendance à augmenter puisque la différence de potentiel aux bornes de la jonction PN augmente. La valeur de V_{bs} ne peut donc pas être modifiée infiniment [Mon1999]. La polarisation est modifiée, dans les mesures effectuées,

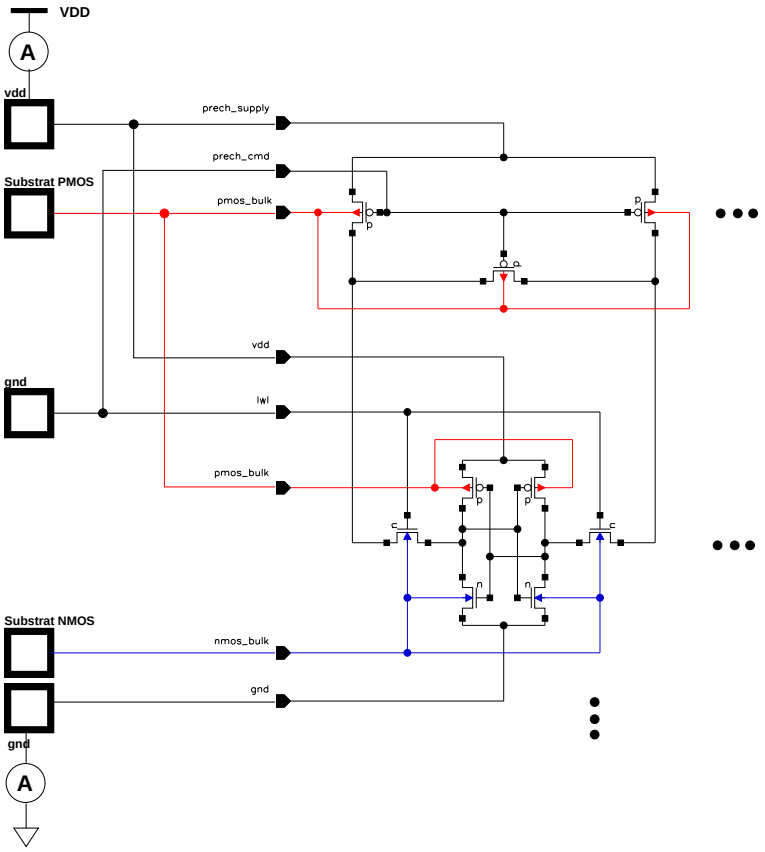


FIG. 3.14 – Montage utilisé pour mesurer l'influence de la polarisation du substrat sur la réduction du courant sous le seuil dans les points mémoires de type SRAM

de $0.4V$ et de $0.8V$, c'est-à-dire $Vdd+0.4V$ pour les Pmos et $Gnd-0.4V$ pour les Nmos. Le gain obtenu est présenté dans la table 3.7.

Gain en pourcentage	Polarisation du substrat	Précharge Sélective
SRAM	80%	20%
DPRAM	94%	37%
ROM	87%	99.7%

Gain par décade	Polarisation du substrat	Précharge Sélective
SRAM	-0.7	-0.1
DPRAM	-1.2	-0.2
ROM	-0.9	-2.5

TAB. 3.7 – Comparaison des courants de fuite réduits par la polarisation du substrat ou la précharge sélective

Les figures 3.15 à 3.17 présentent une comparaison entre la précharge sélective et la polarisation de substrat pour les trois types de mémoires étudiées. Pour les ROMs la précharge sélective est beaucoup plus efficace que la polarisation de substrat. Il faut savoir que la polarisation de substrat est coûteuse en terme de technologie puisqu'elle nécessite l'emploi d'un procédé à trois caisson (Triple-well). De plus, les temps de passage d'un mode à l'autre (Entre mode repos et mode actif) sont longs ($\sim 50ns$ pour passer du mode actif au mode repos et $\sim 370ns$ pour revenir au mode actif). De plus, cette technique requière la mise en œuvre de générateurs de tensions internes [Miz1999].

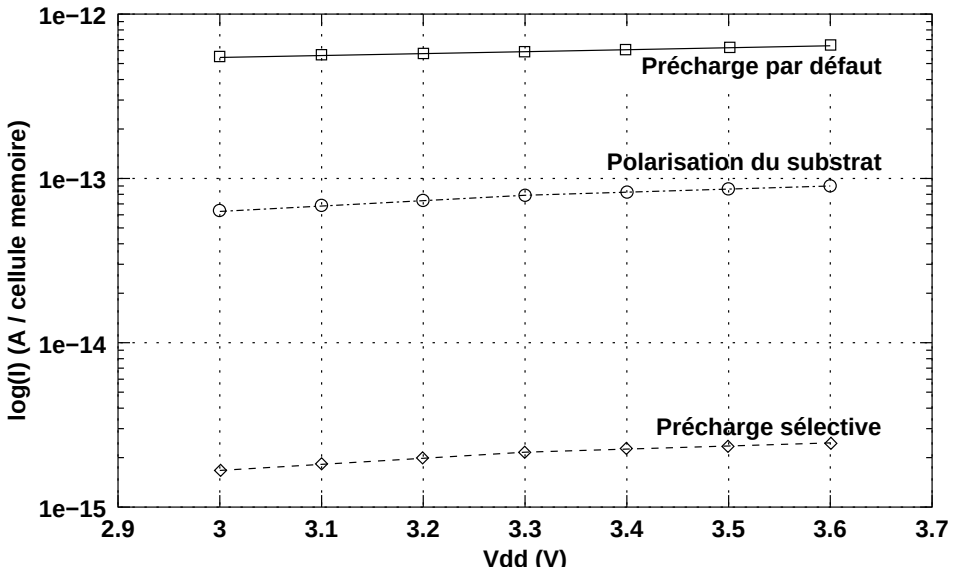


FIG. 3.15 – Comparaison entre les gains obtenus par la précharge sélective et la polarisation du substrat dans le cas des ROMs.

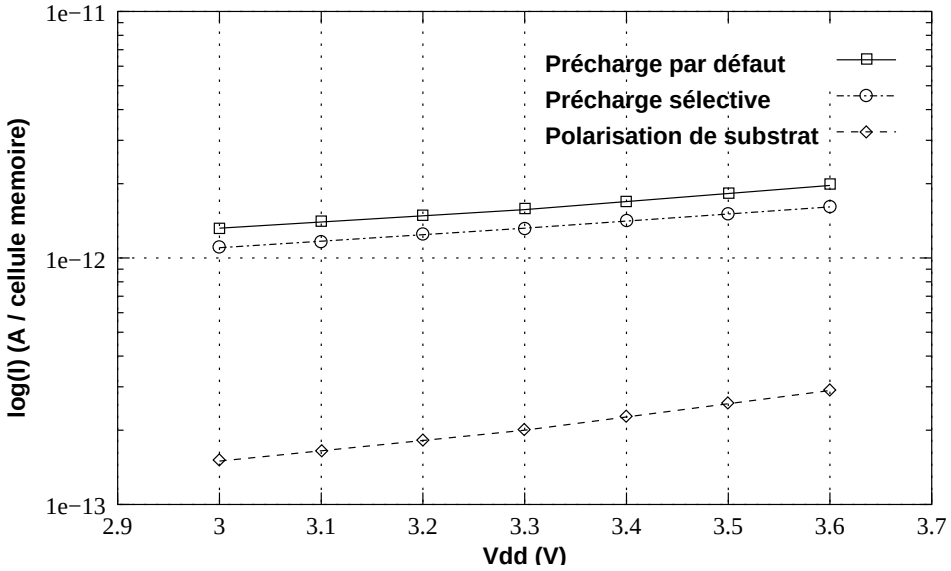


FIG. 3.16 – Comparaison entre les gains obtenus par la précharge sélective et la polarisation du substrat dans le cas des SRAMs.

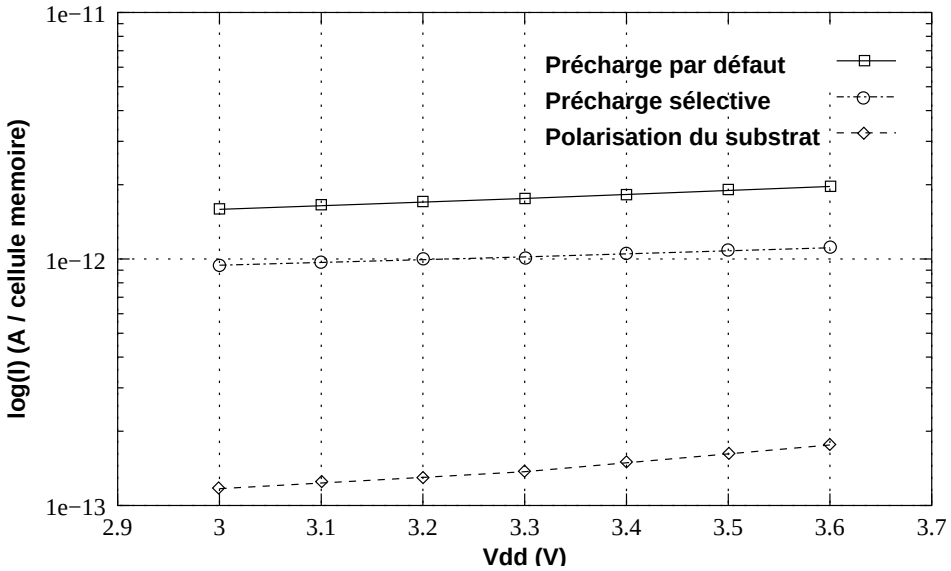


FIG. 3.17 – Comparaison entre les gains obtenus par la précharge sélective et la polarisation du substrat dans le cas des DPRAMs.

3.2 Comportement dynamique des mémoires à basse tension

3.2.1 Fonctionnement des transistors à basse tension

En diminuant la tension d'alimentation on cherche à réduire la puissance dynamique par le biais de la formule suivante : $P = f \times C \times V^2$. C'est ainsi que depuis plusieurs années, la tension d'alimentation des circuits baisse avec l'évolution de la technologie, sans que la tension de seuil diminue dans les mêmes proportions comme le montre la figure 3.18. Ce qui n'est pas sans danger pour le maintient des marges de bruit. De plus, en se rapprochant de la tension de seuil V_t, la valeur du courant de saturation devient comparable à celle des courants de fuite, puisque l'on est à la limite entre les zones de faible et forte inversion. Le principal inconvénient de la diminution de la tension d'alimentation est la perte en vitesse de fonctionnement du circuit.

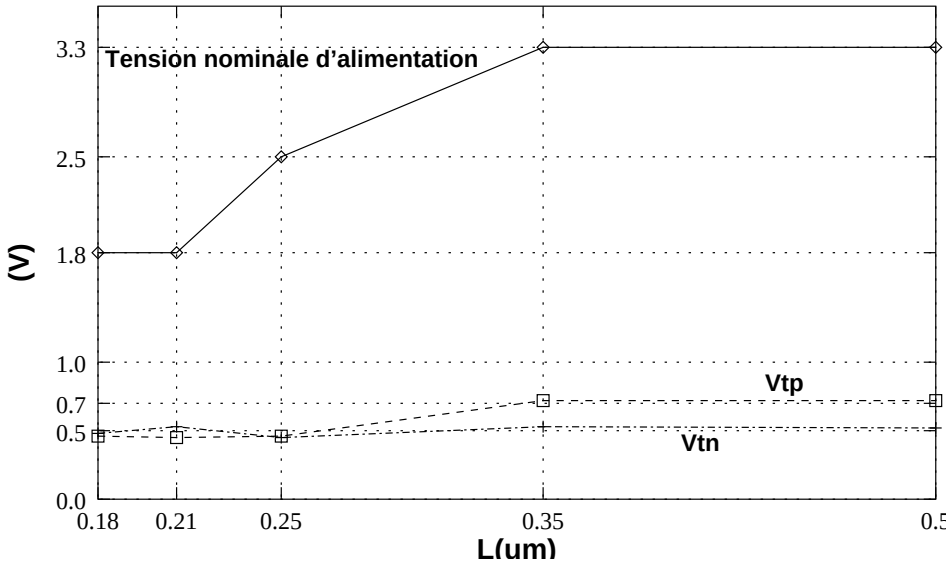


FIG. 3.18 – Évolution des tensions d'alimentation et de seuil en fonction de la technologie.

3.2.2 Étude d'une SRAM

Afin de réaliser des mémoires SRAM fonctionnant à faible tension d'alimentation ($0.9V$ puis à $0.65V$ en technologie $0.25\mu m$ et $0.18\mu m$), on souhaite identifier les limites de tension minimales de fonctionnement et de rétention de l'information et apporter des optimisations pour atteindre des tensions au dessous de 1V. Pour cela nous avons étudié

une SRAM en technologie $0.25\mu m$, de taille 16Kx16, dont nous avons progressivement baissé la tension d'alimentation. La conséquence immédiate de cet abaissement de la tension, est l'allongement des temps de propagation comme en témoigne la figure 3.19.

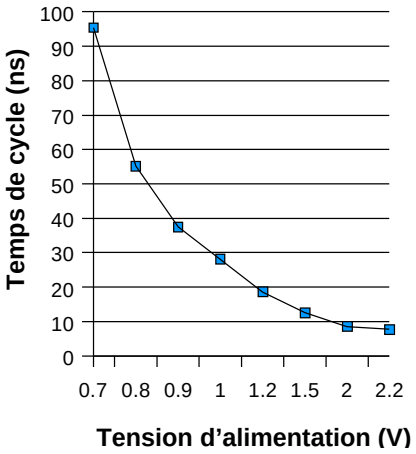


FIG. 3.19 – Évolution du temps de cycle pour une mémoire SRAM 16Kx16 en technologie $0.25\mu m$.

Amplitude maximale des signaux (Full swing)

Avec la basse tension, la tension de fonctionnement se rapproche des tensions de seuil. Aussi, il est souhaitable que les signaux puissent évoluer avec une amplitude maximale. La logique de décodage et la partie de contrôle étant réalisées en technologie CMOS complémentaire, il n'y a pas de risque à en baisser la tension. En revanche, pour le chemin de lecture et d'écriture, la logique est parfois réalisée à l'aide de porte de passage non complémentaires ce qui risque de poser des problèmes. Un chemin d'écriture classique est représenté à la figure 3.20.

Les transistors de passage du point mémoire sont de type NMOS, ainsi lors de l'écriture on aura d'un côté du point mémoire, un potentiel à 0 et de l'autre, un potentiel à $V_{dd}-V_t$. Bien que les deux potentiels n'aient pas une différence égale à V_{dd}, l'écriture se fait correctement grâce au rebouclage des inverseurs. Remplacer ces transistors de passage par des portes de passage complémentaires n'auraient donc aucun intérêt. La marge de bruit sur le point mémoire n'est pas remise en cause puisque les niveaux d'entrée sur le point mémoires sont déjà dégradés par les transistors de passage du point mémoire lui-même.

Les transistors de passage d'écriture sont eux aussi de type NMOS, ce qui ne modifie en rien les niveaux décrits précédemment. Ces transistors ne peuvent être remplacés par des PMOS, car dans ce cas, on aurait de part et d'autre du point mémoire des tensions égales à V_t et $V_{dd} - V_t$ ce qui rendrait l'écriture plus difficile étant donnée la faible différence de potentiel entre les deux nœuds. Enfin, leur remplacement par des portes

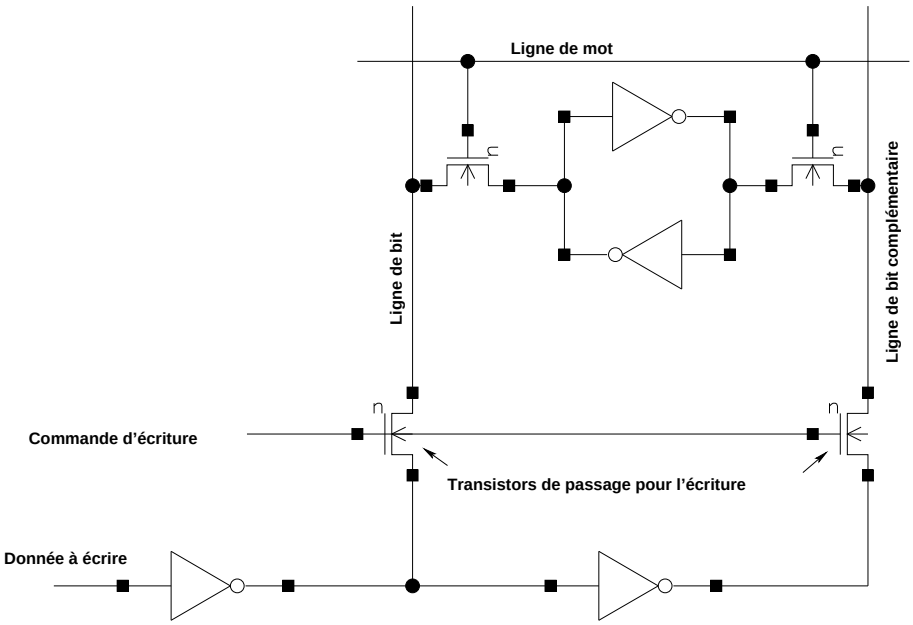


FIG. 3.20 – Chemin d'écriture dans une SRAM

complémentaire n'aurait pas d'intérêt, puisque le niveau V_{dd} serait filtré par le transistor de passage du plan mémoire. Ainsi, bien que l'on veuille fonctionner à basse tension, les portes de passage complémentaires ne sont pas nécessaires pour le circuit d'écriture.

Architecture de lecture sans amplificateur de lecture

Dans les SRAMs, un amplificateur de lecture vient accélérer la répartition de charges lors de la lecture. Cette structure amplifie la différence de potentiel entre les deux lignes de bit. En abaissant la tension de fonctionnement, la marge signal à bruit diminue et l'amplificateur risque de se déclencher sans qu'une lecture ait réellement commencée. Ainsi, pour sécuriser le schéma de lecture, on se propose de supprimer cet amplificateur (Figure 3.21). Seul le point mémoire va réaliser la répartition de charge, ce qui va allonger le temps de lecture. Au départ, les transistors de passage pour la lecture sont de type PMOS. Cela représente un inconvénient : à la fin de la répartition de charge, on a respectivement V_{dd} et $V_{dd} - V_t$ sur les lignes de lecture, ce qui ne représente pas un bon niveau pour les commandes de l'étage trois-états. En remplaçant ces transistors par des transistors NMOS, on aurait respectivement V_{dd} et 0 en prenant le soin de précharger les lignes de lecture avant l'opération de lecture.

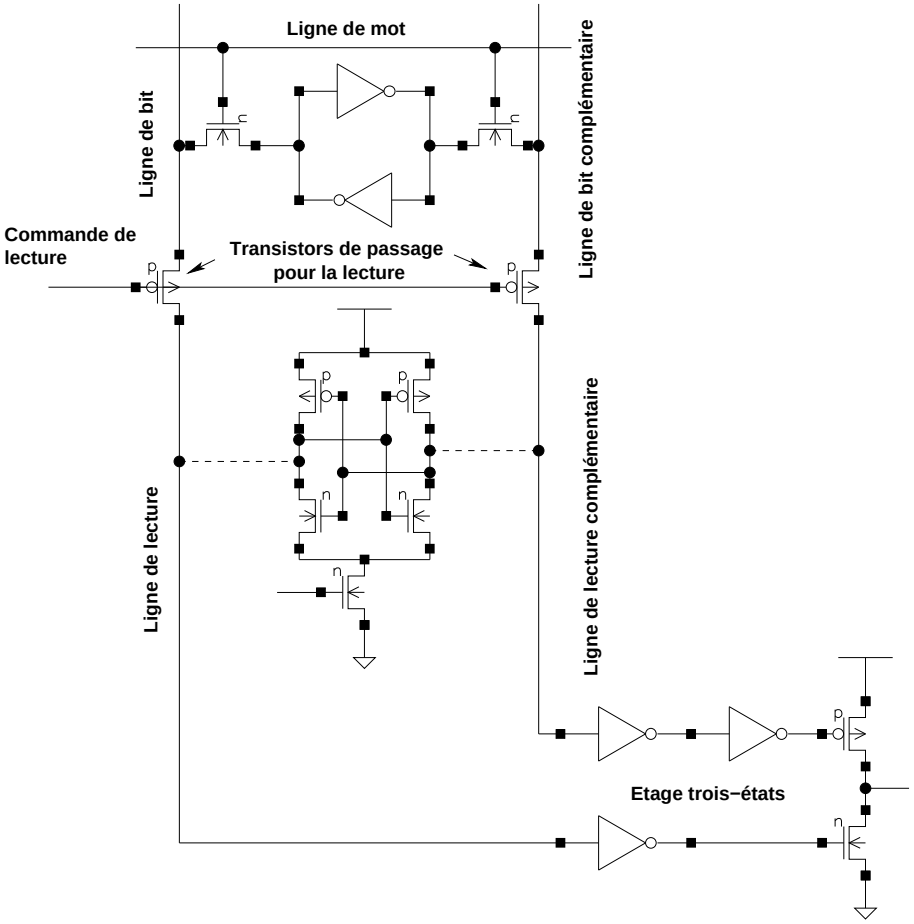


FIG. 3.21 – Chemin de lecture

3.2.3 Conclusion

Dans ce chapitre, nous avons expliqué comment simuler les courants de fuites avec Spice. Nous avons décrit les courbes que l'on pouvait attendre de ces simulations ainsi que le paramétrage à respecter pour les obtenir. Ensuite, nous avons étudié les effets de la précharge sélective sur différents types de mémoires avec une division par un facteur compris entre 300 et 400 du courant de fuite pour les ROMs. Nous avons aussi étudié la polarisation par le substrat pour la comparer avec la précharge sélective. Enfin, nous avons analysé le comportement d'une SRAM qui fonctionne à une tension basse, proche de la tension de seuil en précisant les modifications éventuelles à apporter pour assurer un meilleur fonctionnement dans ces conditions d'utilisation inhabituelles.

Pour fonctionner à des tensions encore plus basses, une approche intéressante a été analysée par H. Soeleman et K. Roy [Soe1999], [Soe2000], où les courants de fuite sont utilisés pour opérer au niveau logique, à la place des courants de saturation. Pour le moment leur étude n'a été réalisée que sous la forme de simulation Spice des courants de fuite avec un modèle du fondeur TSMC. Un testchip en préparation permettra de mesurer l'efficacité de cette approche.

CHAPITRE 4

Modélisation et caractérisation du délai et de la consommation

LA caractérisation de toutes les mémoires issues d'un générateur n'étant pas possible à partir de layouts totalement extraits avec la représentation de toutes les capacités parasites, étant donné les tailles de ces mémoires. Il a donc fallu développer un modèle pour les représenter sous la forme d'un chemin critique. Ce modèle doit aussi être facilement paramétrable afin de conserver un nombre réduit de chemins critiques quelques soient les variations des mémoires en terme de lignes, de colonnes, de blocs et de bits par mots.

4.1 Modélisation et simulation électrique d'une mémoire

4.1.1 Extraction Complète

La caractérisation d'une mémoire, peut être obtenue en simulant électriquement le comportement d'un layout complètement extrait de cette mémoire. Cette méthode a l'avantage de la précision mais est extrêmement coûteuse en ressources matérielles : mémoire, disque, accès réseau. Avec cette méthode, tous les transistors sont extraits avec tous leurs paramètres (w, l, aire et périmètre) ainsi que toutes les capacités parasites. Le nombre d'éléments de la netlist obtenue devient rapidement important au fur et à mesure que l'on simule des mémoire de grande capacité en terme de nombre de nœuds et de nombre de transistors. De plus, cette méthode n'est pas recommandée pour la caractérisation de mémoires issues de générateurs, puisqu'il faut générer, puis extraire un layout différent pour chaque configuration à simuler.

4.1.2 Extraction partielle

Les mémoires étant souvent constituées d'éléments répétitifs, les points mémoire notamment, excepté pour la partie de contrôle, il n'est pas nécessaire de simuler la mémoire entièrement pour la caractériser. Seules les parties critiques entrant en ligne de compte pour la caractérisation des délais et de puissance consommée sont utiles. De cette manière, on peut simplifier l'approche précédente, en ne prenant en compte qu'une partie du layout de la mémoire, qui sera extrait et simulé. Cependant, il se peut que la réduction de la taille de la netlist ne soit pas suffisante avec cette méthode. De plus, elle n'est pas pratique à mettre en œuvre pour la caractérisation d'un générateur. En effet, lors des différentes simulations, on souhaite faire varier les caractéristiques de la mémoire telles que le nombre de lignes, de colonnes, de blocs, de nombre de bits par mots, etc... Il est donc difficilement envisageable de générer, puis d'extraire à chaque fois un layout différent.

4.1.3 Chemin critique paramétrable

La génération d'un layout ne peut être réalisée que vers la fin de la période de conception. Aussi, au lieu d'un layout extrait, nous utilisons un schéma rétro-annoté (sur lequel, on reporte toutes les capacités parasites), qui représente le chemin suivi par les signaux critiques entrant en ligne de compte pour déterminer les différents temps et la puissance consommée. Il est ainsi possible, avec un minimum de connaissances sur le layout final, de pouvoir commencer la conception électrique d'un circuit avec une assez bonne précision par rapport au circuit final. On appellera ce schéma le chemin critique. Comme le schéma ne représente qu'une partie de la mémoire, la taille de la netlist est diminuée et le temps de simulation est ainsi réduit.

Dans la plupart des cas, lorsque l'on passe d'une configuration à une autre, les changements ont lieu sur les capacités de routage et de couplage ainsi que sur le nombre de grilles ou de drains vus par un nœud. Avec un schéma rétro-annoté, il est possible de paramétrer l'ensemble des parasites en fonction des caractéristiques de la mémoire en appliquant, par exemple, un principe de proportionnalité pour les capacités. Un exemple de chemin critique simplifié pour une ROM est présenté à la figure 4.1. On y a modélisé la lecture d'un point mémoire en ne décrivant que le "matériel" strictement nécessaire à cette simulation. Le chemin emprunté part du contrôle sur commutation du signal d'horloge Clk, arrive sur le décodeur de ligne représenté par une porte "ET", il suit la ligne de mot pour activer la grille du transistor NMOS du point mémoire et finit par traverser l'amplificateur de lecture, le multiplexeur puis l'étage de sortie qui présente la donnée lue sur le bus de sortie.

Toutes les capacités représentées sur le schéma ont une valeur fonction du nombre de lignes et de colonnes. Aussi, nous ne mettons aucune valeur numérique directe dans notre schéma pour les valeurs des capacités mais une formule, comme par exemple :

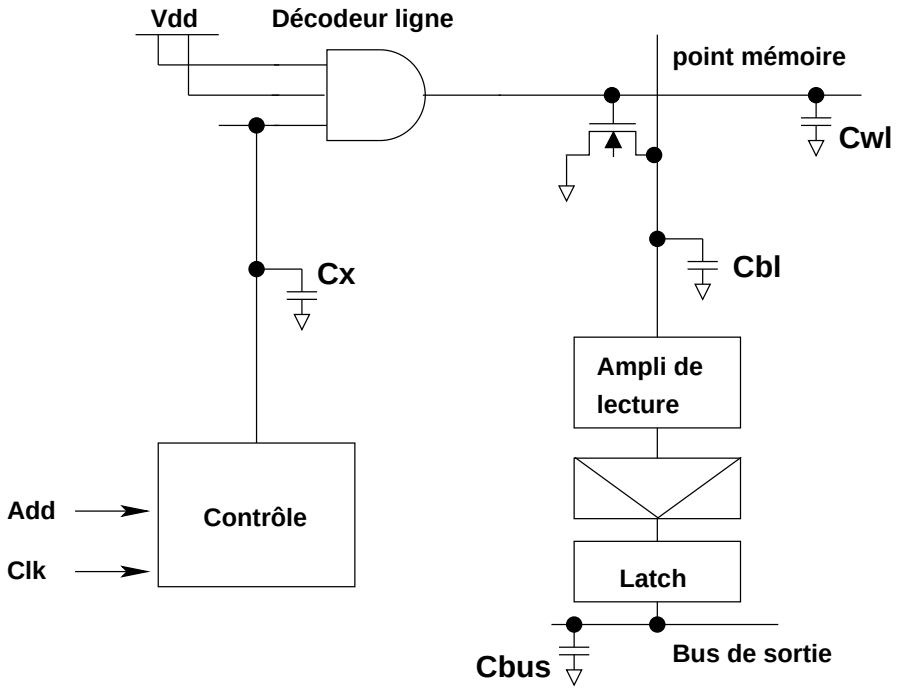


FIG. 4.1 – Schéma simplifié d'un chemin critique pour une ROM

$C_{wl} = C_\alpha + C_\beta \times nb\,de\,colonnes$. Pour réaliser nos schémas nous utilisons l'outil d'édition de schémas de la chaîne de CAO Cadence (Composer-Schematic). Chaque netlist issue d'un schéma est analysée puis ré-écrite par un programme, afin de calculer la valeur des paramètres avant la simulation définitive comme le montre le flot de simulation présenté à la figure 4.2. La netlist temporaire est créée une fois pour toutes puis, à chaque nouvelle simulation, elle est enrichie par les valeurs contenues dans le fichier de paramètres . Avec cette technique, le schéma de départ n'est pas modifié à chaque simulation, ce qui limite le nombre d'erreurs apportées au schéma entre plusieurs simulations pour différentes configurations ou conditions de simulation.

4.1.4 Modélisation des capacités dynamiques

Si les capacités entre niveaux physiques (routage, couplage), peuvent être exprimées de manière linéique et/ou surfacique, les capacité liées aux transistors sont d'ordre dynamiques (Capacités de grille, drain et source) : elles dépendent notamment de la température et de la tension d'alimentation. Elles nécessitent d'être prises en compte différemment. C'est ce que nous proposons à travers l'utilisation de générateurs de courant commandés en exploitant la redondance des différents éléments d'une mémoire. Comme les mémoires sont souvent divisées en blocs, il est facile de ne simuler qu'un seul bloc et de représenter la charge induite par les grilles ou les drains des autres blocs attachés à un nœud par l'utilisation de générateurs de courant : le courant est mesuré

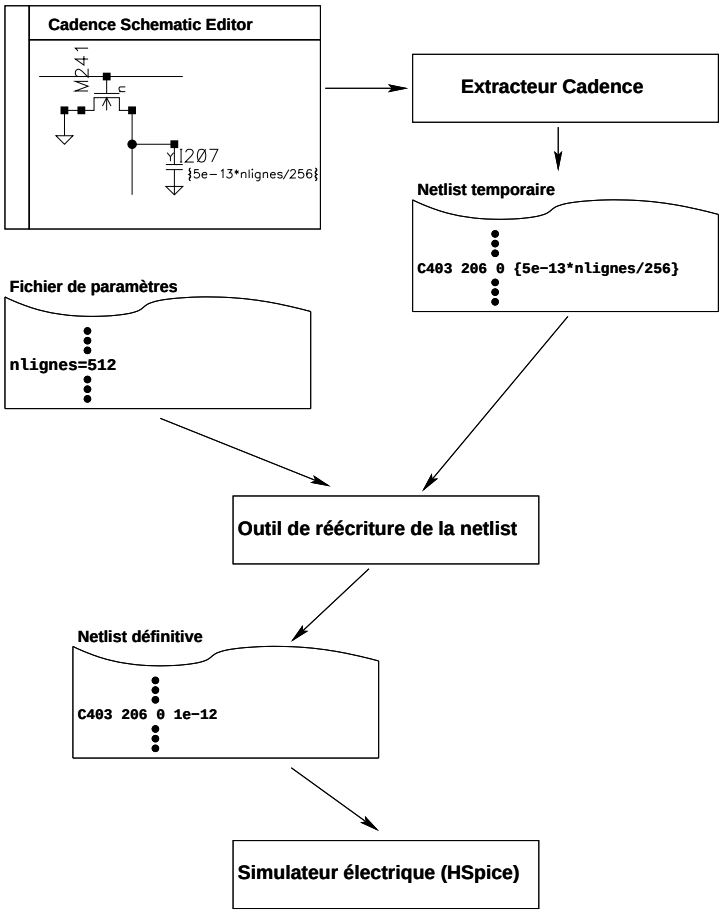


FIG. 4.2 – Flot de simulation utilisant des valeurs paramétrées dans la netlist

sur un nœud et ensuite multiplié par un facteur représentant le nombre de grilles ou de drains rattachés à ce nœud dans la réalité.

De même, à l'intérieur du même bloc, on a souvent le cas d'une porte attaquant plusieurs grilles ou drains pour un niveau de hiérarchie donnée, comme le montre la figure 4.3. Un courant i est fourni par une porte et se réparti en plusieurs courant i_k identiques puisque toutes les cellules sont rigoureusement les mêmes. On appelle C_{rout}, la capacité de routage globale du nœud étudié et C_{rin} la capacité de routage interne à chaque cellule feuille.

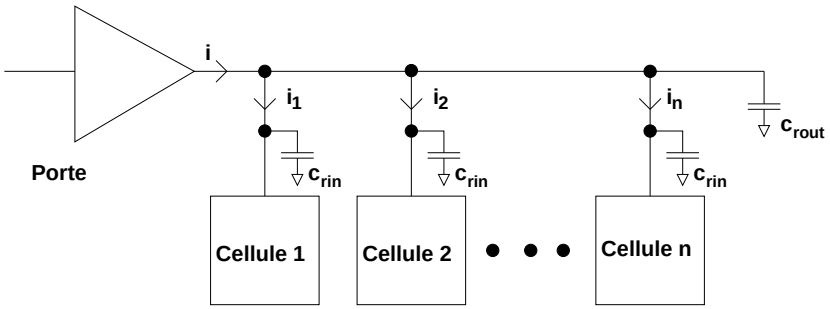


FIG. 4.3 – Une porte attaquant n cellules identiques

Si l'on souhaite ne garder qu'une seule cellule utile pour la simulation, on peut reprendre la méthode exposée au paragraphe précédent pour modéliser les capacités des autres blocs qui ont été retirés comme indiqué sur la figure 4.4. On conserve la capacité globale de routage C_{rout} et on représente la capacité des cellules manquantes par une capacité de valeur $(n-1) \times (C_d + C_{rin})$, où C_d représente la valeur de la capacité du ou des transistors de chaque cellule, rattachés au nœud étudié. C_d peut être soit une capacité de grille, soit une capacité de drain ou de source.

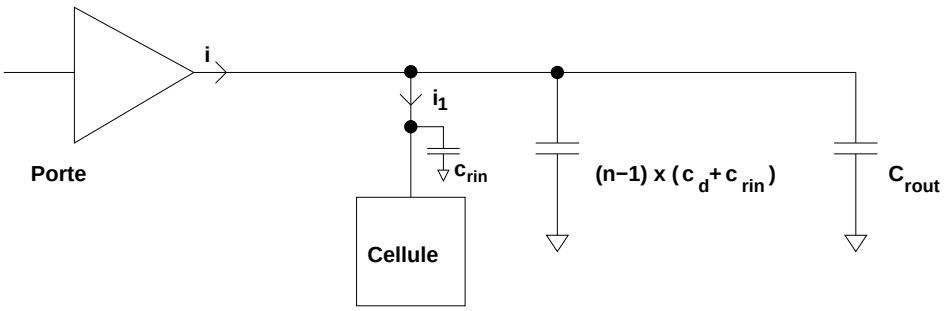


FIG. 4.4 – Modélisation des capacités à valeur statique

Cependant, cette modélisation ne prend pas bien en compte les valeurs possible de C_d : si les valeurs des capacités C_{rout} et C_{rin} peuvent être fixées d'après les valeurs relevées sur un layout extrait, il n'en est pas de même pour C_d dont la valeur dépend notamment de la température et des paramètres process [Wes1994, page 180]. Aussi, pour remédier à ce problème, nous proposons de modéliser les capacités C_d et C_{rin} par un même dispositif, basé sur l'utilisation de générateurs de courant commandés

(Figure 4.5). Étant donné que tous les blocs sont identiques, le courant fourni par la porte est égal à $n \times i_1$. Comme on ne prend en compte qu'une seule cellule feuille, le courant débité par le générateur de courant est égal à $(n-1) \times i_1$.

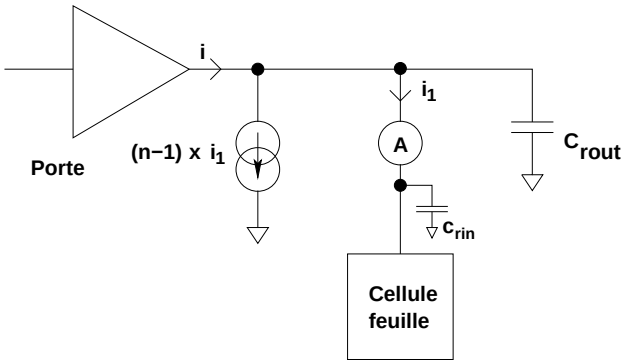


FIG. 4.5 – Modélisation par des générateurs de courant commandés

4.1.5 Comparaison des méthodes sur un circuit très simple

Sur un circuit élémentaire comprenant un inverseur attaquant 256 autres inverseurs, nous allons comparer 3 méthodes, en utilisant le simulateur HSpice, le but étant de mettre en évidence la précision de la méthode que nous avons retenue :

- Simulation du layout complet du circuit, ce qui constituera la référence (Figure 4.6).
- Simulation d'un schéma avec capacités parasites uniquement (Routage + grille) (Figure 4.7).
- Simulation d'un schéma avec capacités parasites (Routage) et générateur de courant (Figure 4.8).

On compare les trois méthodes sur 3 technologies selon 3 conditions de simulation, en mesurant le temps de propagation entre la commande au nœud A et la sortie de l'inverseur (Tableau 4.1) et l'énergie consommée pour la charge de ce nœud (Tableau 4.2). On appelle Réf. la méthode de référence avec le layout complètement extrait, M1, la méthode avec les capacités parasites (Routage+grilles) et M2, la méthode avec capacités parasites (Routage) et générateurs de courant.

On remarque que la méthode des générateurs de courant donne des résultats très précis pour des technologies différentes aussi bien en délai qu'en consommation. Par contre, lorsque l'on modélise les capacités dynamiques de grille, par une capacité parasite de valeur fixe, on introduit une erreur plus ou moins importante dans le calcul du délai et de la consommation. Ainsi, pour caractériser nos mémoires nous avons choisi d'utiliser la méthode des générateurs de courant commandés.

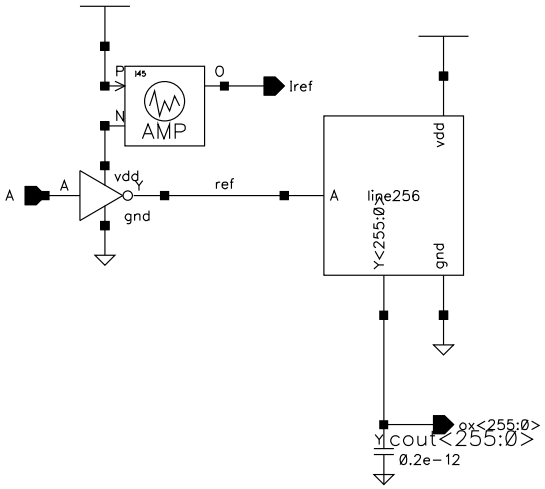


FIG. 4.6 – Layout complètement extrait

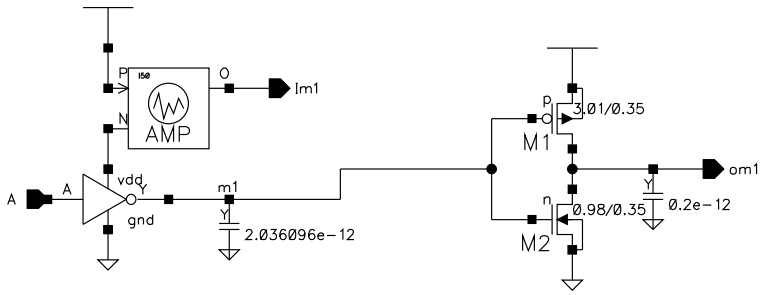


FIG. 4.7 – Capacités parasites uniquement (Routage + grilles)(M1)

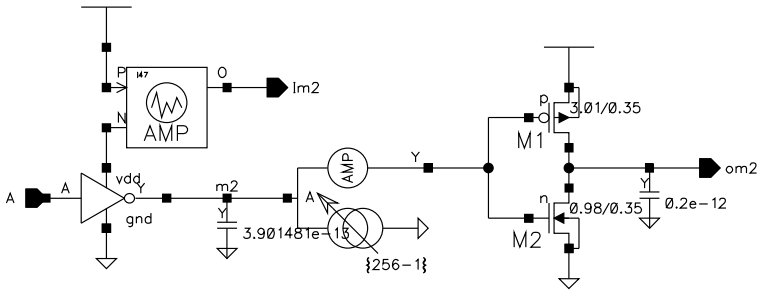


FIG. 4.8 – Capacités parasites (Routage) et générateur de courant (M2)

$0.5\mu m$	Fast (ns)	Gain (%)	Typ (ns)	Gain (%)	Slow (ns)	Gain (%)
Réf.	2.234	-	3.258	-	5.161	-
M1	2.155	-3.5	3.183	-2.3	5.253	$+1.8$
M2	2.256	$+1.0$	3.293	$+1.1$	5.222	$+1.2$
$0.35\mu m$	Fast (ns)	Gain (%)	Typ (ns)	Gain (%)	Slow (ns)	Gain (%)
Réf.	1.644	-	2.210	-	3.122	-
M1	1.887	$+14.8$	2.622	$+18.6$	3.902	$+25.0$
M2	1.637	-0.4	2.202	-0.4	3.112	-0.3
$0.25\mu m$	Fast (ns)	Gain (%)	Typ (ns)	Gain (%)	Slow (ns)	Gain (%)
Réf.	1.294	-	1.616	-	2.148	-
M1	1.107	-14.5	1.409	-12.8	1.917	-10.8
M2	1.290	-0.3	1.612	-0.2	2.144	-0.2

TAB. 4.1 – Comparaison des délais entre les méthodes de modélisation

$0.5\mu m$	Fast (pW/Hz)	Gain (%)	Typ (pW/Hz)	Gain (%)	Slow (pW/Hz)	Gain (%)
Réf.	47.675	-	39.687	-	28.671	-
M1	37.370	-21.6	30.897	-22.1	22.882	-20.2
M2	47.291	-0.8	39.360	-0.8	28.440	-0.8
$0.35\mu m$	Fast (pW/Hz)	Gain (%)	Typ (pW/Hz)	Gain (%)	Slow (pW/Hz)	Gain (%)
Réf.	29.822	-	23.922	-	17.142	-
M1	27.174	-8.9	22.209	-7.2	16.637	-2.9
M2	29.415	-1.4	23.603	-1.3	16.895	-1.4
$0.25\mu m$	Fast (pW/Hz)	Gain (%)	Typ (pW/Hz)	Gain (%)	Slow (pW/Hz)	Gain (%)
Réf.	12.818	-	10.195	-	7.617	-
M1	8.879	-30.7	7.071	-30.6	5.393	-29.2
M2	12.859	$+0.3$	10.230	$+0.3$	7.697	$+1.1$

TAB. 4.2 – Comparaison de l'énergie entre les méthodes de modélisation

4.2 Modélisation d'une ROM

Nous allons désormais étudier la façon d'utiliser le modèle retenu de la section précédente, afin de caractériser les mémoires présentées au chapitre 2.

4.2.1 Modélisation des blocs dans la ROM

L'architecture retenue est partitionnée en blocs. Afin de représenter les grilles des portes "NAND" vues par la sortie du décodeur de lignes, nous utilisons le schéma de la figure 4.9, où C_R représente la capacité de routage d'un bloc de la ligne de mot globale. Nous avons seulement représenté le schéma de la ligne de mot globale, mais un schéma similaire est appliqué aux autres signaux globaux puisque nous avons hiérarchisé tous les signaux sortant du contrôle.

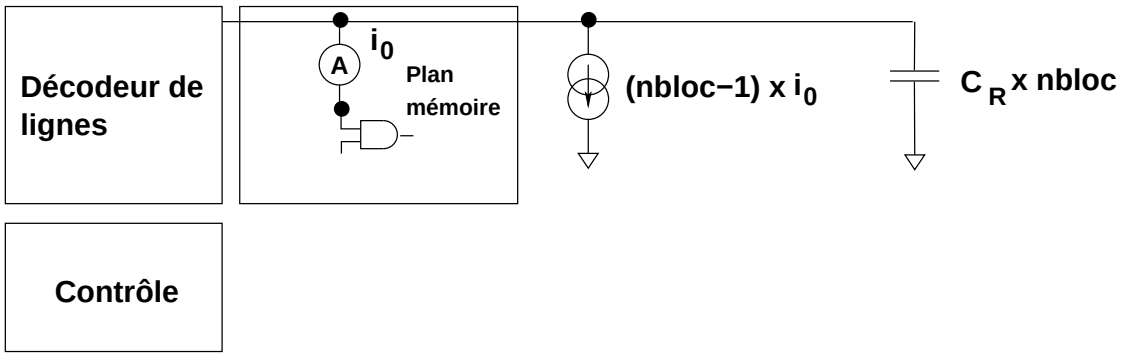


FIG. 4.9 – Modélisation de la ligne de mot globale

Le modèle est valide quelque soit le nombre de blocs. Ainsi, même si il n'y a qu'un seul bloc, la valeur du courant débité par le générateur de courant sera nulle, ce qui rend bien compte de la réalité.

4.2.2 Modélisation de la ligne de mot

Pour modéliser la ligne de mot qui est composée de n transistors et d'une capacité de routage $C_r \times n$ (Figure 4.10), n étant le nombre de colonnes, on utilise le modèle des générateurs de courant (Figure 4.11).

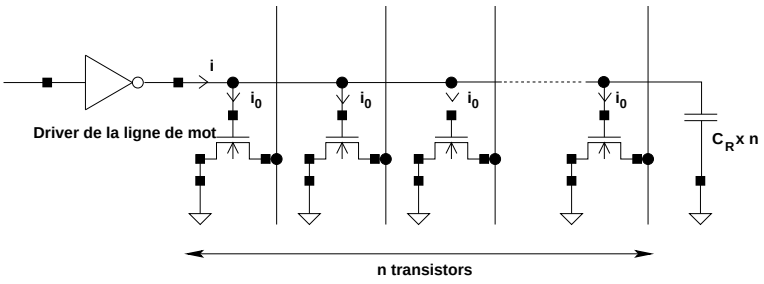


FIG. 4.10 – Ligne de mot à modéliser

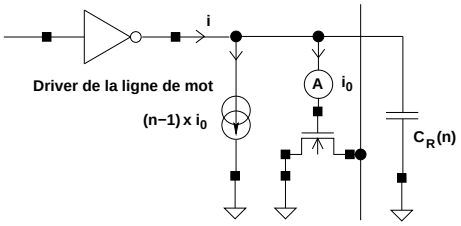


FIG. 4.11 – Modélisation de la ligne de mot

4.2.3 Modélisation de la ligne de bit

Au schéma de la ligne de mot, nous ajoutons la modélisation de la ligne de bit (Figure 4.12). La capacité de routage horizontale pour la ligne de mot est notée C_{rwl} et est proportionnelle au nombre de transistors NMOS. Pour la ligne de bit, nous ajoutons un deuxième générateur de courant commandé par un ampèremètre qui mesure le courant sur le drain d'un transistor d'une autre ligne, donc non sélectionnée au cours des simulations. La ligne de bit comporte elle aussi une capacité parasite $C_{bl}(m)$, proportionnelle à m, m étant le nombre de lignes. Nous avons représenté ici le schéma utilisé pour la modélisation du délai. Comme il s'agit d'un pire cas, il faut connecter le maximum de drains à la ligne de bit, c'est-à-dire que tous les points mémoires sont programmés, à zéro, par un transistor. Avec ce modèle, il est très facile de prendre en compte différents plan mémoires en terme de taille, puisqu'il suffit simplement de faire varier les paramètres n et m pour les modéliser correctement.

4.2.4 Modélisation du chemin de données

Enfin, pour modéliser le chemin de données (Précharge, multiplexeur et l'étage de sortie : amplificateur de lecture, latch, buffer), nous utilisons le schéma de la figure 4.13. Nous souhaitons connaître le délai pire cas, la consommation pour la lecture d'un '0' et la consommation dans le cas de la lecture d'un '1', avec une seule simulation. C'est ainsi que nous utilisons 3 lignes de bits : sur la ligne de bit 'd' pour le délai, le coefficient m du nombre de lignes est maximal, tandis que pour les lignes de bit '0'

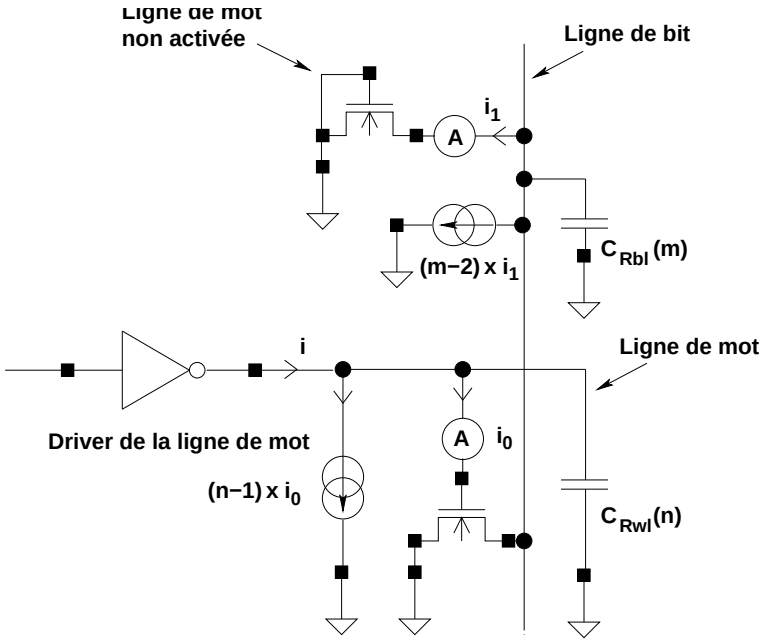


FIG. 4.12 – Modélisation de la ligne de bit

et '1', le coefficient est égal à la moitié du nombre total de lignes puisque l'on veut se placer dans un cas moyen pour la consommation.

Enfin, pour les drivers issus du contrôle, nous utilisons aussi des générateurs de courant afin de bien modéliser les différents blocs et colonnes vus par ces drivers. Sur la figure 4.13, les gains pour les générateurs de courant sont égaux à $d - 4$, où d est le nombre de chemins de données total de la mémoire. De cette valeur, on en soustrait 4, c'est-à-dire que l'on prend en compte les 3 chemins de données représentés sur le schéma, ainsi que les grilles des transistors sur lesquelles le courant est mesuré.

4.2.5 Modélisation globale de la ROM

La modélisation globale de la ROM est présentée à la figure 4.14. Nous y retrouvons les blocs précédemment exposés en plus du bloc de contrôle qui lui n'a pas été modélisé car il ne présente pas de régularité comme les autres parties. Nous avons placé un layout complètement extrait du bloc contrôle dans le schéma de simulation, ce qui ne retire rien à la précision de nos simulation, mais qui allonge un peu les temps de simulation et limite la flexibilité de notre schéma pour toutes les configurations de ROM que nous devons simuler pour caractériser le générateur de ROM.

Au cours d'une simulation, nous mesurons le délai et les différents courants à travers les ampèremètres dans chacun des blocs. Les différents délais, temps d'accès, temps de cycle, temps de "setup" et de "hold", sont mesurés avec un fichier de mesure au format Spice. Les courants sont imprimés dans des fichiers de sorties dont les valeurs

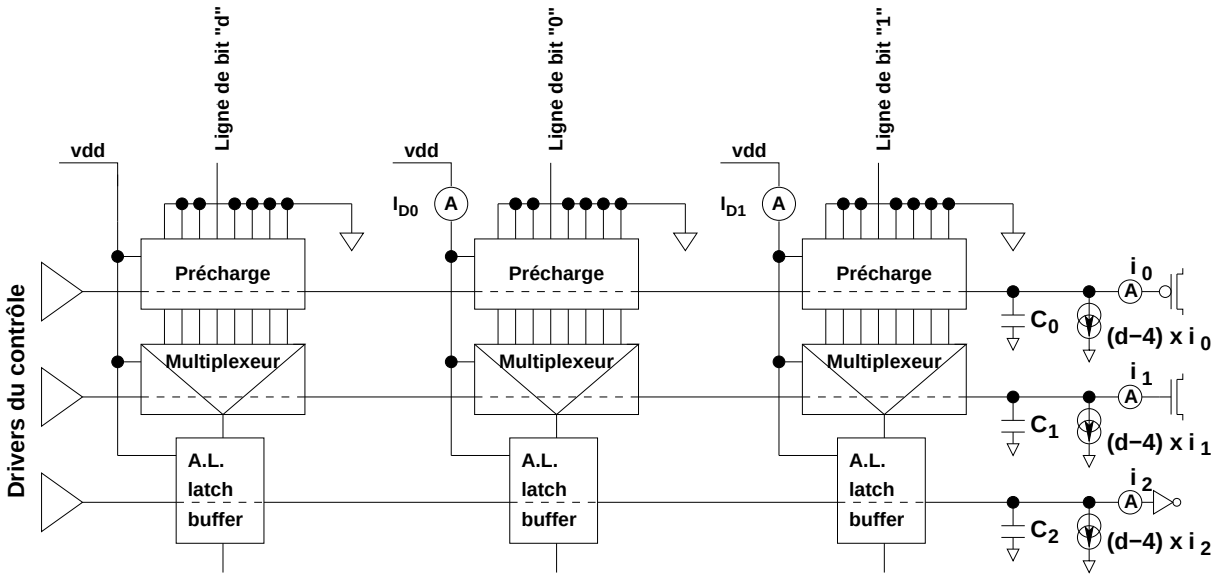


FIG. 4.13 – Modélisation du chemin de données

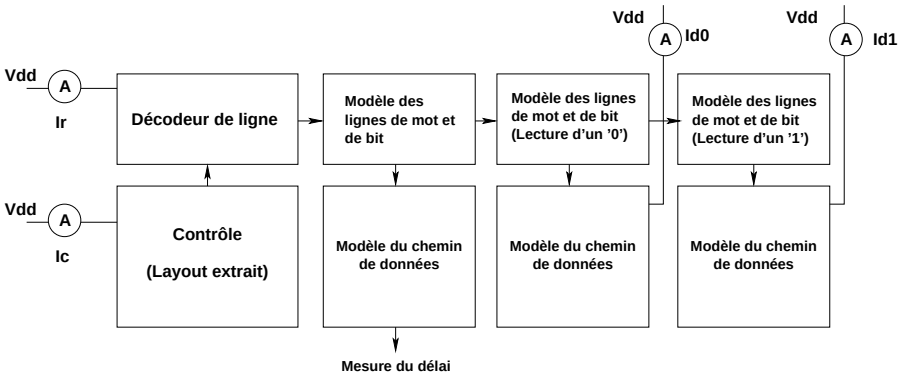


FIG. 4.14 – Schéma de simulation globale des ROMs

sont ensuite filtrées (suppression des courants négatifs), puis intégrés sur les périodes intéressantes pour la simulation. Les courants sont sommés de manière à connaître l'expression du courant total, en se réservant la possibilité de faire jouer la répartition entre le nombre de '0' et de '1' à l'intérieur d'un mot, comme le montre l'équation suivante :

$$I_{total} = I_C + I_R + \frac{nombre\ de\ '0'\ par\ mot}{nombre\ de\ bit\ par\ mot} \times I_{D0} + \frac{nombre\ de\ '1'\ par\ mot}{nombre\ de\ bit\ par\ mot} \times I_{D1}$$

Dans les cas étudiés chez Atmel, la plupart des ROMs sont utilisées pour stocker du code machine pour des micro-contrôleurs ; ainsi, nous avons statistiquement un nombre de '0' et de '1' égaux dans le plan mémoire. L'équation devient alors :

$$I_{total} = I_C + I_R + 0.5 \times (I_{D0} + I_{D1})$$

4.2.6 Comparaison de différents outils sur une mémoire (ROM)

Afin de comparer les différentes méthodes de simulation de délai au niveau d'un circuit, on simule une ROM de 32Kbits en technologie $0.35\mu m$, organisée en 4K mots de 8 bits, selon quatre méthodes :

- Simulation Hspice sur un layout extrait, ce qui constituera la référence (Tableau 4.3).
- Simulation Hspice sur un chemin critique utilisant la méthode des générateur de courant décrite précédemment (Tableau 4.4).
- Simulation avec l'outil TimeMill/Powermill [Pow] sur un layout extrait (Tableau 4.5).

On appelle le temps d'accès, tacc, le temps de "setup", tads et le temps de "hold", tadh.

	Pire cas	Cas typique	Meilleur cas
tacc (ns)	9.328	5.448	3.415
tads (ns)	2.267	1.472	0.992
tadh (ns)	0.750	0.503	0.346
Énergie (uW/MHz)	81.592	70.402	54.222

TAB. 4.3 – Hspice sur layout extrait (Référence)

Face aux outils classiques d'évaluation (TimeMill/PowerMill), notre méthode donne de meilleurs résultats aussi bien en terme de délai que de consommation. Les écarts qui apparaissent sont plus grands et plus irréguliers que dans le simple exemple de la section 4.1.5 : un grand nombre de capacités parasites est à prendre en compte, aussi il est difficile de retro-annoter le chemin critique de manière exacte.

	Pire cas	Écart (%)	Cas typique	Écart (%)	Meilleur cas	Écart (%)
tacc (ns)	9.630	+3.2	5.559	+2.0	3.602	+5.4
tads (ns)	0.983	-1.0	1.485	+0.9	1.004	+1.2
tadh (ns)	0.346	0	0.498	-1.0	0.741	-1.2
Énergie (uW/MHz)	80.354	-1.5	68.288	-3.0	55.578	+2.5

TAB. 4.4 – Hspice sur le chemin critique

	Pire cas	Écart (%)	Cas typique	Écart (%)	Meilleur cas	Écart (%)
tacc (ns)	8.916	-4.4	5.382	-1.2	3.497	+2.4
tads (ns)	2.138	-5.7	1.382	-6.1	1.21	+22.0
tadh (ns)	0.711	-5.2	0.553	+10.0	0.367	+6.1
Énergie (uW/MHz)	86.324	+5.8	75.119	+6.7	61.381	+13.2

TAB. 4.5 – TimeMill/PowerMill sur un layout extrait

car le nombre de capacités qui entrent en jeu est grand. Ainsi, il y a une plus grande incertitude sur la valeur des capacités parasites qui sont extraites en les considérant toutes comme ramenées à la masse de manière à simplifier le schéma.

L'avantage de TimeMill/PowerMill, est de pouvoir simuler dans des temps acceptable, une netlist de taille importante issue d'un layout extrait. Pour caractériser un générateur, Il n'est pas envisageable d'utiliser ces outils pour caractériser un générateur, puisque cela signifierait de générer puis d'extraire, les layouts de toutes les configurations possibles. De plus, pendant la phase de développement, on ne dispose pas de tout le layout. Ainsi, notre méthode est adaptée à la caractérisation de générateurs puisqu'il suffit de changer des paramètres dans la netlist pour changer de configuration. En outre, c'est une méthode appréciable pendant la phase de développement, puisqu'elle permet sans avoir de layout final, de pouvoir faire des choix dans l'architecture du circuit.

4.3 Caractérisation en délai de SRAMs en utilisant une méthode basée sur des générateurs de courant

En utilisant la méthode des générateurs de courant contrôlés, développée pour le générateur de ROMs, nous avons créé, puis caractérisé un chemin critique pour 2 SRAMs de capacité 16Kx16 et 24Kx16, en technologie $0.25\mu m$. Le schéma de la figure 4.15 représente

la modélisation du chemin de données. Plusieurs lignes de bit sont raccordées à l'amplificateur de lecture, après le multiplexeur, ce qui est représenté par les transistors supplémentaires dans le bas du schéma.

Cette fois, nous avons modélisé aussi la partie de contrôle contrairement au cas de la ROM. A cause des limitations matérielles, il n'a pas été possible de simuler une mémoire complètement extraite avec Hspice, aussi, nous avons simulé le chemin critique avec Hspice et le layout complètement extrait avec TimeMill. Les résultats sont présentés dans les tableaux 4.6 et 4.7.

	Meilleur cas (ns)		Cas typique (ns)		Pire Cas (ns)	
	Crtp	TM	Crtp	TM	Crtp	TM
temps d'accès	2.26	2.05	3.18	2.95	4.91	4.82
temps de "setup"	1.03	1.07	1.40	1.44	2.10	2.23
temps de "hold"	0.46	0.39	0.58	0.52	0.77	0.75

TAB. 4.6 – Délais mesurés entre le chemin critique (Crtp) et TimeMill (TM) pour une SRAM 16Kx16

	Meilleur cas (ns)		Cas typique (ns)		Pire Cas (ns)	
	Crtp	TM	Crtp	TM	Crtp	TM
temps d'accès	2.42	2.28	3.41	3.26	5.26	5.24
temps de "setup"	1.09	1.16	1.51	1.49	2.34	2.32
temps de "hold"	0.46	0.39	0.58	0.52	0.77	0.75

TAB. 4.7 – Délais mesurés entre le chemin critique (Crtp) et TimeMill (TM) pour une SRAM 24Kx16

L'écart entre les 2 méthodes est peu important et les temps relevés sur le chemin critique sont toujours pessimistes par rapport à la simulation TimeMill sur layout extrait.

4.4 Conclusion

Nous avons présenté une méthode de modélisation des capacités dynamiques (capacités de grille et de drain) qui s'adapte très bien à la caractérisation de générateur puisqu'elle permet de faire varier certaines des caractéristiques du circuit en jouant sur des paramètres numériques de la netlist. Grâce à cette méthode, on peut évaluer de manière précise, à la fois les choix architecturaux et les dimensionnement comme par exemple : le nombre optimal de blocs dans une architecture de mémoire ou encore le nombre maximal de lignes pour que l'amplificateur de lecture d'une SRAM réagisse dans un temps donné. La principale difficulté de cette méthode est de déterminer avec précaution les facteurs multiplicatifs des générateurs de courant. Il faut aussi extraire

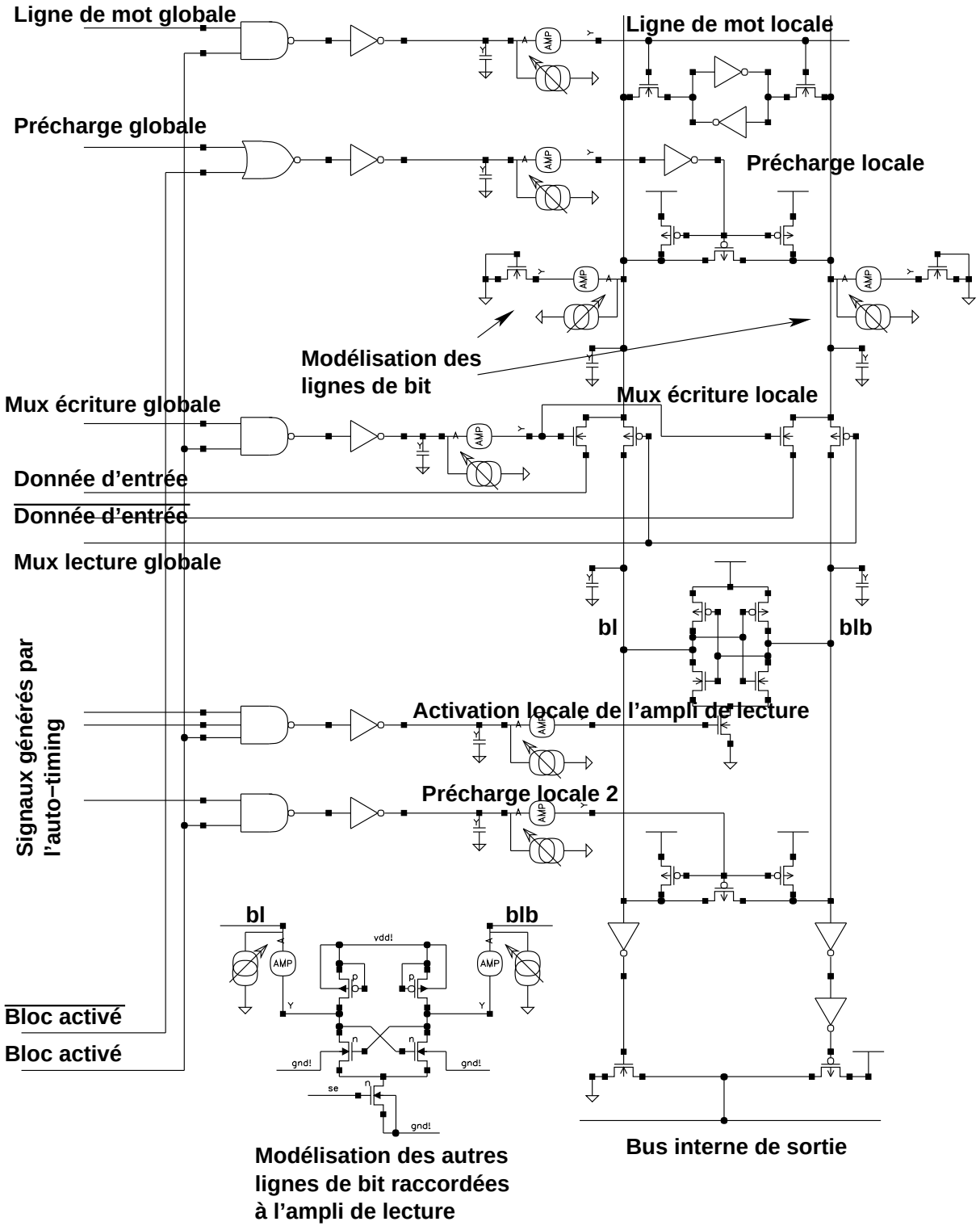


FIG. 4.15 – Modélisation du chemin de donnée de la SRAM en utilisant les générateurs de courant

correctement les valeurs des capacités de routage. Afin d'améliorer notre méthode, nous envisageons de rajouter les effets de couplage notamment entre les lignes de bit et les lignes de mot dans les plans mémoire des ROMs et des SRAMs.

Conclusion

Dans cette thèse nous avons présenté des architecture de mémoires destinées à des applications basse consommation, en utilisant plusieurs techniques indépendantes de la technologie : la précharge sélective et la hiérarchisation de tous les signaux et pas seulement ceux des lignes de mot, l'utilisation d'une ligne de bit factice pour synchroniser les signaux internes pendant le temps nécessaire afin de limiter la consommation et une gestion non classique du protocole d'accès à la ROM avec l'utilisation d'un pipe-line. Cette architecture nous a permis de réaliser dans deux types de technologies $0.5\mu m$ et $0.35\mu m$, des ROMs d'une taille pouvant aller de 64Kb jusqu'à 4Mb.

Bien souvent, les impératifs de basse consommation sont opposés à ceux de la vitesse. A travers le travail réalisé sur une SRAM $0.25\mu m$, nous avons montré qu'il était possible, dans certains cas, de diminuer la consommation jusqu'à 25% sans toucher aux délais en utilisant : la hiérarchisation des signaux, le conditionnement et la génération locale des signaux de commande et surtout en abaissant la tension de précharge à $V_{dd} - V_t$, ce qui constitue une valeur optimale compte-tenu de l'architecture des points-mémoire.

Avec l'évolution des technologies fortement submicroniques comme le $0,25\mu m$ et le $0,18\mu m$, les courants de fuite dans les mémoires deviennent importants. De manière à les caractériser précisément et à en déterminer les différentes composantes, nous avons dessiné des structures de test comprenant plusieurs plans mémoire correspondant aux différents types de mémoires (ROM, SRAM à simple et double port). Des techniques autorisant plusieurs tensions de polarisation du substrat ont pu être étudiées. Nous avons montré qu'il était possible avec la précharge sélective de diminuer considérablement le courant sous le seuil dans les ROMs, bien plus qu'avec les techniques de polarisation de substrat. Les mesures ont été réalisées dans une technologie $0,35\mu m$, puis le seront en $0,25\mu m$. Les technologies multi-caissons pourront être utilisées à partir du $0,25\mu m$, puisque les tests montrent une diminution des fuites en relevant le niveau de V_T par polarisation du substrat pour les SRAMs à simple et double port.

Durant le développement des instances utilisant les architectures précédemment citées, nous avons été confronté à un problème de modélisation : comment représenter de manière précise les capacités dynamiques des transistors sans alourdir les netlists ? Alors que la simulation de layout complètement extrait n'est pas possible compte-tenu de la taille des mémoires, nous avons exploité la redondance matérielle dans les mémoires, exprimée sous forme de blocs, de manière à mettre en place une caractérisation utilisant des générateurs de courant pour représenter la charge créée par

les blocs non présents. Les chemins critiques développés permettent ainsi de prendre en compte les différentes caractéristiques d'une mémoire telles que le nombre de lignes, de colonnes, de blocs ou encore de bits par mots à travers l'utilisation de générateurs de courant dont le gain est paramétrable sans qu'il soit nécessaire de modifier la netlist.

Enfin, en développant un générateur de ROMs, nous avons permis aux concepteurs de circuits de disposer de manière très rapide et très fiable, de toutes les vues nécessaires à l'intégration de ROMs pouvant atteindre jusqu'à 4Mb, ce qui constitue l'une des plus grosse tailles obtenue par un générateur.

Perspectives

De manière à prolonger notre travail sur les Roms nous envisageons de porter l'architecture sur des technologies nouvelles comme le $0.18\mu m$ par exemple. A cette occasion, nous vérifierons si le concept de hiérarchie des signaux est toujours valide compte-tenu du rapport capacité de routage sur capacités de drain et de grille qui tend à augmenter.

De même, pour l'architecture présentée pour une Sram $0.25\mu m$, nous allons étudier son comportement pour les technologies $0.18\mu m$, où les problèmes de bruit sur les amplificateurs de lecture pourraient nous amener à en changer ses caractéristiques. Nous souhaitons, aussi pour de plus grandes tailles de mémoires, évaluer l'utilité de l'amplification hiérarchique désormais accessible avec l'utilisation de technologies 4 voire 5 niveaux de métal.

Enfin, la diminution des courants de fuites, constituera certainement le plus grand défi à relever avec l'apparition de technologies avec des tensions de seuil encore plus faibles que celles utilisées aujourd'hui. Des solutions basées à la fois sur des améliorations technologiques et architecturales devront alors être étudiées et implémentées.

ANNEXE A

Méthodologie de conception de générateurs de mémoires

A.1 Introduction

LE temps de mise sur le marché d'un circuit (Time to market), étant de plus en plus réduit, il est fondamental pour les concepteurs de circuits, de pouvoir disposer le plus tôt possible, de toutes les vues nécessaires pour développer leur applications. Le temps de développement d'une mémoire étant long, l'utilisation d'un générateur s'impose, puisqu'elle permet en quelques minutes de disposer de toutes les vues : modèle de simulation, symbôle, fichiers pour le placement routage, layout. Nous allons décrire dans ce chapitre, une méthode pour la conception d'un générateur de mémoire, en nous appuyant sur le générateur de ROMs que nous avons développé en nous basant sur l'architecture présentée au chapitre 2.

A.2 Développement d'un générateur

A.2.1 Architecture du générateur

Un générateur est articulé autour d'une bibliothèque de cellules feuilles et d'un logiciel d'assemblage de ces cellules. Les cellules feuilles sont les vues layout élémentaires qui, une fois assemblées de manière contigüe, créeront le layout final de l'instance. On trouve aussi un générateur de modèles Verilog/VHDL pour intégrer la mémoire générée à la simulation d'un circuit complet. La méthode de conception utilisée est de type "full-custom" dans la mesure où les structures sont répétitives. La partie contrôle

n'est pas synthétisée non plus afin de garder une maîtrise totale des temps de propagation des signaux et de la consommation.

A.2.2 Modèle comportemental et schémas

Une fois que l'architecture des mémoires est définie, on commence par écrire une description comportementale de la mémoire dans un langage de haut niveau (VHDL/Verilog), ainsi que des stimulis pour valider le modèle. Ensuite, on transcrit le modèle comportemental sous la forme de schémas en portes et transistors. Les schémas sont validés par les même stimulis que ceux utilisés pour la validation du modèle. Selon l'architecture et la complexité de la mémoire, plusieurs schémas sont réalisés. On peut ensuite passer à l'implémentation physique et aux simulations électriques.

A.2.3 Réalisation physique et simulations électriques

Le plan mémoire

Pour développer un générateur, on crée les cellules du plan mémoire en premier. Ce sont ces cellules qui vont déterminer la hauteur et la largeur de toutes les autres cellules de la mémoire qui viendront s'abouter avec elles. Comme les cellules du plan mémoires sont instanciées un très grand nombre de fois, leur contribution aux capacités parasites des lignes de mot et de bit est importante. Aussi, un soin particulier est apporté au développement de ces cellules. En plus dans le cas de cellules pour des SRAMs, il faut veiller au respect des marges de stabilité.

Chemin critique

Le chemin critique est dessiné à partir des schémas décrivant l'architecture. Il est créé autour de la cellule du point mémoire, c'est-à-dire que l'on commence par la simulation des signaux sur les lignes de mot et de bit. Ainsi, les premières capacités parasites ajoutées sont celles des lignes de mot et des lignes de bit. Le chemin critique est enrichi par les valeurs des capacités parasites au fur et à mesure que l'on dessine le layout. Le dimensionnement des transistors est réalisé sur le chemin critique de façon à atteindre les objectifs en délai et en consommation. Les tailles des transistors sont ensuites reportées sur les schémas décrivant l'architecture au niveau portes.

Développement et validation des autres cellules feuilles

Comme il s'agit d'un générateur, il arrive qu'un schéma corresponde à des layouts différents pour des questions d'aboutement notamment. Les layouts sont créés, suivis des vérifications habituelles : vérification des règles de dessin (DRC), comparaison du dessin des masques de fabrication avec les schémas logiques (LVS). Une dernière simulation sur des blocs complètement extraits permet de rendre compte des effets des capacités parasites sur le fonctionnement de la mémoire et de réitérer le flot de conception en cas de besoin.

Le flot de développement et de validation utilisé est présenté à la figure A.1.

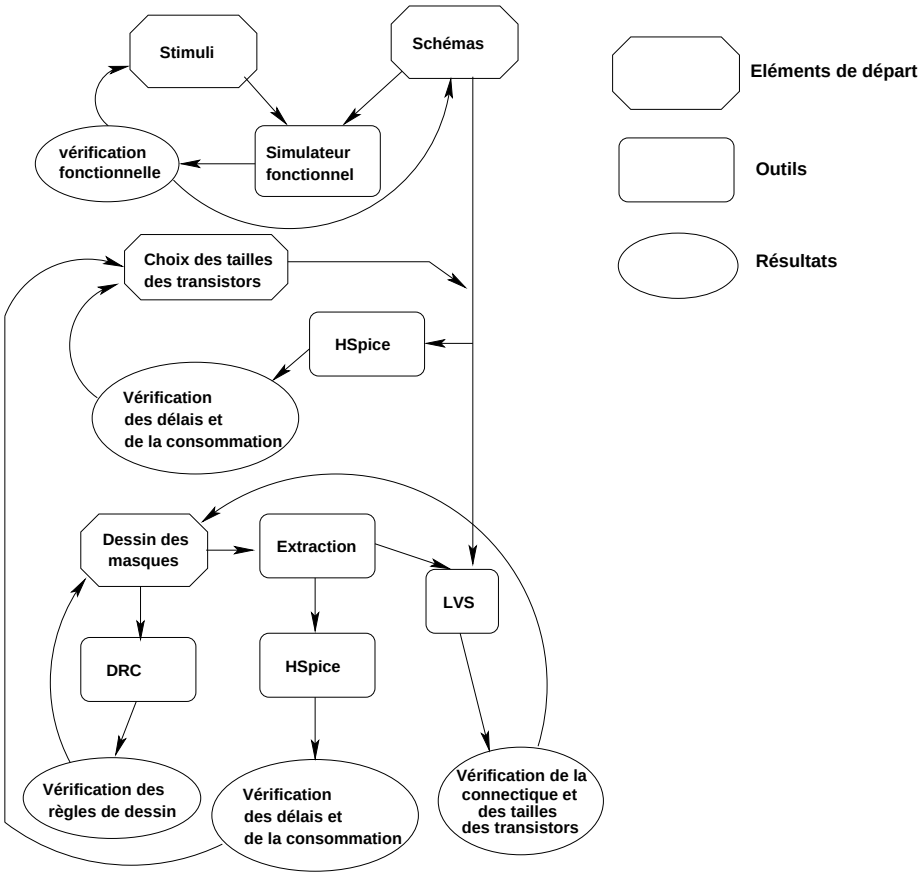


FIG. A.1 – Flot de conception

Comme il est difficile de générer toutes les configurations possibles pour valider exhaustivement un générateur, un jeu représentatif des configurations critiques est défini. Ces configurations (quelques dizaines), permettent de vérifier le bon fonctionnement et la validité du générateur.

A.2.4 Caractérisation des différentes configurations du générateur

Les temps de propagation et les contraintes (temps de “setup” et de “hold”) indiqués par le générateur sont pré-calculés c’est-à-dire qu’ils sont issus d’une caractérisation effectuée sur des schémas critiques utilisés uniquement pendant la phase de développement puis recoupés avec une caractérisation faite sur silicium. Les chemins critiques sont paramétrables et peuvent ainsi reproduire la flexibilité du générateur. Pour la création des chemins critiques, nous avons utilisé un modèle basé sur des générateurs de courant contrôlés par des ampèremètres, pour représenter les capacités dynamiques des transistors (Chapitre 4). Un circuit de test contenant un jeu réduit des configurations critiques est réalisé afin de vérifier la fonctionnalité et de mesurer les performances des mémoires.

A.3 Interface utilisateur

Une interface écrite en C permet à l’utilisateur de rentrer les caractéristiques de la mémoire à générer. Les ROMs compilées sont dédiées à des applications embarquées avec une capacité totale qui varie de 64Kb à 4Mb avec des mots de 8, 16 ou 32 bits de largeur. Ce qui correspond aux demandes les plus courantes, dans la mesure où les ROMs servent souvent à stocker du code pour des micro-contrôleurs dont la largeur en bits des mots est fixée. L’utilisateur a le choix, pour une capacité mémoire donnée, entre plusieurs facteurs de forme en partant de quatre modèles possibles pour l’organisation de la mémoire. En agissant sur le facteur de forme, on modifie le temps d’accès, la consommation et la surface de manière à atteindre les objectifs fixés. La figure A.2 montre les différents modèles d’organisation des blocs pour construire une mémoire. La structure de base est le modèle 1 qui, en étant symétrisé plusieurs fois, donne les autres modèles possibles. La capacité d’une mémoire augmente avec le numéro du modèle : la plus petite des capacités (64 Kb) est générée à partir du modèle 1 et la plus grande (4 Mb) à partir du modèle 4.

Une vue symbolique est fournie ainsi qu’un modèle fonctionnel intégrant les temps de propagation et les contraintes afin d’être intégrés à la méthodologie de conception du système. Une vue dite ”abstract” est générée et est utilisée pour commencer le placement et le routage du circuit avant même que le dessin de masques ne soit encore connu ce qui permet de retarder jusqu’au dernier moment la programmation de la mémoire. Les différentes vues que peut compiler le générateur, pour une configuration donnée, sont indiquées sur la figure A.3.

La construction finale du dessin des masques des mémoires est réalisée par le logiciel “Structure Compiler” de Cadence, à partir d’une bibliothèque de cellules feuilles préexistante. Les vues symbole, “abstract” et la vue structurelle au niveau transistor sont créées à l’aide de programmes en langage Skill de la chaîne Cadence.

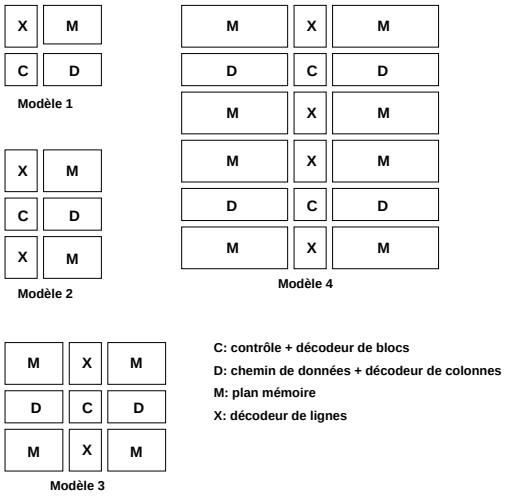


FIG. A.2 – les différents modèles d'organisation

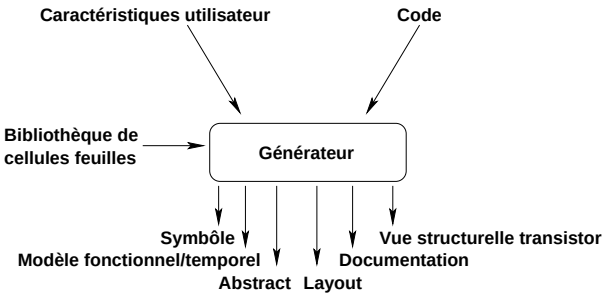


FIG. A.3 – Les différentes vues générées pour une mémoire

A.4 Validation par l'utilisateur

Comme nous l'avons indiqué précédemment, toutes les configurations possibles ne sont pas validées. Aussi, pour éliminer tout risque de défaillance, l'utilisateur a la possibilité de vérifier lui-même chaque mémoire générée en lançant successivement une vérification des règles de dessin (DRC), une extraction du dessin des masques de fabrication, puis une comparaison du résultat de l'extraction avec une "netlist" au niveau transistor créée automatiquement par le générateur. La cohérence des différentes vues est aussi assurée : nom des "pins", leur type (entrée, sortie, alimentation) (Fig. A.4).

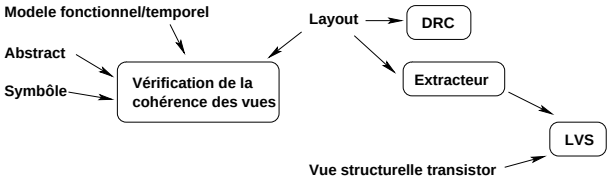


FIG. A.4 – Flot de vérification utilisateur

A.5 Conclusion

Nous avons développé un générateur de ROMs de grande capacité et destinées à des applications embarquées, faible consommation. Les ROMs peuvent atteindre jusqu'à 4Mb, ce qui constitue l'une des plus grosses tailles de ROM issue d'un générateur. Enfin, de manière à garantir une meilleur fiabilité dans notre méthodologie, nous mettons à la disposition des utilisateurs des fichiers qui leur permettent de vérifier par eux-mêmes les différentes vues fournies ainsi que leur cohérence croisée.

Bibliographie

[Aka1996] H. Akamatsu, T. Iwata, H. Yamamoto, T. Hirata, H. Yamauchi, *et al.* A low-power data holding circuit with an intermittent power supply scheme for sub-1V MT-CMOS LSIs. In *Symposium on VLSI Circuits, Digest of Technical Papers*, pp. 14–15, Honolulu, HA, USA, Juin 1996.

[Alo1995] J. Alowersson and P. Andersson. SRAM cells for low-power write in buffer memories. In *IEEE Symposium on Low Power Electronics*, pp. 60–61, , , USA, Octobre 1995.

[Amr1994] B. S. Amrutur and M. Horowitz. Techniques to reduce power in fast wide memories. In *IEEE Symposium on Low Power Electronics*, pp. 92–93, , , USA, Octobre 1994.

[Amr1999] B. S. Amrutur. *Design and analysis of fast low power SRAMs*. PhD thesis, University of Stanford, Stanford, CA, USA, Août 1999.

[Bel1995] A. Bellaouar and M. I. Elmasry. *Low-power digital VLSI design circuits and systems*. Kluwer Academic Publishers, Norwel, MA, USA, 1995.

[Cha1995] A. P. Chandrakasan and R. W. Brodersen. *Low power digital CMOS design*. Kluwer Academic Publishers, Norwel, MA, USA, 1995.

[Cor1998] Avant ! Corporation. *Star-Hspice manual, release 1998.2*. 46871 Bayside Parkway, Fremont, CA, USA, 1998.

[Dav1995] B. Davari, R. H. Dennard, and G. G. Shahidi. CMOS scaling for high performance and low-power - The next ten years. *Proceedings of the IEEE*, 83 :595–606, Avril 1995.

[DeA1997] E. De Angel and E.E. Swartzlander. Survey of low power techniques for ROMs. In *International Symposium on Low Power Electronics and Design*, pp. 7–11, Monterey, CA, USA, Août 1997.

[Dou1997] T. Douseki, S. Shigematsu, J. Yamada, M. Harada, H. Inokawa, *et al.* A 0.5-V MTCMOS/SIMOX logic gate. *IEEE Journal of Solid-State Circuits*, 32 :1604–1609, Octobre 1997.

[Duh1995] M. Duhalde, A. Greiner, and F. Pétrot. A high performance modular embedded ROM architecture. In *IEEE International Symposium on Circuits and Systems*, pp. 1057–1060, Seattle, WA, USA, Mai 1995.

[Fre2000] C. Frey, F. Genevaux, C. Issartel, D. Turgis, and J.-P. Schoellkopf. A low voltage embedded single port SRAM generator in a $0.18\mu m$ standard CMOS process. In *IEEE International Workshop on Memory Technology, Design and Testing*, pp. 106–110, San José, CA, USA, Août 2000.

[Har1997] Y. Haraguchi, T. Wada, and Y. Arita. A hierarchical sensing scheme (HSS) of high-density and low-voltage operation SRAMs. In *Symposium on VLSI Circuits, Digest of Technical Papers*, pp. 79–80, Kyoto, Japan, Juin 1997.

[Has1998] M. Hasegawa, M. Nakamura, S. Narui, S. Ohkuma, Y. Kawase, *et al.* A 256 Mb SDRAM with subthreshold leakage current suppression. In *International Solid-State Circuit Conference*, pp. 80–81, San Francisco, CA, USA, Février 1998.

[Hir1990] T. Hirose, H. Kuriyama, S. Murakami, K. Yuzuriha, T. Mukai, *et al.* A 20-ns 4-Mb CMOS SRAM with hierarchical word decoding architecture. *IEEE Journal of Solid-State Circuits*, 25 :1068–1074, Octobre 1990.

[Hor1993] M. Horiguchi, T. Sakata, and K. Itoh. Switched-source-impedance CMOS circuit for low standby subthreshold current giga-scale LSI's. *IEEE Journal of Solid-State Circuits*, 28 :1131–1135, Novembre 1993.

[IRA2000] IRAM. Intelligent RAM. Technical report, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, http ://iram.cs.berkeley.edu, 2000.

[Ito1995] K. Itoh, K. Sasaki, and Y. Nakagome. Trends in low-power RAM circuit technologies. *Proceedings of the IEEE*, 83 :524–543, Avril 1995.

[Ito1999] K. Itoh, S. Kimura, and T. Sakata. VLSI memory technology : current status and future trends. In *European Solid-State Circuits Conference*, Duisburg, Germany, Septembre 1999.

[ITR1999] ITRS. Overall roadmap technology, characteristics and glossary. Technical report, International Technology Roadmap for Semiconductors, http ://public.itrs.net, 1999.

[Kab1996] H. Kabuo, M. Okamoto, I. Tanaka, H. Yasoshima, S. Marui, *et al.* An 80-MPOS-peak high-speed and low-power-consumption 16-b Digital Signal Processor. *IEEE Journal of Solid-State Circuits*, 31 :494–503, Mars 1996.

[Kaw1993] T. Kawahara, M. Horiguchi, Y. Kawajiri, G. Kitsukawa, and T. Kure et. al. Subthreshold current reduction for decoder-driver by self-reverse biasing. *IEEE Journal of Solid-State Circuits*, 28 :1136–1144, Novembre 1993.

[Kaw1998a] H. Kawaguchi, Y. Itaka, and T. Sakurai. Dynamic leakage cut-off scheme for low-voltage SRAM's. In *Symposium on VLSI Circuits, Digest of Technical Papers*, pp. 140–141, Honolulu, HA, USA, Juin 1998.

[Kaw1998b] H. Kawaguchi, K.-I. Nose, and T. Sakurai. A CMOS scheme for 0.5V supply voltage with pico-ampere standby current. In *International Solid-State Circuit Conference*, pp. 192–193, San Francisco, CA, USA, Février 1998.

[Kur1996] T. Kuroda, T. Fujta, S. Mita, T. Nagamatsu, S. Yoshioka, *et al.* A 0.9-V, 150-MHz, 10-mW, 4 mm^2, 2-D discrete cosine transform core processor with variable threshold-voltage (VT) scheme. *IEEE Journal of Solid-State Circuits*, 31 :1770–1779, Novembre 1996.

[Kus1995] N. Kushiyama, C. Tan, R. Clark, J. Lin, F. Perner, *et al.* an experimental 295 MHz CMOS 4K $\times$ 256 SRAM using bidirectional read/write shared sense amps and self-timed pulsed word-line drivers. *IEEE Journal of Solid-State Circuits*, 30 :1286–1290, Novembre 1995.

[Liu1999] W. Liu, X. Jin, J. Chen, M.-C. Jeng, Z. Liu, *et al. BSIM3v3.2.2 MOSFET model, user's manual*. http ://www-device.eecs.berkeley.edu/˜bsim3, 1999.

[Liu2000] W. Liu, K. M. Cao, X. Jin, and C. Hu. BSIM4.0.0 technical notes. Technical report, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, http ://www-device.eecs.berkeley.edu/˜bsim3/bsim4.html, 2000.

[Mas2000] S. Masuoka, K. Noda, S. Ito, K. Matsui, K. Imai, *et al.* A 0.99-μm^2 loadless four-transistor SRAM cell in 0.13-μm generation CMOS technology. In *Symposium on VLSI Technology, Digest of Technical Papers*, pp. 164–165, Honolulu, HA, USA, Juin 2000.

[Miz1996] H. Mizuno and T. Nagano. Driving source-line cell architecture for sub-1-V high-speed low-power applications. *IEEE Journal of Solid-State Circuits*, 31 :552–557, Avril 1996.

[Miz1999] H. Mizuno, K. Ishibashi, T. Shimura, T. Hattori, S. Narita, *et al.* A 18 μA-standy-current 1.8V 200MHz microprocessor with self substrate-biased data-retention mode. In *International Solid-State Circuit Conference*, pp. 280–281, San Francisco, CA, USA, Février 1999.

[Mon1999] A. H. Montree, A. C. M. C. van Brandenburg, D. B. M. Klaassen, R. Peset Llopis, Y. V. Ponomarev, *et al.* Limitations to adaptative back bias approach for standby power-reduction in deep sub-micron CMOS. In *European Solid-State Device Research Conference*, pp. 580–583, Leuven, Belgium, Septembre 1999.

[Mor1998] T. Mori, B. Amrutur, K. Mai, M. Horowitz, I. Fukushi, *et al.* A 1V 0.9mW at 100MHz 2kx16b SRAM utilizing a half-swing pulsed-decoder and write-bus architecture in 0.25μm dual-Vt CMOS. In *International Solid-State Circuit Conference*, pp. 22.4–1–22.4–2, San Francisco, CA, USA, Février 1998.

[Mut1995] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, *et al.* 1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS. *IEEE Journal of Solid-State Circuits*, 30 :847–854, Août 1995.

[Mut1996] S. Mutoh, S. Shigematsu, Y. Matsuya, H.Fukuda, T.Kaneko, *et al.* A 1-V multithreshold-voltage CMOS digital signal processor for mobile phone application. *IEEE Journal of Solid-State Circuits*, 31 :1795–1802, Novembre 1996.

[Neb1997] W. Nebel and J. Mermet. *Low power design in deep submicron electronics*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[Oot1990] T. Ootani, S. Hayakawa, M. Kakumu, A. Aono, M. Kinugawa, *et al.* A 4-Mb CMOS SRAM with a PMOS Thin-Film-Transistor load cell. *IEEE Journal of Solid-State Circuits*, 25 :1082–1092, Octobre 1990.

[Osa1997] K.-I. Osada, H. Higuchi, K. Ishibashi, N. Hashimoto, and K. Shiozawa. A 2ns access, 285 MHz, two-port cache macro using double global bit-line pairs. In *International Solid-State Circuit Conference*, pp. 402–403, San Francisco, CA, USA, Février 1997.

[Pow] PowerMill. http ://www.epic.com/powermill.html.

[Pri1991] B. Price. *Semiconductors memories, second edition*. John Wiley and Sons, Chichester, UK, 1991.

[Rab1996] J. M. Rabaey. *Digital integrated circuits, a design perspective*. Prentice-Hall, UpperSaddle River, NJ, USA, 1996.

[Sai1996] M. Saito, J. Ogawa, K. Gotoh, S. Kawashima, and H. Tamura. Technique for controlling effective Vth in multi-Gbit DRAM sense amplifier. In *Symposium on VLSI Circuits, Digest of Technical Papers*, pp. 106–107, Honolulu, HA, USA, Juin 1996.

[Sak1994] T. Sakata, K. Itoh, M. Horiguchi, and M. Aoki. Subthreshold-current reduction circuits for multi-gigabit DRAM's. *IEEE Journal of Solid-State Circuits*, 29 :761–769, Juillet 1994.

[Sas1990] K. Sasaki, K. Ishibashi, K. Shimohigashi, T. Yamanaka, N. Moriwaki, *et al.* A 23-ns 4-Mb CMOS SRAM with 0.2-μA standby current. *IEEE Journal of Solid-State Circuits*, 25 :1075–1081, Octobre 1990.

[Shi1995] Y. Shimazaki, K. Ishibashi, K. Norisue, S. Narita, and K. Uchiyama. An Automatic-Power-Save cache memory for low-power RISC processors. In *IEEE Symposium on Low Power Electronics*, pp. 58–59, , , USA, Octobre 1995.

[Shi1998] S. Shigematsu, T. Hatano, Y. Tanabe, and S. Mutoh. Low-power high-speed 1-VLSI using a $0.25-\mu m$ MTCMOS/SIMOX technique. In *IEEE International ASIC Conference*, Rochester, NY,USA, Septembre 1998.

[Soe1999] H. Soeleman and K. Roy. Ultra-low power digital subthreshold logic circuits. In *International Symposium on Low Power Electronics and Design*, pp. 94–96, San Diego, CA, USA, Août 1999.

[Soe2000] H. Soeleman, K. Roy, and B. Paul. Robust ultra-low power subthreshold DTMOS logic. In *International Symposium on Low Power Electronics and Design*, pp. 25–30, Rapallo, Italy, Juillet 2000.

[SSM2000] SSMP. Stanford Smart Memories Project. Technical report, University of Stanford, http ://velox.stanford.edu/smart_memories, 2000.

[Sug1993] T. Sugibayashi, T. Takeshima, I. Naritake, T. Matano, H. Takada, *et al.* A 30ns 256Mb DRAM with multi-divided array structure. In *International*

Solid-State Circuit Conference, pp. 50–51, San Francisco, CA, USA, Février 1993.

[Tak1998] H. Takahashi, S. Muranatsu, and M. Itoigawa. A new contact programming ROM architecture for digital signal processor. In *Symposium on VLSI Circuits, Digest of Technical Papers*, pp. 158–161, Honolulu, HA, USA, Juin 1998.

[Tak2000] K. Takeda, Y. Aimoto, N. Nakamura, H. Toyoshima, T. Iwasaki, *et al.* A 16Mb 400MHz loadless CMOS four-transistor SRAM macro. In *International Solid-State Circuit Conference*, pp. 16.1.1–16.1.2, San Francisco, CA, USA, Février 2000.

[Tra1996] H. Tran. Demonstration of 5T SRAM and 6T dual-port RAM cell arrays. In *Symposium on VLSI Circuits, Digest of Technical Papers*, pp. 68–69, Honolulu, HA, USA, Juin 1996.

[Tsa1998] T. Tsang. A compilable read-only-memory library for ASIC deep submicron applications. In *International Conference on VLSI Design*, pp. 490–494, Chennai, India, Janvier 1998.

[Vee1984] H. J. M. Veendrick. Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. *IEEE Journal of Solid-State Circuits*, 19 :468–473, Août 1984.

[Wes1994] N. H. E. Weste and K. Eshraghian. *Principles of CMOS VLSI design, a systems perpective*. Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 1994.

[Yos1983] M. Yoshimoto, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, *et al.* A divided word-line structure in the static RAM and its applications to a 64k full CMOS RAM. *IEEE Journal of Solid-State Circuits*, 18 :479–485, Octobre 1983.