ÉLECTRONIQUE

Les mémoires à semi-conducteurs

Mémoires : plan

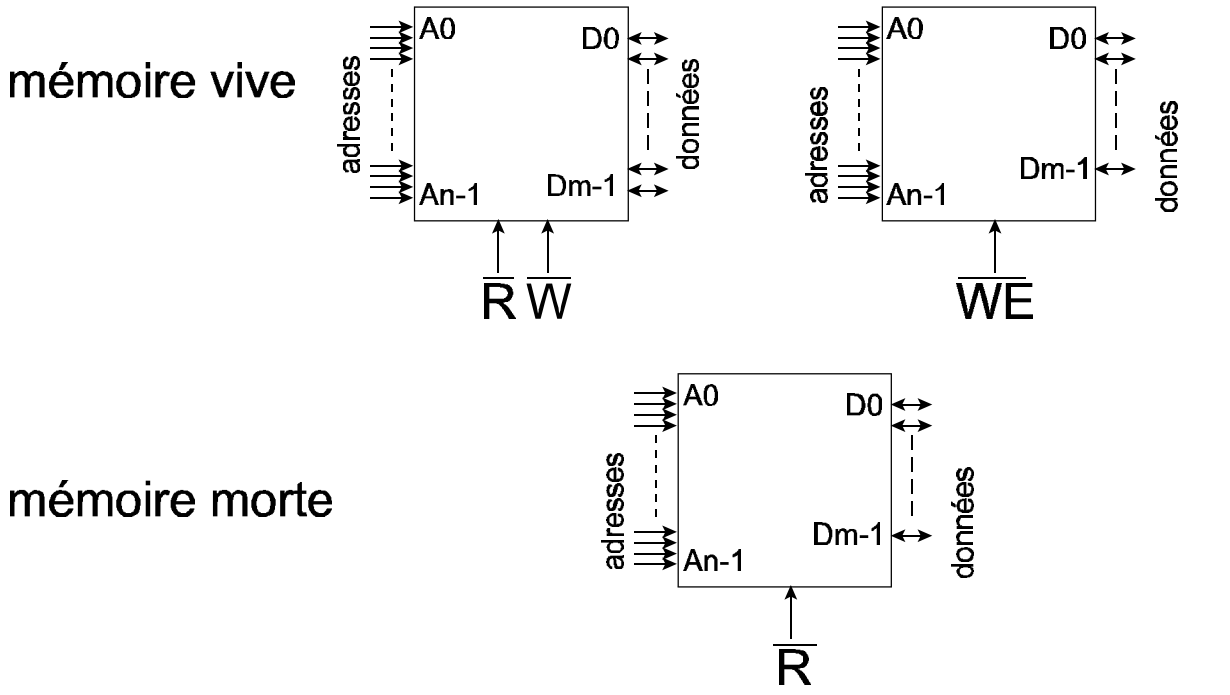
- **Introduction**
  - rappels
  - schéma-bloc
- Mémoires vives (Read-Write)
- Mémoires mortes (ROM)

Bit, Byte, Nibble, Word

- **bit**
  - **B**inary dig**IT** : quantité minimum d'information binaire
- **byte** ou **octet**
  - groupe de 8 bits
  - séparable en 2 **nibbles** ou **quartet** de 4 bits
- **word** ou **mot**
  - groupe de 16 bits ou
  - groupe de bit de la taille du bus de données (préciser la taille : mot de ... bits)

Les mémoires, ou la manière de les grouper, doivent refléter la structure d'information que l'on veut en extraire

Schéma-bloc des mémoires



Les mémoires s'interfacent par rapport au monde extérieur à l'aide de 2 groupes de bits appelés bus :
- le ***bus de données*** (de M bits) permet d'extraire de la mémoire 1 mot de M bits en une seule opération; les valeurs courantes de M sont : 1,4,8,16,32 et 64,...
- le ***bus d'adresse*** (de N bits) permet de sélectionner dans la mémoire 1 mot parmi 2^N

S'y ajoutent plusieurs signaux de contrôle :
- **R** (Read) : indique à la mémoire une lecture c'est-à-dire une extraction d'un mot de la mémoire à l'adresse désignée par le bus d'adresse
- **W** (Write) : indique à la mémoire une écriture c'est-à-dire le chargement d'un mot dans la mémoire à l'adresse désignée par le bus d'adresse
- **OE** (Output Enable) : commande de l'état de haute impédance des "buffers" de sortie vers le bus de données. Cet étage (3-state) est indispensable pour que la mémoire puisse partager le bus de données avec d'autres boîtiers de mémoires ou de périphériques.

REM : les signaux Read et Write étant mutuellement exclusifs, ils sont souvent regroupés en une ligne unique appelée R/W' ou WE'(Write Enable)

On distingue les mémoires :
- ***VIVE ou Read-Write Memory***, où l'on peut lire et écrire; une mémoire de données est nécessairement vive. Une mémoire vive peut aussi servir de mémoire programme (c'est le cas dans les micro-ordinateurs).

- ***MORTE ou ROM (Read-Only Memory)***, où l'accès est limité à la lecture. Il n'y a donc pas de signal de contrôle pour l'écriture. Une ROM ne peut contenir qu'un programme ou des constantes (exemple CD ROM, cassette de jeu).

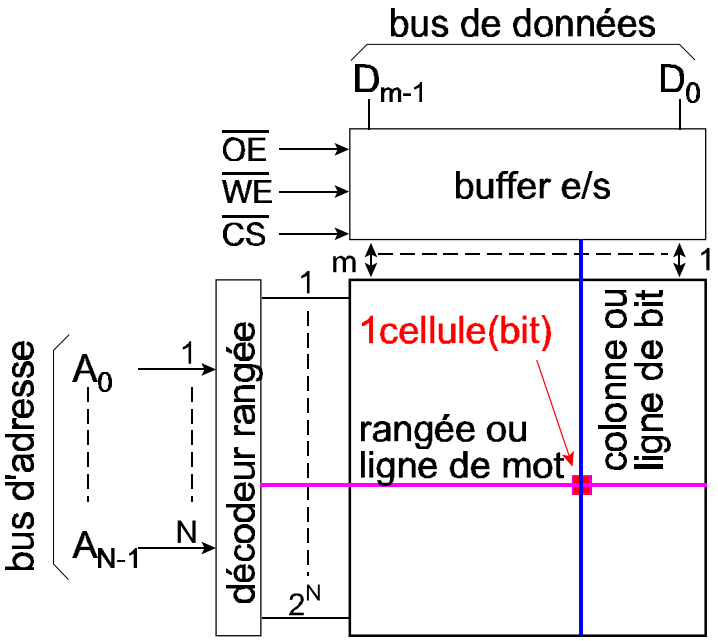
REM : la mémoire vive est le plus souvent désignée par le terme ***RAM (Random Access Memory)***, qualifiant ici la possibilité d'accéder aux mots de la mémoire dans n'importe quel ordre (par opposition aux mémoires à accès séquentiel, comme les bandes magnétiques). RAM est donc un nom impropre puisque, les ROM à semi-conducteurs sont également à accès aléatoire !

Mémoires : plan

- Introduction
- **Mémoires vives (Read-Write)**
  - **structure matricielle**
  - **interface**
  - **cellules statiques**
  - **cellules dynamiques**
- Mémoires mortes (ROM)

Mémoires : organisation matricielle

mémoire organisée en mots

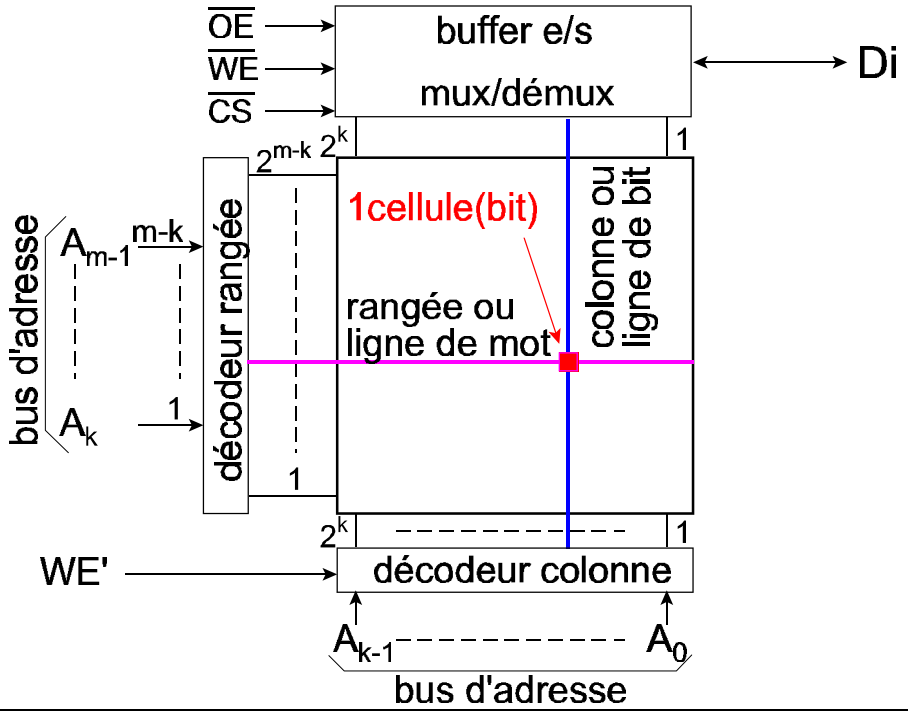


Fondamentalement, l'unité de stockage (ou cellule de mémoire) est le bit. Si cette mémoire doit être connectée à un bus de données de M bits, l'organisation la plus simple est de grouper les cellules en blocs de M unités. Une mémoire contenant 2^N mots de M bits est alors organisée en une matrice de 2^N rangées de M bits.
L'accès à une cellule de mémoire, pour la lire ou la modifier, se fait via une des 2^M lignes de bits, qui sont les colonnes de la matrice.

Une cellule est donc à l'intersection d'une ligne de mot (rangée) et d'une ligne de bit (colonne).

Dans une mémoire organisée en 2^N mots de M bits, les N bits du bus d'adresse sont directement présentés à un décodeur N vers 2^N qui sélectionne la ligne de mot concernée. Toutes les lignes de bits sont activées simultanément.
L'interface vers le bus de données se fait à travers un jeu de "buffers" d'entrée et de sortie dont le sens d'activation dépend de l'état des signaux RD' et WR' ou du signal unique WE' (Write Enable).
Il existe encore deux signaux de contrôle CS (Chip Select) et OE(Output Enable), dont le rôle sera explicité plus tard.

Mémoire vive de 2^M x 1 bit



Une mémoire peut être organisée en 2^M mots de 1 bit et ne comporter qu'un seul fil de données. Cette borne est alors connectée à un des bits Di du bus des données, et l'on place autant de boîtiers de mémoire en parallèle qu'il y a de bits de données au processeur.

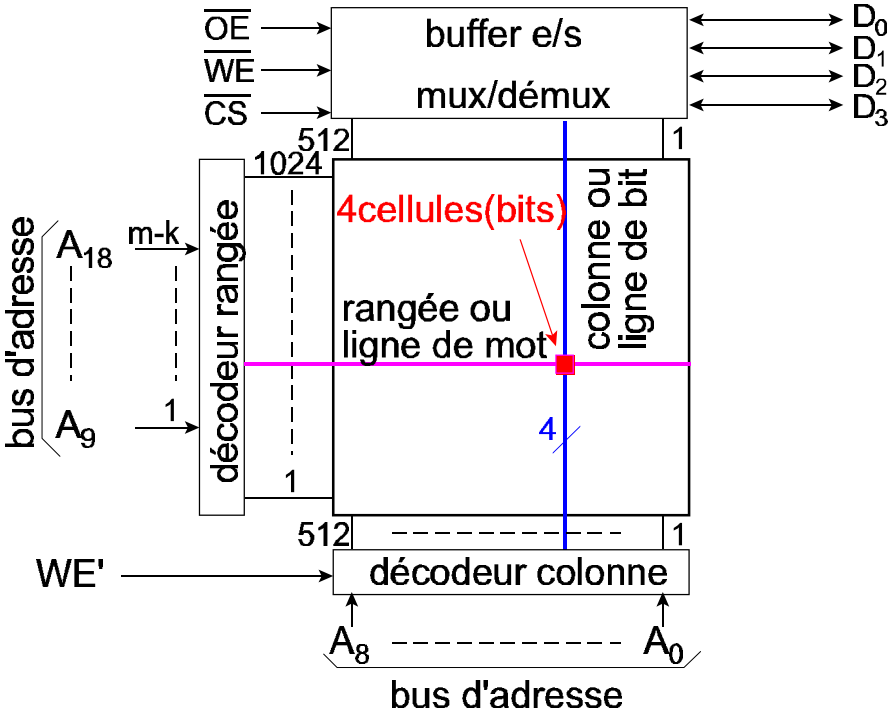
La mémoire conserve une structure matricielle en $2^{(M-k)}$ rangs de mots de 2^k bits. Les M bits d'adresse sont partagés en deux parties
- le décodeur de rangée reçoit les (M-k) bits de poids fort et sélectionne un mot de 2^k bits
- le décodeur de colonne reçoit les k bits de poids faible et sélectionne un bit parmi les 2^k

Le buffer d'entrée-sortie est précédé d'un multiplexeur (en sortie)/démultiplexeur (en entrée) également commandé par la ligne de bit, afin de connecter le bit sélectionné avec la borne unique de données Di.

Exemple numérique :
- mémoire de 128 Kbits soit M=17 bits d'adresse
- organisation en 16K mots de 8 bits soit k=3 et M-k =14
- décodeur de rangée 14 vers 2^{14} soit 16K
- décodeur de colonne 3 vers 8
- (dé)multiplexeur de 8 vers 1

NOTE sur les préfixes de multiplication
k (kilo) => x1000
K (Kilo) => x1024

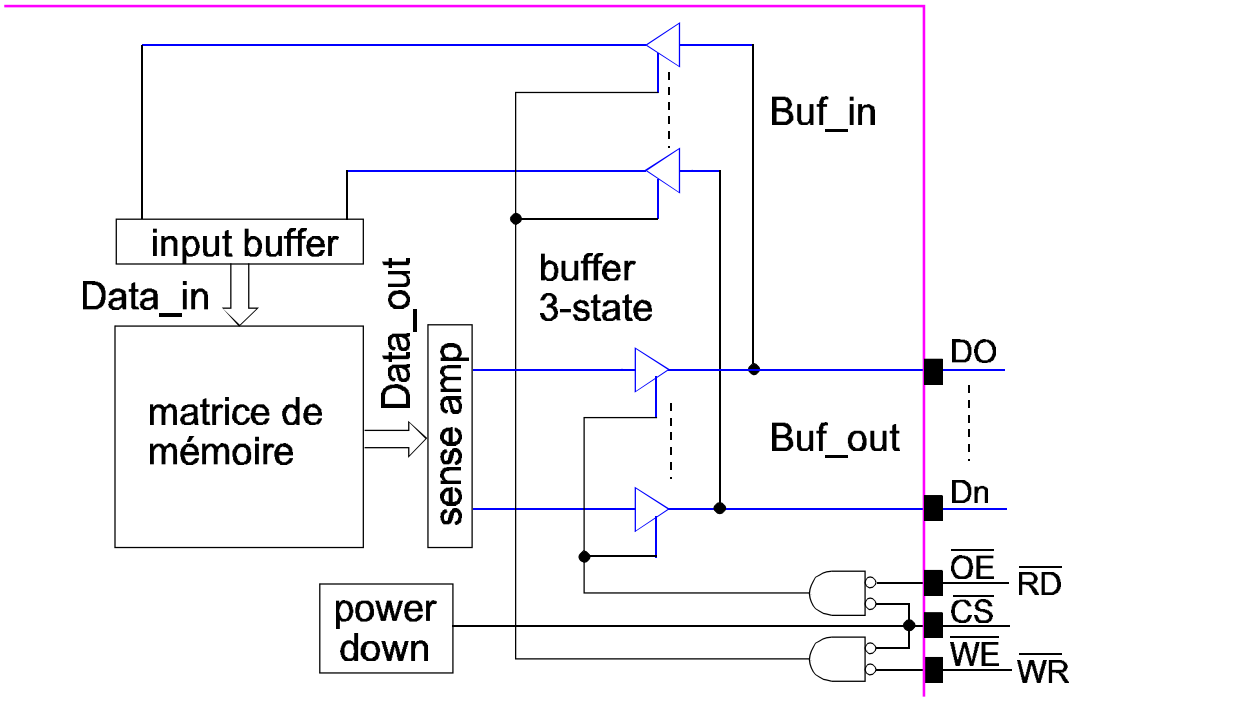
Mémoire vive 1024 x 512 x 4bits



Il existe également des structures matricielles hybrides. La figure montre une RAM 1024 x 512 x 4bits dans laquelle le décodeur de colonne active simultanément 4 lignes de bits qui seront dirigées vers un quadruple buffer d'entrée-sortie.

Pour un processeur à 8 bits, on placera deux mémoires identiques, l'une connectée à D0..D3, l'autre à D4..D7.

Interface mémoire↔bus



Soit une matrice de mémoire organisée par mots de N bits.
En lecture, les bits de données sont extraits par les "sense amplifiers" (littéralement amplificateurs chargés d'aller "sentir" le niveau des bits dans le coeur de la mémoire) et dirigés vers les bornes de sortie via N buffers (Buf_out).
Ces derniers ont une sortance suffisamment élevée pour garantir un temps de transition acceptable sur la charge capacitive du bus.
Ce sont nécessairement des buffers "3-state" puisque le bus de données est partagé avec d'autres boîtiers. Seul un boîtier à la fois peut quitter l'état HiZ et fixer le niveau logique du bus des données.
La ligne de commande qui active les buffers en sortie s'appelle OE' (Output ENABLE) et est reliée au signal de contrôle RD' (ReaD) du processeur.

Un autre jeu de N buffers (Buf_in) est placé en entrée. Lors d'une écriture, ces buffers sont activés dans le sens bus-vers-mémoire par la ligne WE' (Write Enable), reliée au signal de contrôle WR' (WRite) du processeur.

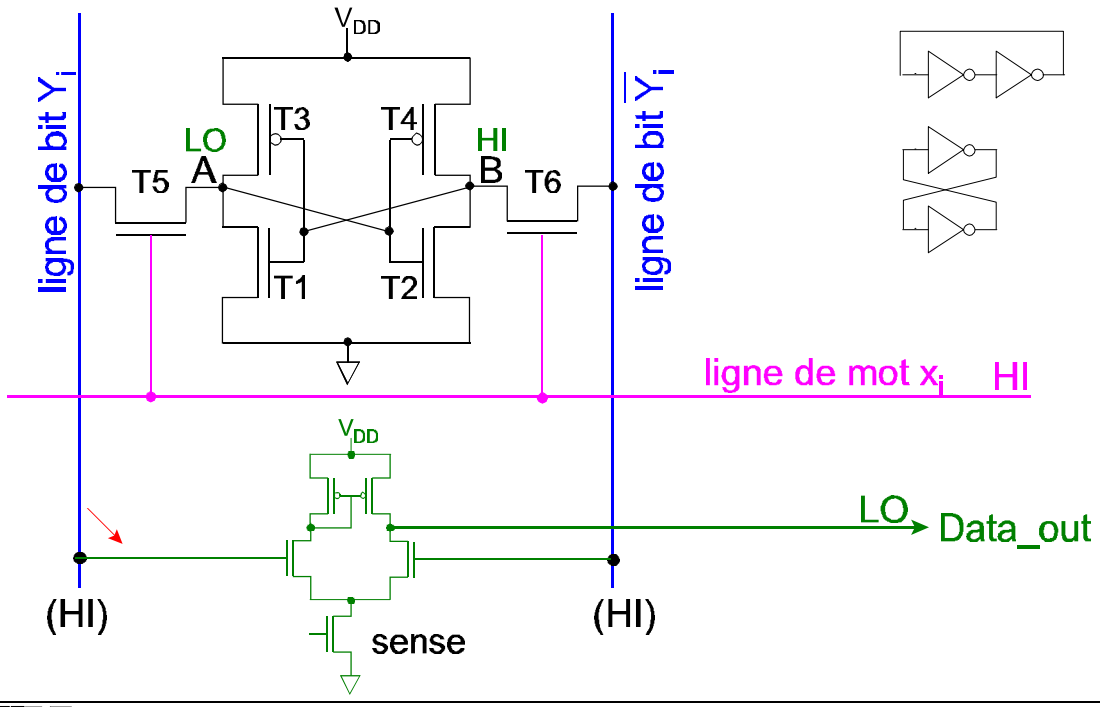
La ligne CS' (Chip Select) est la ligne d'activation globale du boîtier et est reliée à une sortie du décodeur d'adresse. Si CS est inactif
- ni OE' ni WE' ne sont pris en considération
- la mémoire passe en mode de consommation minimale (ceci n'a de sens que pour les mémoires statiques que nous définirons plus loin)

REM1 : le rôle des buffers "Buf_in" et "Buf_out" est aussi de séparer les entrées et les sorties des données au niveau de la matrice, alors qu'elles partagent les mêmes bornes au niveau du boîtier.

REM2: le placement des entrées et des sorties de la matrice de mémoire sur deux côtés adjacents est lié à un allégement du dessin, il n'a rien à voir avec le sens réel des lignes de mots et de bits.

REM3 : il existe des mémoires où les données entrantes et sortantes sont sur des bornes différentes du boîtier.

Cellule statique CMOS : lecture



Une cellule de mémoire classique à transistors est le bistable constitué par deux paires CMOS (inverseurs) qui sont rebouclées.
On vérifie aisément l'existence de deux états stables :
- supposons T1 saturé, alors T3 est coupé par principe de la paire complémentaire
- la sortie de l'inverseur T1-T3 est donc en LO et constitue le signal de commande de l'inverseur T2-T4, dont la sortie est donc en HI
- cette sortie commande à son tour l'entrée de l'inverseur T1-T3 dont la sortie est donc en LO, et la boucle est fermée.
- l'autre état du bistable est symétrique

Supposons que le bistable est dans l'état indiqué sur la figure : T1 et T4 sont saturés, alors que T2 et T3 sont coupés. Convenons que c'est l'état LO du bistable.

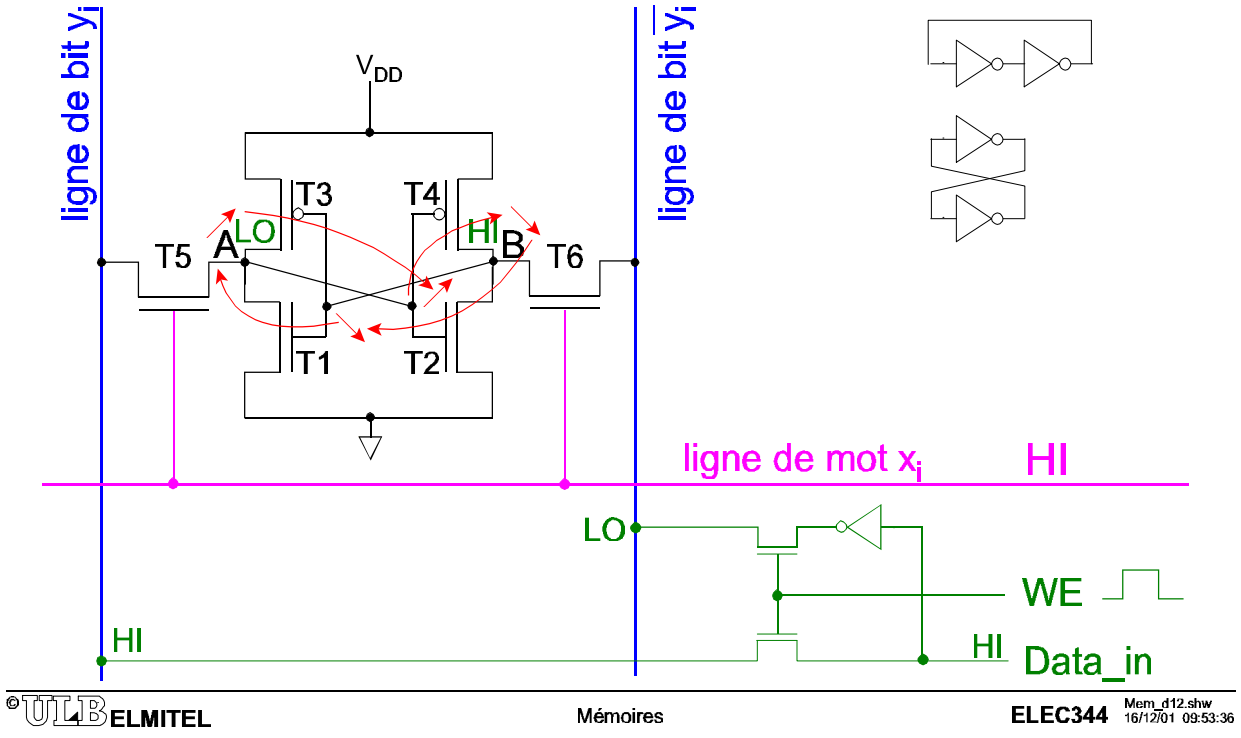
Remarquons que, comme tout CMOS, la cellule consomme très peu pour maintenir l'information puisqu'il y a un transistor coupé dans chaque paire.

Pour lire l'état du bistable :
- on met les deux lignes de bits à HI, (via des transistors non représentés sur la figure) avec une impédance de sortie supérieure à celle du bistable
- on active la ligne de mot, ce qui sature T5 et T6
- la conduction de T6 ne peut que renforcer légèrement l'état HI de la ligne de bit Y'
- la conduction de T5, par contre, fait baisser le potentiel de la ligne de bit Y à car l'impédance de sortie du bistable est plus faible que celle de la source qui alimente Y
- l'amplificateur différentiel de "sense" sature en LO et l'état Data_out de la cellule est donc bien LO

La lecture est non destructive, car
- la tension de drain de T1 est insuffisante pour faire basculer la paire T2- T4
- la ligne de bit Y' tend à maintenir la paire T1-T3 dans son état initial.

REM1 : chaque cellule nécessite 6 transistors au total
REM 2: l'amplificateur différentiel de "sense" nécessite 5 transistors, mais il n'en faut qu'un par colonne

Cellule statique CMOS : écriture



Pour écrire dans la cellule :
- on place la donnée Data_in, issue du buffer d'entrée, sur la ligne de bit Yi et son complément logique sur la ligne de bit Y'i à travers deux transistors pilotés par WE (Write Enable)
- on active la ligne de mot pour rendre T5 et T6 conducteurs

Supposons que la donnée soit HI (c'est-à-dire le complément de l'état du bistable) :

- la tension du point A s'élève, car un courant circule de la ligne de bit Y vers la masse, en passant par T5 et T1; cette tendance est renforcée par l'action de la ligne de bit Y' qui, via T6, fait baisser la tension de grille de T1-T3, donc tend à couper T1 et à allumer T3
- la tension au point B s'abaisse car un courant circule de l'alimentation vers la ligne de bit Y', en passant par T4 et T6; cette tendance est renforcée par l'action de la ligne de bit Y qui, via T5, fait monter la tension de grille de T2-T4, donc tend à couper T4 et à allumer T2

Cette double rétroaction positive fait changer le bistable d'état; les points A et B prennent l'état des lignes de bit Y et Y'.

A partir de ce moment, on peut désactiver WE, donc les lignes de bit, et désactiver la ligne de mot, ce qui coupe T5 et T6.

Le bistable conservera son état tant que l'on ne vient pas réécrire la donnée opposée.

Cellule statique CMOS : bilan

- avantages
  - rapidité d'accès (qq ns)
  - cellule statique
    - lecture non destructrice
    - rétention indéfinie
  - consommation faible
    - extrêmement faible pour garder la donnée
    - petite pointe de courant à la lecture
    - pointe de courant à l'écriture
    - consommation due à la charge capacitive des buffers de sortie
- inconvénients
  - cellule volumineuse (6 transistors), donc chère
  - volatile (contenu perdu si on cesse d'alimenter)

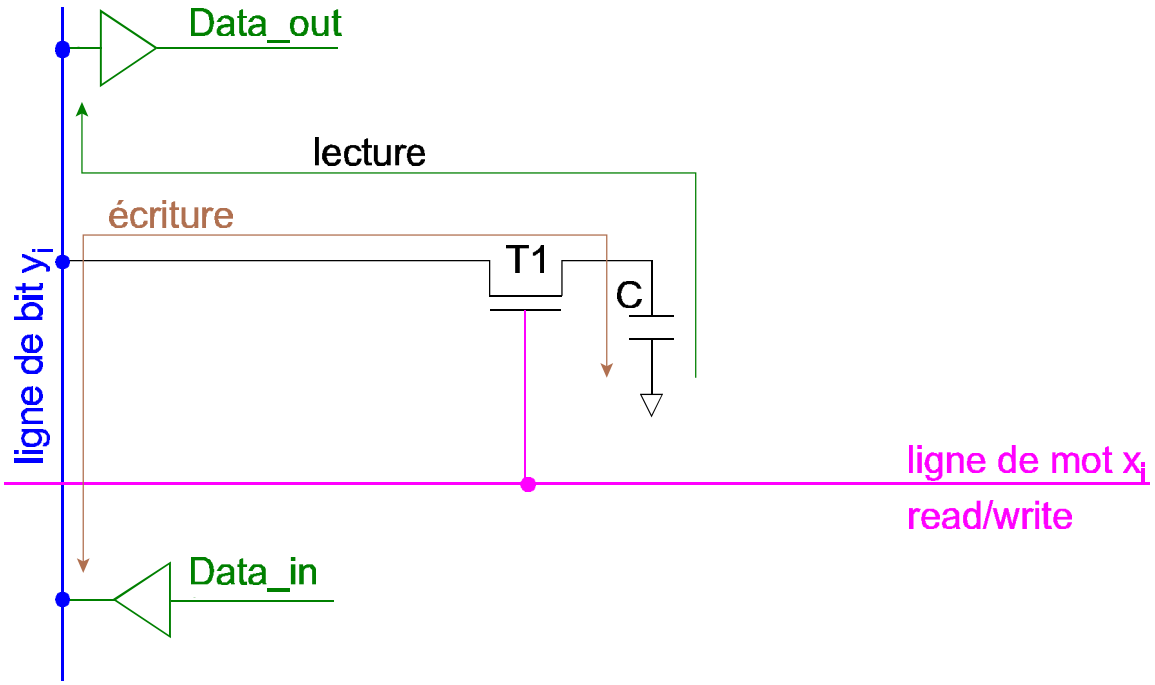
La cellule que nous venons de décrire ne possède pratiquement que des avantages :

- elle est statique, c'est à dire qu'elle ne changera de contenu que si l'on vient explicitement écrire d'autres données et conserve indéfiniment son état tant qu'on l'alimente, quel que soit le nombre de lectures.
- elle consomme très peu.

Comme tous les CMOS, elle ne consomme que pour changer d'état et charger les condensateurs parasites du bus des données en sortie.
La rétention des données, en particulier, est très peu énergivore; cela permet fabriquer des circuits contenant une RAM statique et une pile au lithium capables de mémoriser des données pendant plusieurs années (comme les mémoires de configuration des PC)

Ses seuls inconvénients sont d'être
- volatile : le contenu est perdu si l'on coupe l'alimentation
- volumineuse, car il faut 6 transistors pour mémoriser un bit, et donc coûteuse.

Cellule dynamique MOS



Vu les grandes quantités de mémoire nécessaires dans beaucoup de micro-ordinateurs (1000 fois plus qu'il y a 20 ans), les mémoires statiques sont trop chères.
On a donc conçu divers types de cellules de mémoires plus petites. La plus courante se sert d'un condensateur comme élément de stockage de l'information.

Supposons le condensateur chargé : la cellule est dans l'état HI.

La lecture s'opère par activation de la ligne de mot. Le transistor T1 est alors passant et le condensateur est connecté à la ligne de bit; comme la capacité de celle-ci est 10 fois supérieure, la tension sur le condensateur de stockage est réduite d'un facteur 10. Cette tension est toutefois suffisante pour être détectée par l'amplificateur de "sense" placé sur la ligne de bit.
La lecture est donc destructive et doit être immédiatement suivie d'une réécriture de la valeur qui vient d'être lue.

L'écriture s'opère en présentant la donnée Data_in sur la ligne de bit via un buffer en basse impédance; en activant la ligne de mot, T1 est mis en conduction et (dé)charge le condensateur, selon la valeur du bit de donnée.
Cette cellule est appelée ***dynamique***, parce que, si on n'y accède pas pendant un long intervalle de temps, le (très petit) condensateur se décharge par le courant de fuite de T1 et l'information est perdue. La rétention n'est donc pas indéfinie.
Pour conserver l'information il faut "rafraîchir" la mémoire, c'est-à-dire provoquer une lecture (automatiquement suivie d'une réécriture) de chaque cellule à intervalle régulier (typiquement quelques ms)

Cellule dynamique CMOS : bilan

- avantages
  - compacité
    - 1 transistor et 1 condensateur
    - 1 ligne de bit
    - lecture/écriture via la ligne de mot
  - consommation faible
- inconvénients
  - lecture destructrice
    - nécessité de réécrire => ralentissement
  - rafraîchissement régulier
    - nécessité d'un séquencement (interne ou externe)
    - consomme du temps (qq %)
  - volatile (contenu perdu si on cesse d'alimenter)

Par rapport aux mémoires statiques, les mémoires dynamiques n'ont que la compacité comme avantage, mais c'est un avantage décisif : toutes les mémoires centrales d'ordinateurs sont dynamiques.

Les inconvénients liés au rafraîchissement doivent être acceptés pour obtenir la quantité de mémoire désirée.

Pour de "faibles" quantités (de quelque Ko à quelques centaines de Ko) de mémoire ultra-rapide (mémoires caches par exemple) la mémoire statique reste le meilleur choix.

Mémoires : plan

- Introduction
- Mémoires vives (Read-Write)
- **Mémoires non-volatiles (ROM)**
  - **spécificité des mémoires de programme**
  - **MaskROM dite "ROM"**
  - **UV-EPROM dite "EPROM"**
  - **FLASH-EPROM dite "FLASH"**
  - **EEPROM ou E²PROM**

Le stockage des programmes

- non-volatile : pas de perte du programme si l'on cesse d'alimenter
- µordinateurs banalisés :
  - stockage magnétique ou optique
    - programmes énormes (qq 10 ou 100 Mo)
    - nombreux programmes
  - mémoire programme en RAM (sauf BIOS)
- embedded systems
  - stockage sur silicium ("solid-state")
    - programme unique
    - programme peu volumineux (qq Ko à qq 100 Ko)
    - environnement sévère (vibrations, chocs, T°, poussière)

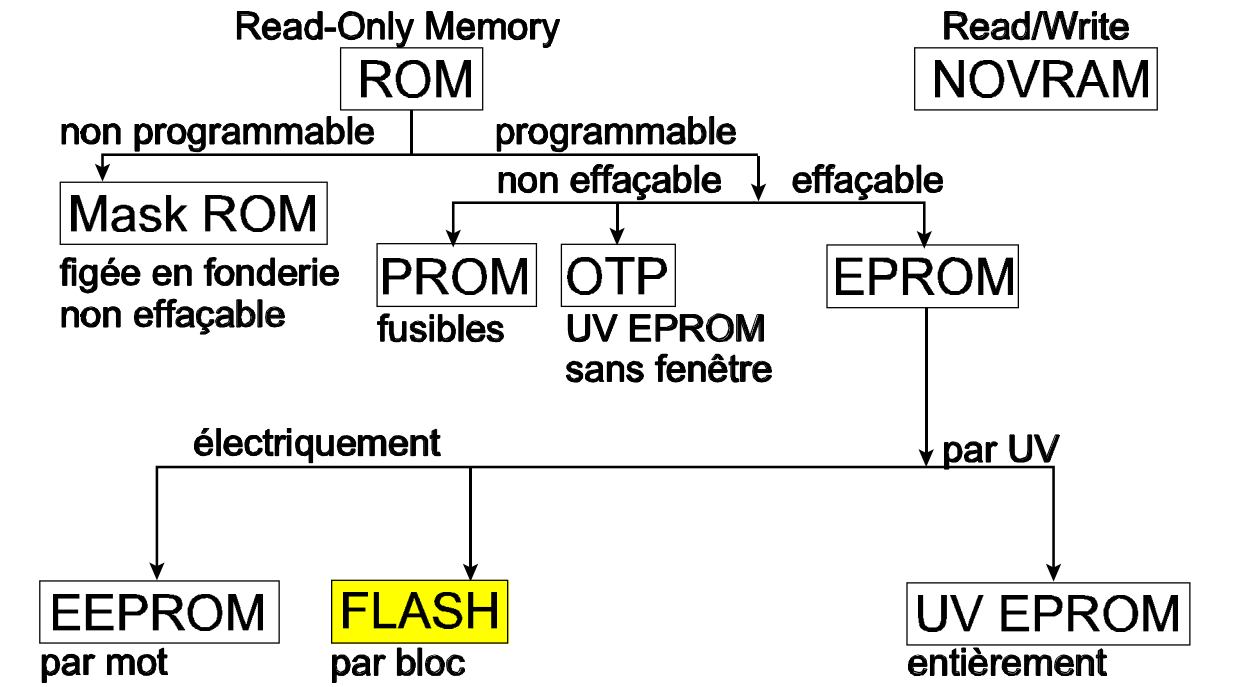
Pour stocker les programmes des microprocesseurs, on a recours à des médias non-volatiles c'est-à-dire ne perdant pas leur contenu lorsque l'on cesse de les alimenter.

Dans les micro-ordinateurs à usage général, les programmes ont une taille très élevée (plusieurs dizaines, voire centaines de Moctets) et l'utilisateur désire avoir plusieurs programmes différents à portée de main. Pour stocker de telles quantités, les seuls médias possédant un coût par bit abordable sont les matériaux magnétiques (bandes magnétiques, disques durs ou souples) ou optiques (CD-ROM, DVD-ROM).

Les programmes en cours d'exécution sont donc situés en mémoire vive (RAM), à l'exception du "boostrap" qui
- est le tout premier segment de programme exécuté au RESET
- est stocké en mémoire à non-volatile à semi-conducteurs (de type EPROM ou FLASH, voir plus loin)
- est chargé de programmer les périphériques donnant accès aux médias magnétiques ou optiques pour la suite des opérations

Dans les "embedded systems", par contre, on a souvent recours à des mémoires programmes à semi-conducteurs car :
- il n'y a qu'un seul programme, toujours le même
- la taille de ce programme est plus réduite (de quelques Koctets à quelques centaines de Koctets)
- l'environnement n'autorise pas toujours la présence d'éléments mécaniques fragiles (variations de température, chocs, vibrations, poussières) et interdit donc les disques optiques ou magnétiques.

Mémoires non-volatiles



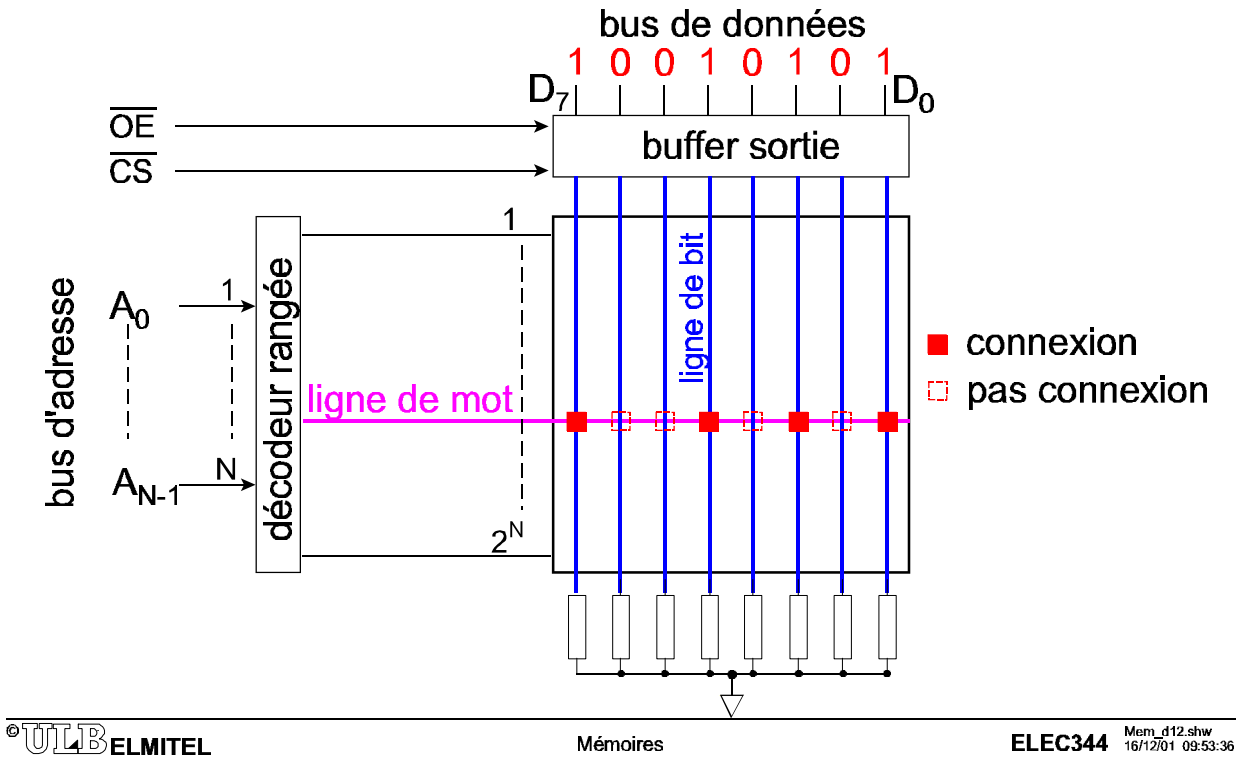
Les mémoires programmes sont pour la plupart des mémoires portent le nom de MEMOIRES MORTES ou ROM (Read-Only Memory), ce qui indique que l'on n'y a accès qu'en lecture pendant l'exécution du programme. Il faut évidemment avoir accès au moins une fois en écriture pour y stocker le programme.

Les ROM se classifient d'après
- la programmabilité par l'utilisateur
- la possibilité d'effacement
- le mode d'effacement

- les "MASK ROM" non-programmables et non effaçables; l'inscription du programme se fait par le dernier masque de fabrication, d'où son nom. C'est une solution n'offrant aucune souplesse, ni droit à l'erreur, mais très peu coûteuse. Les "MASK ROM" sont souvent appelées simplement ROM.
- les mémoires programmables électriquement une seule fois par l'utilisateur et non effaçables :
  - les PROM (Programmable ROM): basées sur une technologie à fusibles, elles ont pratiquement disparu
  - les OTP (One Time Programmable); ce sont en fait des UV EPROM (voir ci-dessous) sans fenêtre d'effacement
- les EPROM ou "Erasable PROM" peuvent être programmées électriquement par l'utilisateur, puis effacées et reprogrammées un grand nombre de fois. Elles se divisent en trois sous-catégories liées au mode d'effacement.
  - les UV EPROM où toute l'information est effacée en une fois par exposition aux ultra-violets. Elles sont souvent appelées simplement EPROM.
  - les EEPROM ou E²PROM (Electrically Erasable PROM) s'effacent électriquement et peuvent être effacées mot par mot.
  - les FLASH qui sont des EEPROM nettement moins chères, grâce à une densité beaucoup plus élevée; le compromis est la nécessité d'effacer puis de réécrire complètement un bloc de mémoire dès que l'on doit changer un seul mot de ce bloc.

Rajoutons enfin une catégorie à part pour les NOVRAM ou NOnVolatile RAM, qui sont des mémoires vives munies d'une pile.

Structure des mémoires mortes



La structure des mémoires mortes est analogue à celle des RAM organisées par mots.
Le décodeur d'adresse active une ligne de mot et toutes les lignes de bits sont présentées simultanément sur les bornes de sortie de données, via un jeu de buffers "3-state".

La figure présente ici une mémoire morte de 2^N mots de 8 bits.

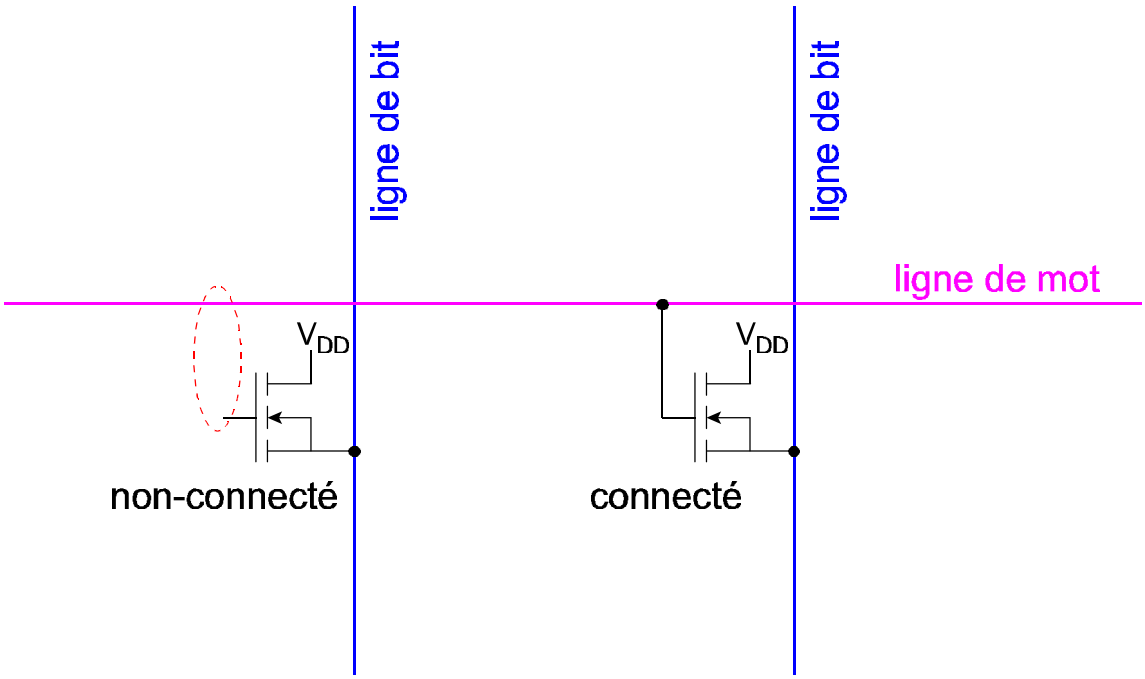
La différence par rapport aux mémoires vives est la présence de résistances qui fixent par défaut le niveau des lignes de bits. Dans cet exemple nous illustrons la mise à L0 par défaut via des résistances "pull-down".

La cellule de mémoire élémentaire sera alors un élément programmable (au moins une fois) qui va réaliser ou non une connexion non-volatile entre la ligne de mot et la ligne de bit qui la traversent.

Lorsque l'on active la ligne de mot à l'état HI :
- les lignes de bits connectées à la ligne de mot passent en HI
- les lignes de bit non-connectées à la ligne de mot restent au niveau LO fixé par les résistances de "pull-down".

Les différentes versions de mémoires mortes correspondent à différents types de cellule d'interconnexion ligne de bit/ligne de mot, que nous allons passer en revue.

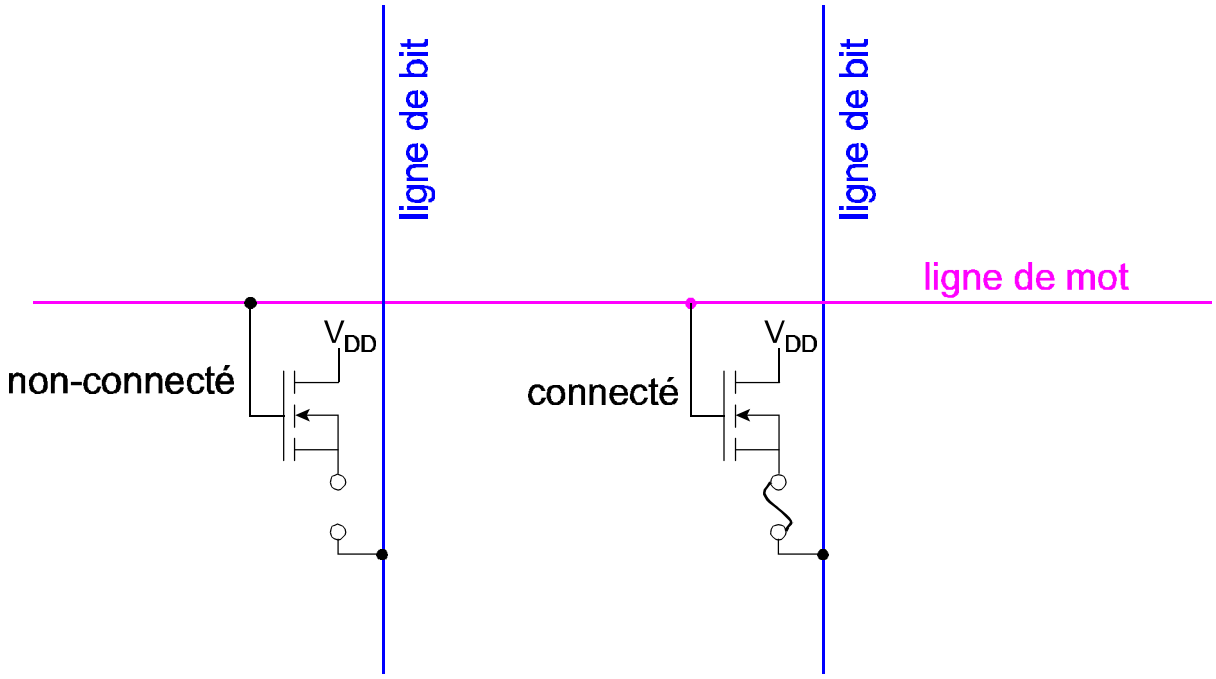
Cellule de Mask ROM



Dans les "MaskROM", plus communément appelées ROM, la connexion est réalisée par un transistor dont la grille est pilotée par la ligne de mot.

L'absence de connexion consiste à ne pas créer le transistor, ou à ne pas connecter sa grille à la ligne de mot. Cela se réalise en personnalisant le dernier masque de fabrication, d'où le nom de "MaskROM".

Cellule de PROM à fusible



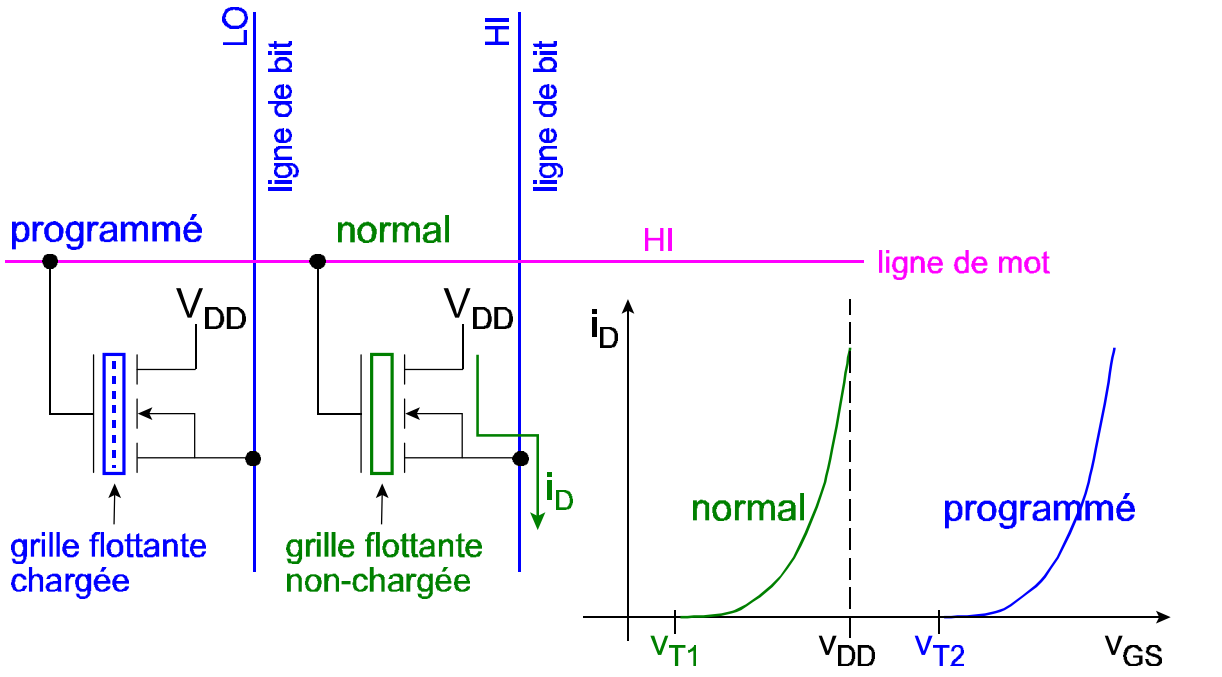
La PROM (Programmable ROM) peut être programmée une fois par l'utilisateur. Tous les transistors préexistent et leur drain est connecté à la ligne de bit via une zone fusible. Par défaut, tous les bits sont donc programmés à 1.

Pour programmer la mémoire, on sélectionne la ligne de mot et on applique une impulsion de tension négative sur les lignes de bit où l'on veut inscrire un 0. Le courant qui en résulte fait fondre le fusible et coupe la connexion.

REM1 : le mécanisme inverse existe : faire claquer une diode polarisée en inverse et exploiter la chaleur dégagée localement par l'avalanche pour transformer cette diode en court-circuit.

REM2: Les PROM à fusibles ont pratiquement disparu.

Cellule EPROM : principe



Les EPROM (Erasable Programmable ROM) peuvent être programmées par l'utilisateur, effacées et reprogrammées un grand nombre de fois (entre 10^3 et 10^6 fois suivant la technologie).

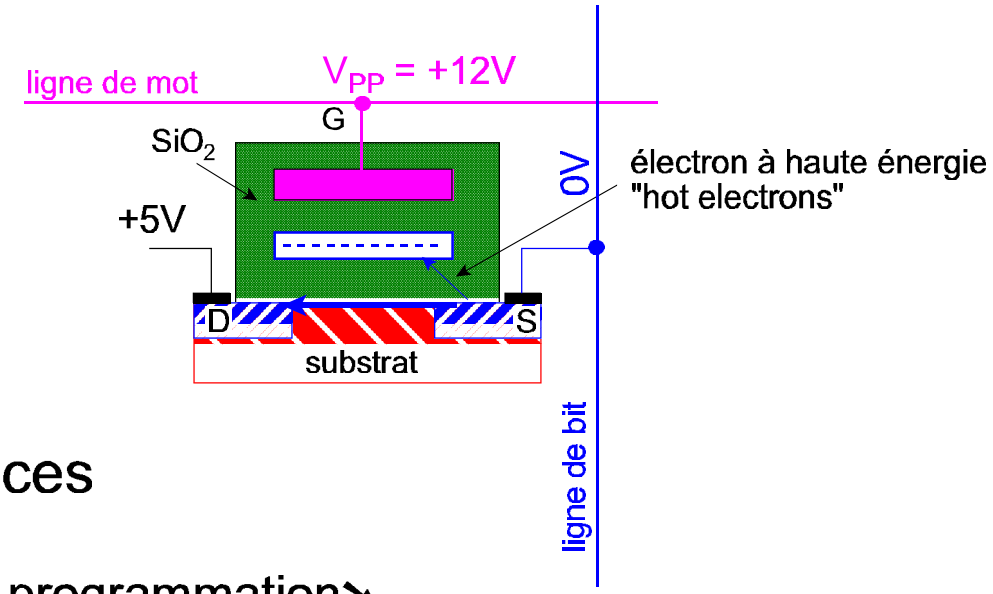
Tous les transistors sont identiques et présentent une double grille. Un transistor non-programmé (ou effacé) se comporte comme un MOS normal, et conduit lorsque l'on active la ligne de mot. La source étant raccordée en permanence à V_{DD}, tous les bits sont donc lus comme HI par défaut.

Programmer un bit en LO consiste à rendre la grille du transistor inopérante. Ceci est réalisé à par interposition, entre la grille normale et le canal, d'une seconde grille dite "flottante" ("floating gate") dans laquelle on arrive à piéger des électrons.
En observant la caractéristique de transfert $i_D(v_{GS})$, on voit que la charge ainsi stockée dans la grille flottante déplace le seuil (threshold) de tension grille-source (c'est-à-dire le minimum de tension V_{GS} pour faire conduire le transistor) de la valeur V_{T1} à la valeur V_{T2}.
V_{T2} étant au-delà de V_{DD}, l'activation de la ligne de mot est incapable de faire circuler un courant dans le transistor.

Nous avons déjà évoqué les principales variantes de modes d'effacement :

- UV-EPROM : effacement aux ultra-violets de toute la mémoire en une fois
- EEPROM ou E²PROM : (Electrically ERasable PROM) effacement électrique mot part mot
- FLASH EPROM/EEPROM : effacement électrique de toute la mémoire -ou d'un bloc- de mémoire à la fois

Cellules (UV)EPROM : programmation



- tendances
  - V_{PP} ↘
  - temps programmation ↘
  - entrée série ("ISP")

Cette figure montre la programmation d'une cellule d'UV-EPROM.

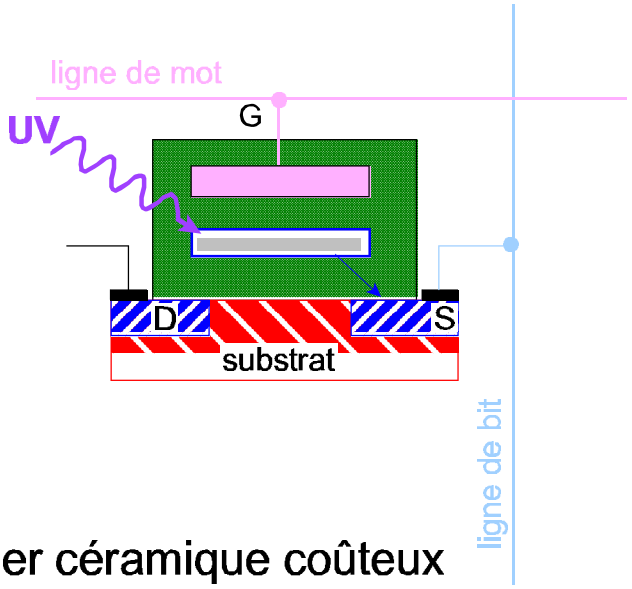
Le drain est toujours connecté à V_{DD} et la source à la ligne de bit. On applique, sur la ligne de mot connectée à la grille, une impulsion de tension plus élevée que l'alimentation normale (tension de programmation V_{PP}, de l'ordre de 12V). Le canal se crée et le transistor se met à conduire; la flèche indique le sens des électrons (de la source vers le drain).

La tension V_{PP} élevée sur la grille crée un champ électrique intense qui permet aux électrons les plus énergétiques de sauter la barrière de potentiel (>3.2 eV) à l'interface avec l'oxyde et traverser l'oxyde jusqu'à la grille flottante. Ce mécanisme s'appelle "injection d'électrons de haute énergie" ou "hot electron injection".

Au bout de quelques centaines de µs à quelques ms, on cesse d'appliquer la tension de programmation. Les électrons sont piégés dans la grille flottante, complètement isolée par un oxyde de qualité; la durée de vie de ces charges est garantie de l'ordre de 10 ans (beaucoup d'EPROM de plus de 20 ans sont encore programmées). L'application d'une tension positive normale sur la grille va attirer les électrons de la grille flottante vers le haut, mais ne sera plus capable de créer le canal.

Les progrès des EPROM ont essentiellement porté sur l'abaissement de la tension V_{PP} et la diminution du temps de programmation à quelques dizaines de secondes (parfois au prix d'une perte en durée de rétention).
L'écriture se fait généralement sur un programmateur spécial, mais les boîtiers à montage de surface n'en permettent plus l'usage, aussi trouve-t'on aujourd'hui des EPROM programmables après qu'elles ont été soudées sur le circuit imprimé. Les données sont entrées en série bit-à-bit via 2 bornes spéciales ("In-circuit Serial Programming").

Cellules (UV)EPROM : effacement



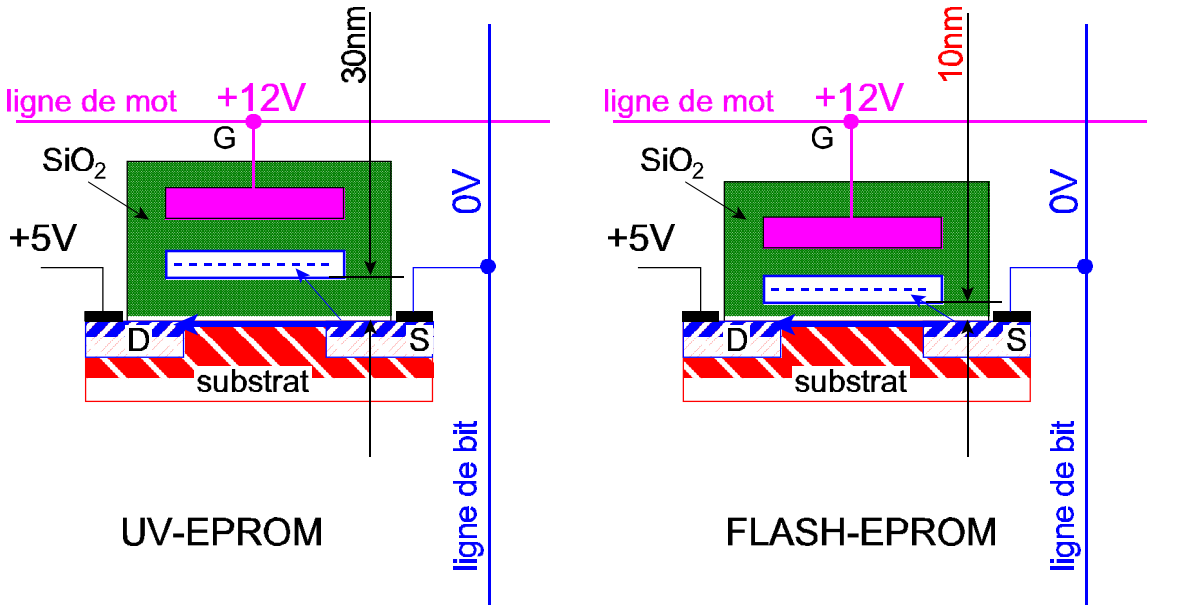
- effacement en bloc
- 15 min aux UV !
- fenêtre quartz =>boîtier céramique coûteux
- boîtier plastique =>pas de fenêtre => OTP

Pour effacer la mémoire, on irradie toute la surface de la puce par des ultra-violets (λ=253,7nm, p=10mW/cm²) pour extraire les électrons des grilles flottantes. On efface donc tout le contenu de la mémoire en une seule opération. L'effacement est malheureusement très long (15 minutes environ), ce qui rend les UV-EPROM fastidieuses à utiliser pour la mise au point des programmes.

Le mécanisme d'effacement rend le boîtier des UV-EPROM assez coûteux car il faut une fenêtre en quartz (le verre arrête les UV). Le coefficient de dilatation du quartz n'est pas compatible avec le plastique des boîtiers ordinaires. Il faut donc un boîtier en céramique beaucoup plus coûteux, normalement réservé aux composants à gamme de température élargie ou aux composants répondant aux normes militaires.

Pour réduire le coût, on peut revenir au boîtier plastique, mais il faut alors supprimer la fenêtre, et l'on perd ainsi la faculté de reprogrammer. Ces mémoires portent le nom de OTP (One Time Programmable).

Cellule FLASH-EPROM : programmation



Les premiers travaux sur les UV-EPROM datent de 1971 et leur commercialisation a décollé vers la fin des années 1970.
Vu le côté fastidieux de l'effacement par UVs, beaucoup d'efforts ont porté depuis sur l'effacement électrique.
L'une des voies qui ont abouti, utilise une cellule dérivée de l'UV-EPROM: la FLASH-EPROM (plus communément appelée FLASH).
Les FLASH sont apparues sur le marché en 1988 et, depuis, leur usage est véritablement en explosion.

A première vue, une cellule EPROM classique (à gauche) et une cellule FLASH (à droite), qui en est dérivée, se ressemblent beaucoup. En regardant plus attentivement, la différence essentielle est la réduction de l'épaisseur de l'oxyde entre la grille flottante et le substrat (de 30nm à 10nm environ).

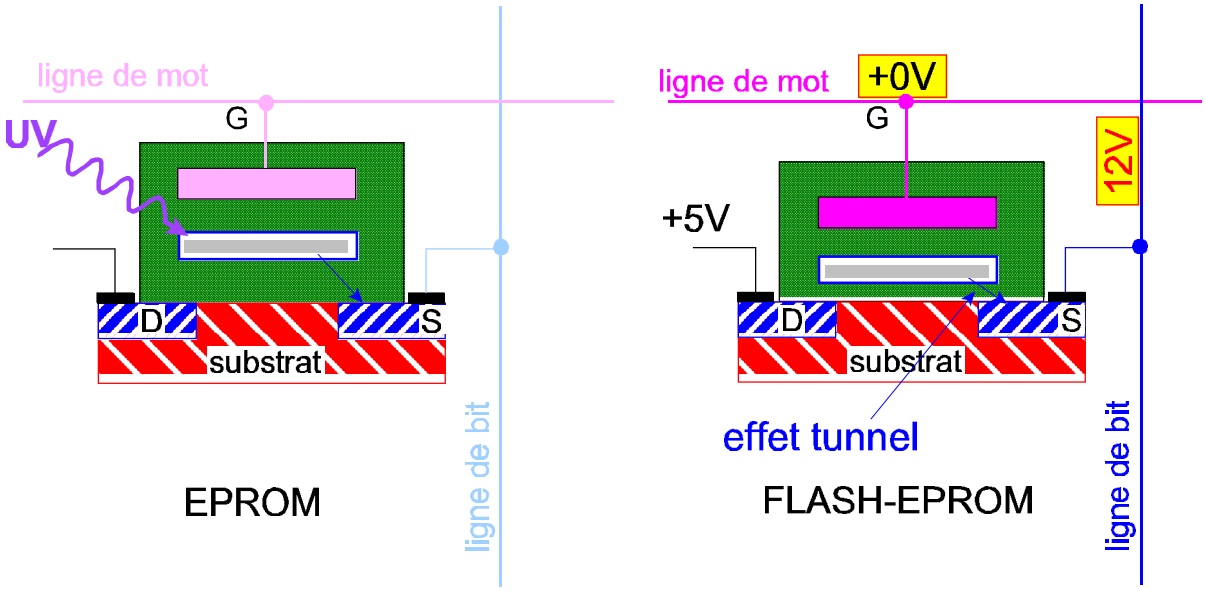
Le mécanisme de programmation par électrons à haute énergie est identique.

On écrit un mot à la fois, avec un temps d'écriture est considérablement plus élevé que pour une RAM (quelques dizaines ou centaines de µs contre quelques ns ou dizaines de ns). La FLASH doit donc être considérée comme une mémoire à accès en écriture occasionnel :
- mise à jour d'un programme
- écriture de données non-volatiles (archivage d'une campagne de mesure)

Le programmeur doit appliquer un algorithme d'écriture, en particulier il faut attendre qu'une donnée soit écrite avant de passer à la suivante. La mémoire doit donc fournir un statut. Les premières adresses sont ainsi souvent des registres de commande et de statut pour la programmation. Chaque constructeur ayant ses propres standards, il est impératif de lire attentivement la notice à ce sujet.

Certaines FLASH possèdent un petit tampon de RAM, dans lequel on peut écrire une rafale de mots à la cadence maximum du processeur. Le programme ne doit donc pas attendre qu'un mot soit écrit avant de charger le suivant, pour autant que le tampon ne soit pas plein, bien sûr.

Cellule FLASH-EPROM : effacement



La FLASH peut-être effacée et réécrite électriquement non seulement dans un programmateur spécial, mais surtout "in circuit" (c'est à dire avec le composant soudé sur le circuit imprimé), ce qui permet une mise à jour aisée des programmes.

L'application d'une tension élevée sur la source, via la ligne de bit, et la mise à la masse de la grille, via la ligne de mot, engendrent un champ électrique inverse de celui d'écriture. Ce champ extrait les électrons de la grille flottante par un mécanisme appelé "effet tunnel Fowler-Nordheim ou Fowler-Nordheim tunnelling". C'est un effet quantique dont l'étude sort du cadre de ce cours. Il est dû à la minceur de l'oxyde entre la grille flottante et le substrat.

L'effacement diffère de la programmation car :
- il n'y a pas de canal créé et donc pas de courant dans le transistor, vu la polarité de la grille
- les électrons n'ont pas besoin d'acquérir une énergie plus élevée que la barrière de potentiel pour migrer. ("cold electron migration").

Le terme FLASH veut dire en fait "effacement éclair" de toute la mémoire.

Un sur-effacement est possible, notamment si l'on essaie d'effacer une cellule qui n'est pas programmée (c'est-à-dire dont la grille flottante n'est pas chargée). Dans ce cas, la cellule ne pourra plus jamais être programmée et la mémoire devient inutilisable. Pour éviter cela, les FLASH contiennent aujourd'hui une logique interne, qui commence par programmer toutes les cellules qui ne le sont pas, avant d'effacer. La progression de l'effacement est un processus interne itératif en boucle fermée, avec arrêt automatique lorsque l'effacement est complet.

Les registres de commande et de statut déjà évoqués permettent au programmeur de provoquer l'effacement "in situ" et de vérifier si le processus s'est convenablement déroulé.

Avantages des FLASH

- non volatile
- effacement électrique rapide (qq s)
- cellule plus petite que l'EPROM
  - moins de contraintes de routage (pas besoin de laisser passer les UV)
  - grosses capacités (plusieurs Mo/boîtier)
- boîtiers moins cher et modernes
  - pas de fenêtre en quartz
  - plus compact et plus plat
- effacement par bloc
  - permet le "boot-loading"
  - utilisation comme mémoire de données par bloc (FLASH DISK, carte de mémoire pour photo ou musique)

Les FLASH n'ont que des avantages par rapport aux UV-EPROM. Les contraintes liées à la nécessité d'éclairer tous les transistors sont levées. Cela permet des transistors plus petits, des connexions internes plus denses et des boîtiers plus compacts et plus plats.

Comme les UV-EPROM, les premières FLASH s'effaçaient complètement en une fois ("bulk erase"), en appliquant les tensions d'effacement à toutes les lignes de mot et de bits. Aujourd'hui, les FLASH sont divisées en blocs effaçables séparément, ce qui permet :

- de mettre en oeuvre le principe du "boot loader" : une partie du programme est immuable et sert à booter le système, notamment pour donner accès à un moyen de communication (port série, lecteur de disquette, ...). Les autres blocs de la FLASH peuvent être effacés et mis à jour à travers ce canal. Un exemple est la mise à jour du BIOS d'un PC via une disquette.
- d'utiliser les FLASH comme mémoire de données non-volatile (stockage de mesures, de musique, de photos, "boîte noire")
- d'utiliser les FLASH comme mémoire de masse pour remplacer les disques durs (disque "solid state" ou FLASHDISK). Le temps d'accès aux données est divisé par 105, avec une fiabilité nettement plus élevée; le nombre de réécritures de l'ordre du million donne une bonne durée de vie, pour autant que l'on égalise l'utilisation des secteurs virtuels. Le prix au bit est malheureusement 1000 fois plus élevé que celui du médium magnétique, ce qui réserve les FLASH DISK à des applications spéciales (pour le moment).

Inconvénients des FLASH

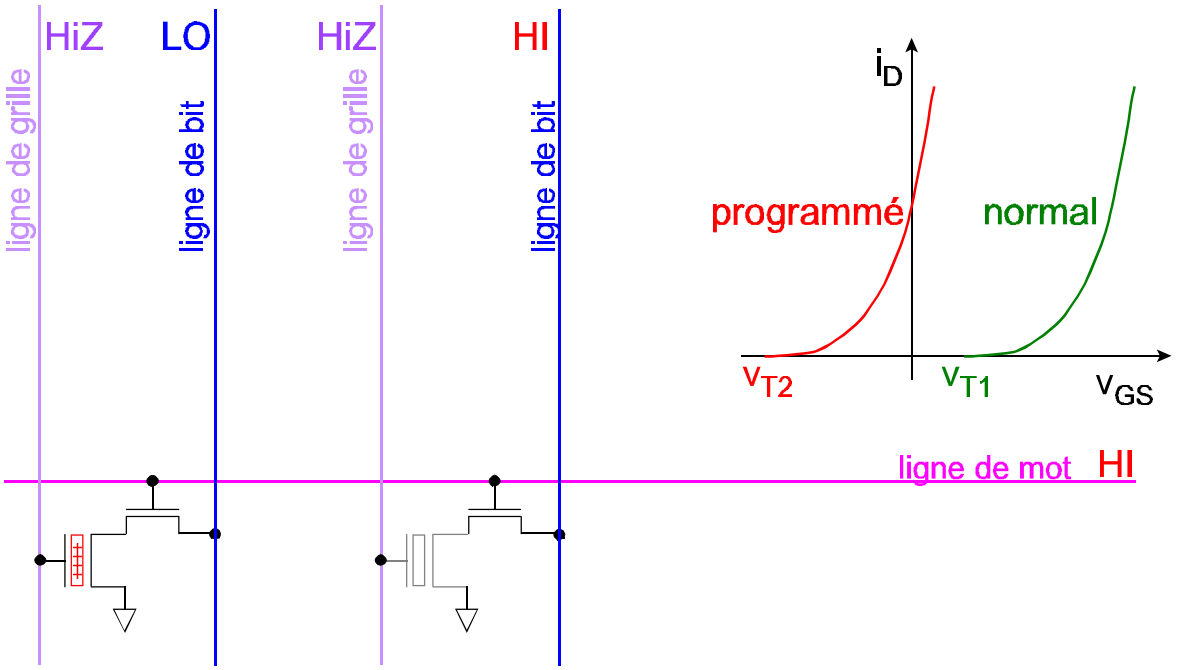
- n'est pas une bonne RAM
  - temps d'écriture long (100 µs) avec un bout de programme à exécuter
  - pas moyen de réécrire un mot sans d'abord effacer tout le bloc qui le contient
  - nombre d'écritures fini (10E5 à 10E6)
    - OK pour programmes ou blocs de mesures
    - impossible pour données fréquemment mises à jour
- restrictions d'utilisation
  - souvent pas moyen d'exécuter depuis un bloc quand on efface un autre => on doit copier le boot-loader en RAM

La FLASH est accessible en écriture, mais ne peut pas être employée comme RAM.
Dans les systèmes à microprocesseurs, les mots d'une RAM sont très fréquemment réécrits (variables temporaires, pile, heap). Les principes de la FLASH ne permettent pas de répondre à ces besoins
- le temps d'écriture est trop long
- chaque fois que l'on écrase une donnée, il faut effacer tout le bloc qui la contient
- enfin on épuiserait vite le capital des 10^5 à 10^6 cycles d'écriture autorisés.

Les opérations d'écriture doivent donc être considérées comme occasionnelles (données ou programme qui doivent être stockés pour une longue période)

Beaucoup de FLASH imposent des restrictions sur l'utilisation pendant une écriture ou un effacement. En particulier, l'algorithme de programmation que doit coder le programmeur ne peut pas toujours s'exécuter depuis un bloc de la même FLASH que celle que l'on programme. Il faut donc disposer d'une autre mémoire (RAM ou EPROM exemple) externe pour stocker le bout de code correspondant.

Cellule EEPROM : lecture



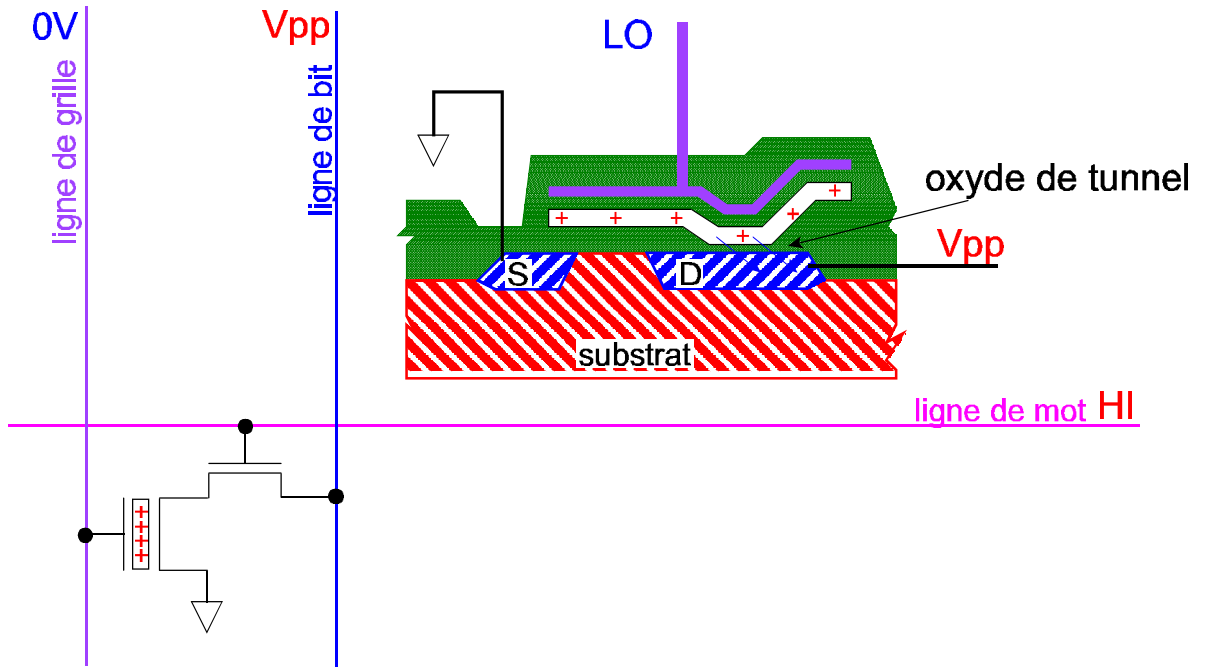
En parallèle sur le développement des UV-EPROM, des efforts de recherche ont porté sur d'autres cellules à grille flottante et à effacement électrique. Deux pistes ont abouti :
- l'EAROM (Electrically Alterable ROM) aujourd'hui abandonnée
- l'EEPROM (Electrically Erasable PROM) qui est assez répandue et que nous allons décrire ici.

Un exemple de cellule est présenté sur cette figure. Le fonctionnement est toujours basé sur un transistor à grille flottante, mais cette fois le transistor programmé voit cette grille chargée positivement, et il devient conducteur en permanence. Sa caractéristique de transfert est translatée vers la gauche.

La source est reliée à la masse en permanence. Le drain est connecté à la ligne de bit via un transistor de sélection commandé par la ligne de mot. Les grilles de tous les transistors d'une colonne sont reliées à une ligne de grille commune.

Pour la lecture
- la grille du transistor est laissée flottante
- on active la ligne de mots :
  - un transistor programmé est conducteur et met la ligne de bit à LO (l'ampli de "sense détecte un courant dans la ligne de bit)
  - un transistor normal est coupé et laisse la ligne de bit à HI (pas de courant dans la ligne de bit)

Cellule EEPROM : programmation



La programmation (charge de la grille flottante) se fait, comme dans l'effacement des FLASH, par des électrons à faible énergie en exploitant l'effet tunnel dans une zone d'oxyde très mince (de l'ordre de 10nm) entre la grille flottante et le drain. La plupart des technologies des constructeurs portent un nom incluant les lettres TOX (Tunnel OXyde)

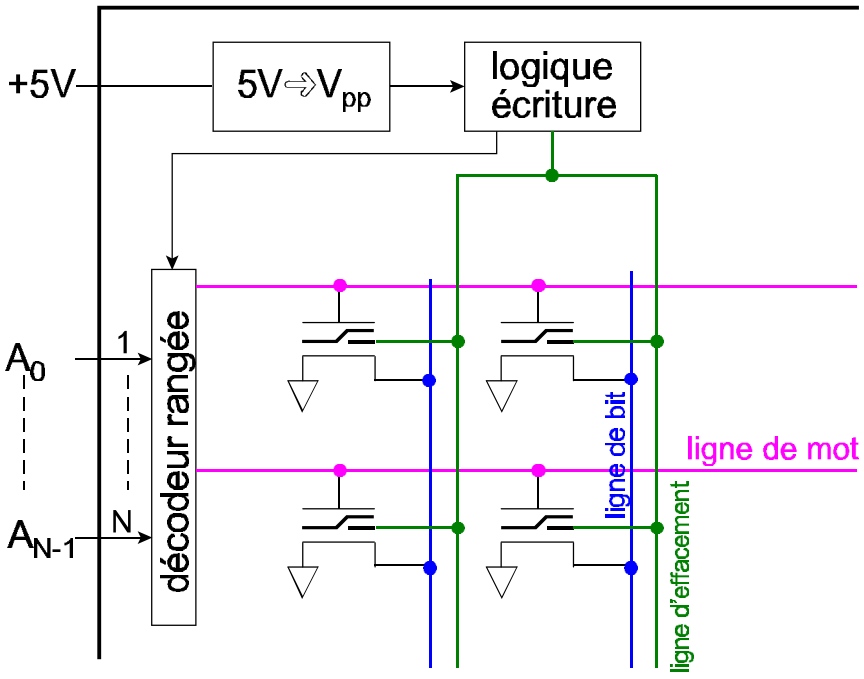
Pour programmer :
- on sélectionne le transistor en activant la ligne de mot
- on met la ligne de grille à la masse
- on place la ligne de bit, donc le drain, à la tension de programmation V_{PP} (par ex 12V)

Par effet tunnel, des électrons de la grille flottante sont attirés par le drain, migrent à travers l'oxyde, et la grille se charge positivement.

Remarquons que, dans cette structure, on peut programmer une cellule à la fois. En réalité, comme les échanges avec le bus de données se font par mot, on programme simultanément tous les bits d'un mot que l'on désire mettre à LO, pour gagner du temps.

Comme dans les FLASH, le temps d'écriture est long (quelques dizaines de µs) et nécessite un algorithme au niveau du programme (voir mémoires FLASH).

FLASH EEPROM basse tension



La figure monte la structure d'effacement d'une mémoire FLASH à cellule type EEPROM, avec électrodes spéciales pour effacement accéléré.
La présence de telles électrodes impose de router dans la puce une ligne supplémentaire d'effacement parallèle à chaque ligne de bit (soit une par colonne).

La tension d'alimentation V_{DD} de la logique (5V ou 3.3V) est insuffisante pour effacer et programmer. A bord d'un ordinateur on dispose généralement d'une alimentation +12V, mais à bord d'instruments sur batteries, ce n'est pas le cas. C'est pourquoi les fabricants implantent un micro-convertisseur dans le boîtier même de la mémoire pour élever la tension de V_{DD} vers V_{PP}.

Cette tension est distribuée à toutes les lignes d'effacement, via le séquenceur logique responsable des opérations d'effacement et d'écriture. La liaison vers le décodeur de rangées permet d'activer simultanément toutes les lignes de mot appartenant au bloc à effacer.

Avantages des EEPROM

- non volatile
- effacement électrique plus rapide que la FLASH
- effacement/écriture par mot
  - surtout utile pour mémoires de configuration et de paramètres (prix au litre par ex.)
- il existe des versions "haute performance" pour MIL ou spatial
  - ECC
  - gamme T° étendue
  - durci aux radiations

Les EEPROM n'ont que quelques avantages sur les FLASH. Elles sont plus rapides à effacer que les FLASH dérivées des EPROM.
Leur principal avantage est la possibilité d'effacer, puis de réécrire mot-à-mot, ce qui en fait le composant idéal pour stocker quelques données indépendantes de petite taille.
Enfin les EEPROM peuvent être rendues particulièrement fiables pour des applications spatiales grâce à :
- des codes détecteurs et correcteurs d'erreur (ECC)
- une gamme de T° étendue
- un "durcissement" permettant de mieux résister à des altérations par les rayons cosmiques.

Inconvénients des EEPROM

- cellule volumineuse
  - 2 transistors
  - redondance dans les versions haute fiabilité
- coût/bit >> FLASH
- limité à qq 10Ko

La possibilité de programmer et d'effacer mot-à-mot se paie par un second transistor de sélection et une cellule plus volumineuse que la FLASH.
Les EEPROM les plus fiables reposent sur des redondances (cellules à 4 transistors, ECC).
Comme le coût des mémoires est principalement lié à la surface de silicium, la faible densité de transistors des EEPROM donne un coût par bit élevé.
On ne trouve donc pas de boîtiers pouvant stocker des quantités équivalentes aux FLASH.

Mémoires programme : tableau résumé

type	qualités	défauts
(mask)ROM	- coût faible	- grande série (>10 000) - manque de souplesse - délais de fabrication
(UV)EPROM	- souplesse - écriture in situ (récent)	- très coûteux - temps d'effacement UV - 1K écritures
OTP	- peu coûteux	- pas d'effacement
E²PROM	- souplesse - écriture in situ par byte - effacement par byte	- très coûteux - écriture lente - qq 100K écritures
FLASH	- souplesse - écriture in situ par byte - 1 M écritures - faible coût	- effacement par bloc - écriture lente

Le tableau reprend les mérites respectifs des différentes catégories de mémoire programme.

La MASK ROM (en abrégé ROM) a la plus forte densité (cellule de petite taille) et donc le coût le plus faible par bit.
Ses inconvénients sont :
- l'impossibilité de retoucher le programme, qui doit donc être stable
- le programmeur doit envoyer un fichier au fabricant de la mémoire, commander une quantité importante (>10 000 pièces) et se mettre dans la file d'attente du fabricant, ce qui peut prendre plusieurs semaines.

L'UV-EPROM (en abrégé EPROM) est souple puisque l'utilisateur peut effacer un programme et le remplacer par une version plus récente. Cette opération est limitée à quelques centaines ou milliers de fois, suivant le constructeur.
L'effacement aux UV est lent (15 minutes) et donne un coût élevé (densité limitée, boîtier céramique)
Pour réduire le coût, on revient au boîtier plastique, mais sans fenêtre. On perd alors toute possibilité de reprogrammation et la mémoire est dite OTP (One Time Programmable).

L'E²PROM est la plus souple, car elle peut être écrite et effacée mot par mot de 10K à 100K fois. L'écriture est toutefois lente, comme pour les EPROM. Leur plus gros défaut est le coût, car l'effacement par mot se paie par le doublement du nombre de transistors comparativement à l'EPROM.

La FLASH est un excellent compromis, et remplace partout l'EPROM. Ses seuls défauts sont l'effacement par bloc et le temps d'écriture, ce qui peut être gênant pour des données, mais pas pour les programmes.

Mémoires non-volatiles : stockage de programmes ou de tables de constantes

type	convient mieux pour
ROM	très grande série d'un produit stable
OTP	petite ou moyenne série d'un produit stable
FLASH	prototype et moyenne série; notion de "bootblock"
EPROM	prototype et petite série (préférer la FLASH)
E²PROM	peu utilisé : applications spéciales (MIL ou spatial)
RAM+pile	seulement si présente à cause des données

L'utilisation majeure des mémoires non-volatiles est bien sûr le stockage de programmes. On peut aussi avoir besoin de tables volumineuses de constantes (fonctions transcendantes, idéogrammes)

Classons les mémoires par ordre croissant de coût par bit croissant :

- la ROM reste la technologie la moins chère en grande série, surtout pour les micro-contrôleurs, où la version ROM est pratiquement au même prix que la version sans mémoire programme interne. La durée de rétention des données sera égale à la durée de vie du composant. Le contenu doit nécessairement être très stable, car l'investissement pour changer le programme est très élevé.
- l'OTP convient pour des séries moyennes mais exige également la stabilité du programme.
- la mémoire FLASH est le meilleur compromis actuel entre le prix/bit, la facilité de programmation et le confort qu'offre la reprogrammabilité. Elle convient très bien pour les prototypes et les moyennes séries. Les versions avec "boot block" inaltérable sont très intéressantes pour la mise à jour du système.
- l'EPROM s'applique dans le même créneau que la FLASH tout en étant moins pratique et plus chère; elle est donc condamnée à terme
- l'EEPROM est très chère comme mémoire programme. Seules des applications spéciales (militaires et spatiales) justifient son coût, en particulier pour les versions à haute fiabilité.
- les RAM avec pile au lithium incorporée ne sont pas conçues a prori pour les programmes; toutefois, dans de petits systèmes où l'on a besoin de cette RAM pour les données et où il reste quelques Ko libres pour le programme, c'est une bonne solution si on se contente de quelques années de durée de vie.

Mémoires non-volatiles : stockage de données

type	convient mieux pour
FLASH	réécritures lentes, peu fréquentes et par bloc: FLASHDISK, photo numérique, baladeurs, enregistreurs médicaux
E²PROM	réécritures lentes, peu fréquentes et par mot: mémoires de configuration / calibration
RAM+pile	mémoire de configuration (souvent couplée à une horloge/calendrier) réécriture fréquente et rapide (buffers cycliques, tampons d'acquisition)

Examinons les mérites respectifs des différentes mémoires non volatiles pour le stockage de données.

La mémoire FLASH ne convient que si l'on est pas pénalisé par la lenteur d'écriture et par nombre de cycles d'écriture (environ 10^6). Les "disques" en mémoire FLASH en sont un bon exemple, car l'écriture en FLASH, si elle est lente par rapport aux RAM, reste 1000 fois plus rapide que sur disque dur magnétique. Les gestionnaires de disque en FLASH s'arrangent pour uniformiser l'emploi des "secteurs" pour éviter de d'écrire systématiquement sur les mêmes zones.
La mémoire FLASH est aussi très utilisée dans des applications "grand public" comme les baladeurs ou les appareils photos numériques; la partage en blocs effaçable séparément est primordial pour cette dernière application.

Si l'on veut mémoriser des données de petite taille, et indépendantes les une des autres (constantes de calibration, date d'événement, paramètre d'un système, numéro de série) l'E²PROM est la meilleure option, puisque l'on peut réécrire un mot sans altérer les autres.

Enfin, certaines applications demandent d'écrire des rafales rapides de données, ou de rafraîchir les données avec grande une fréquence; dans ce cas on risque d'épuiser le nombre de cycles des FLASH ou EEPROM avant la durée normale de la vie de l'équipement. La RAM avec pile incorporée apporte la vitesse et le nombre illimité d'écritures.