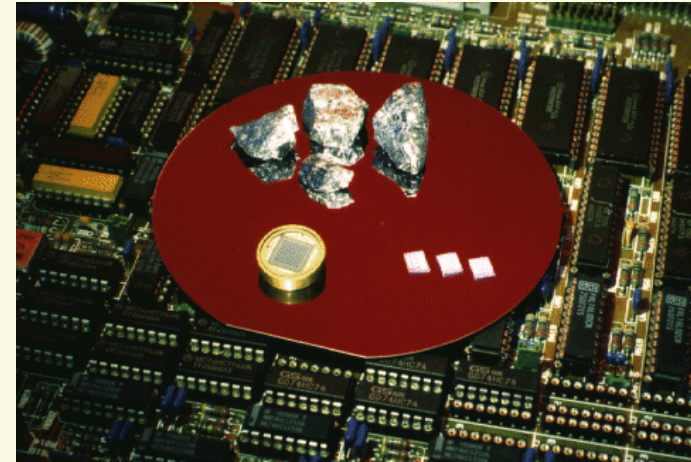
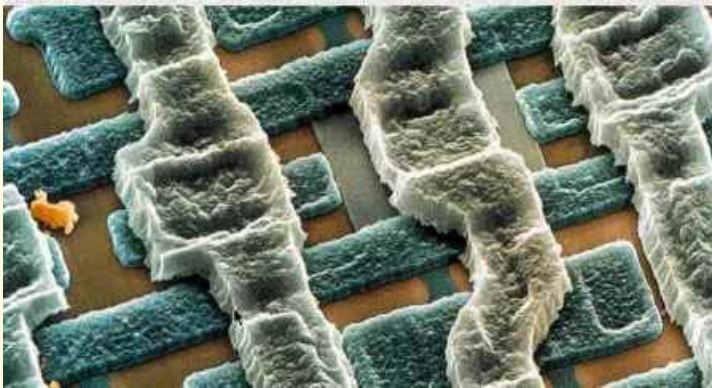


# Micro électronique



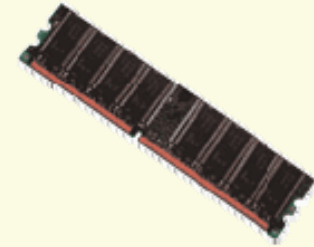
## Éléments de mémoire



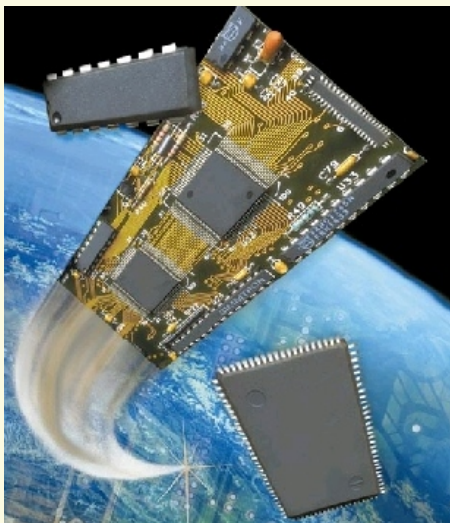
Dominique GINHAC  
dginhac@u-bourgogne.fr



# Éléments de mémorisation



## Généralités

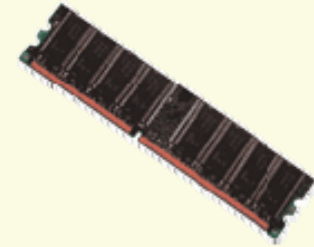


Etre capable de réaliser **un circuit** permettant de **stocker** des **informations numériques** (de manière définitive ou non)

Objectifs :

- ✓ **Capacité** de stockage élevée,
- ✓ **Temps d'accès** le plus faible possible,
- ✓ **Surface** la plus faible possible.

# La fonction mémoire

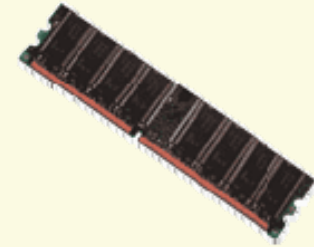


Si **Ad** est l'adresse, **D** est la donnée et **Cont** est un ensemble de container adapté

- ✓ **Mode écriture**  $\text{Cont}(\text{Ad}) \leftarrow D$   
« je pose mes chaussettes dans le tiroir 3 »
- ✓ **Mode lecture**  $B \leftarrow \text{Pos}(\text{Ad})$  alors ( $B$  vaut  $D$ )  
« je prend mes chaussettes dans le tiroir 3 »
- ✓ **Mode rétention**  
« mes chaussettes sont dans le tiroir 3 »

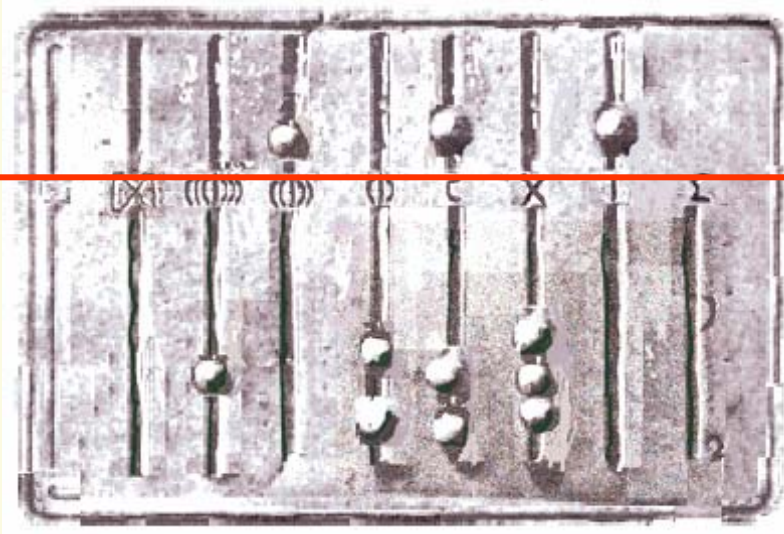
Exploitation de phénomènes bistables et Découpage en bit (cellule élémentaire)

# Un peu d'histoire

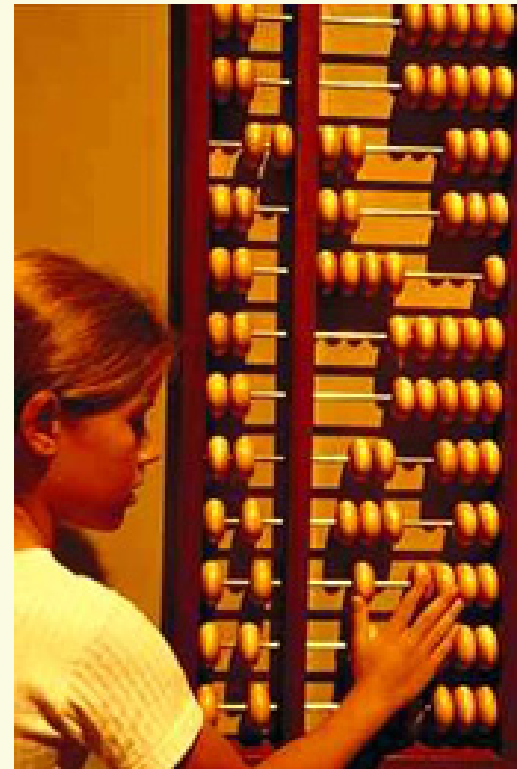


Calculus - Abaques - Bouliers ...

Mémoire de sauvegarde

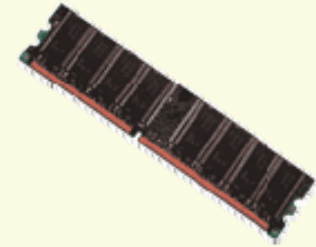


**Abaque romain** : chiffres notés  
par des cailloux (calculus)





# Un peu d'histoire

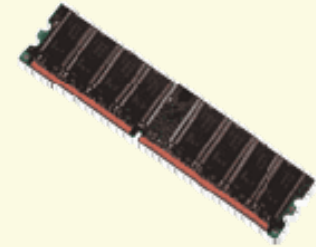


## 1801 : Métiers à tisser à cartes perforées

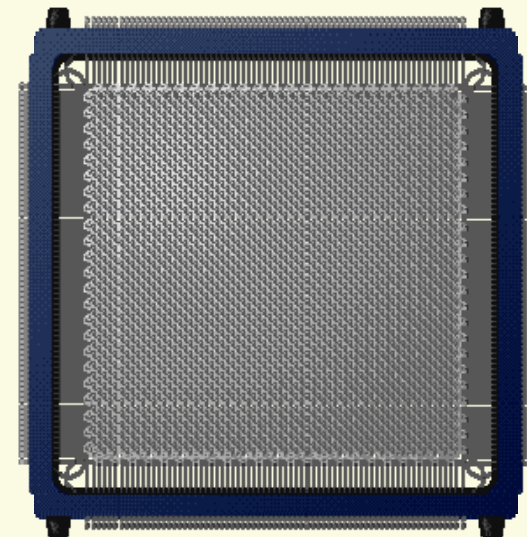
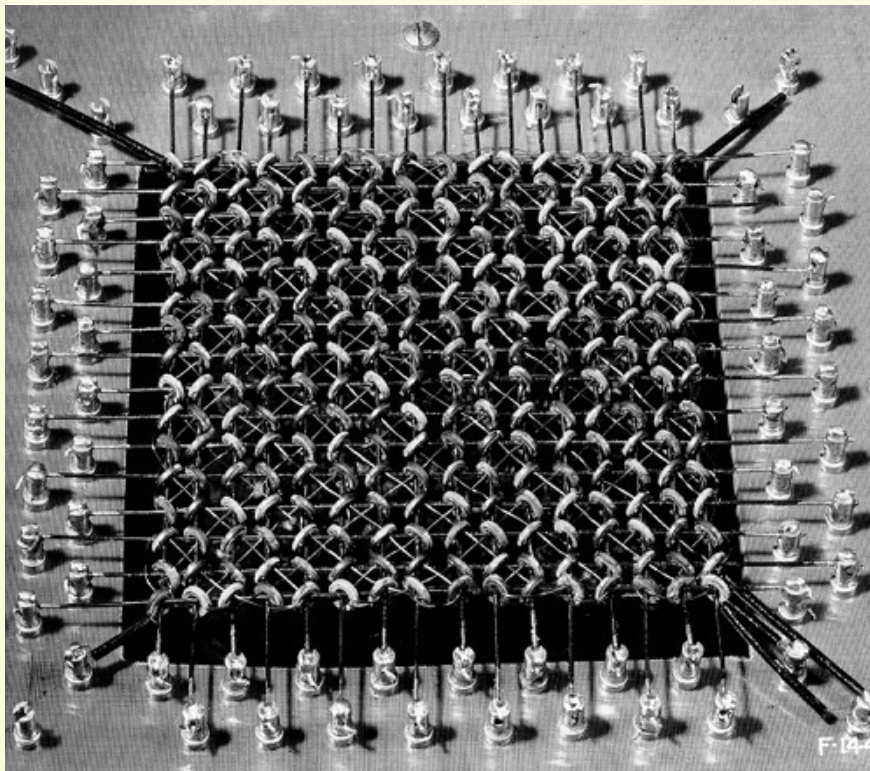
- ✓ Langage binaire
- ✓ Programme enregistré



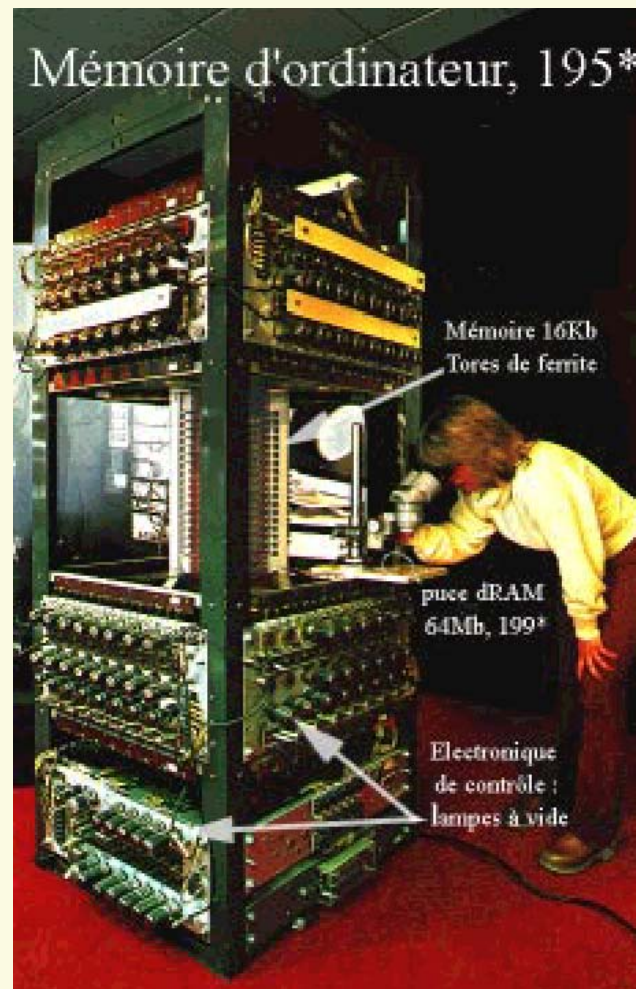
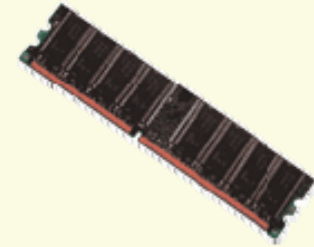
# Un peu d'histoire



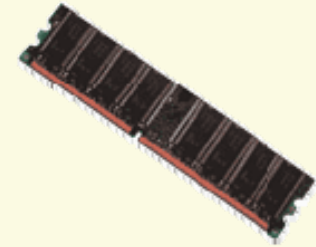
1950 : Mémoires à tores



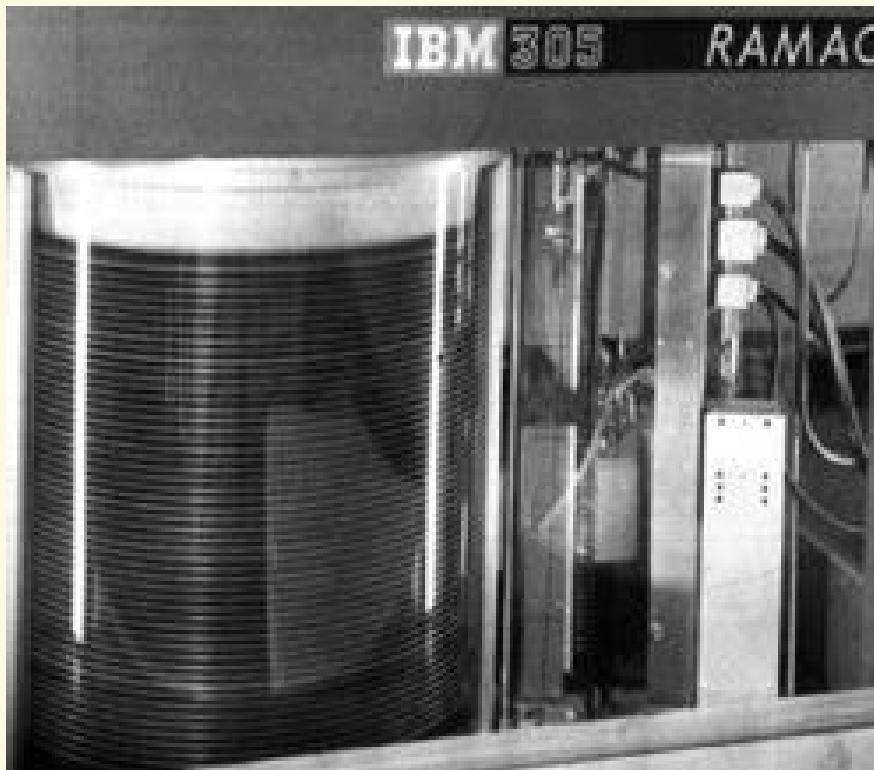
# Un peu d'histoire



# Un peu d'histoire



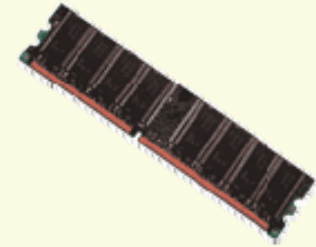
1956 : Premier disque dur (RAMAC de IBM)



50 disques  
61 cm de diamètre  
5 Mo.



# Un peu d'histoire



## La suite...

1970 : DRAM 1Kbit (1103 d'INTEL) Circuit le plus vendu à l'époque

**1971 : Processeur 4004 d'INTEL : 15/11/1971 (2250 Transistors Bipolaires 108 KHz, 4bits, 604 mots ad.)**

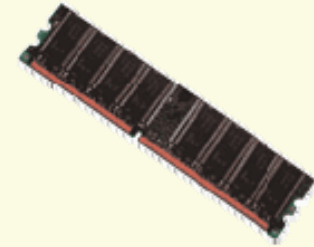
1972 : PROM à fusibles

1973 : Internet (25 machines)

1975 : PLD Signetics (ancêtres des FPGA)

1984 : CD-ROM

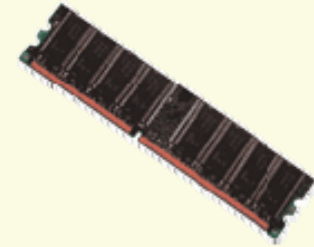
# Quelques définitions



## Sauvegarde des informations

- **Statique** : qui se maintient indéfiniment (tant que l'alimentation est présente)
- **Dynamique** : l'information se dégrade au cours du temps et doit donc être rafraîchie périodiquement
- **Volatile** (ou VIVE) : l'information disparaît avec l'alimentation
- **Non-volatile** (ou MORTE) : l'information est conservée même sans alimentation

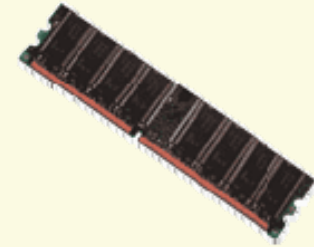
# Quelques définitions



## Accès aux informations

- A **lecture-écriture symétrique** : les opérations de lecture et d'écriture prennent le même temps
- A **écriture lente** et **lecture rapide** (Mémoires Flash)
- A **écriture une seule fois** (PROM) : composant programmable
- A **lecture seule** (ROM) : composant configuré à la fabrication

# Quelques définitions

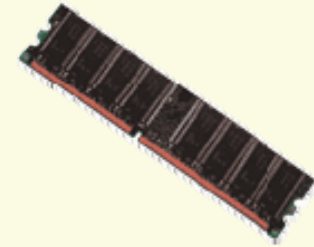


## Mode d'accès

- A **accès séquentiel** : le temps d'accès aux données dépend de leur position de stockage (cassette vidéo)
- A **accès aléatoire** : toutes les données sont accessibles dans le même temps (cd-rom)
- **Associative** : le résultat recherché est la position d'une donnée  
 $\text{Résult} \Leftarrow \text{Ad tel que } (\text{Cont}(\text{Ad}) = D)$   
« quel est le tiroir contenant des chaussettes bleues ? »



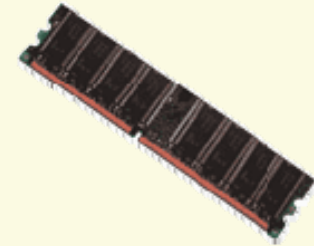
# Quelques définitions



## Technologie

- **Mode veille** : le composant consomme très peu dans ce mode
- **Multiplexage d'adresses** : permet de diviser par deux le nombre de fil d'adresses. Donne accès à des modes rapides (page)
- **Asynchrone** : pas d'horloge, les contrôles doivent être présents pendant toute la durée de l'opération
- **Synchrone** : toutes les opérations sont sous le contrôle d'une horloge

# Éléments de mémorisation



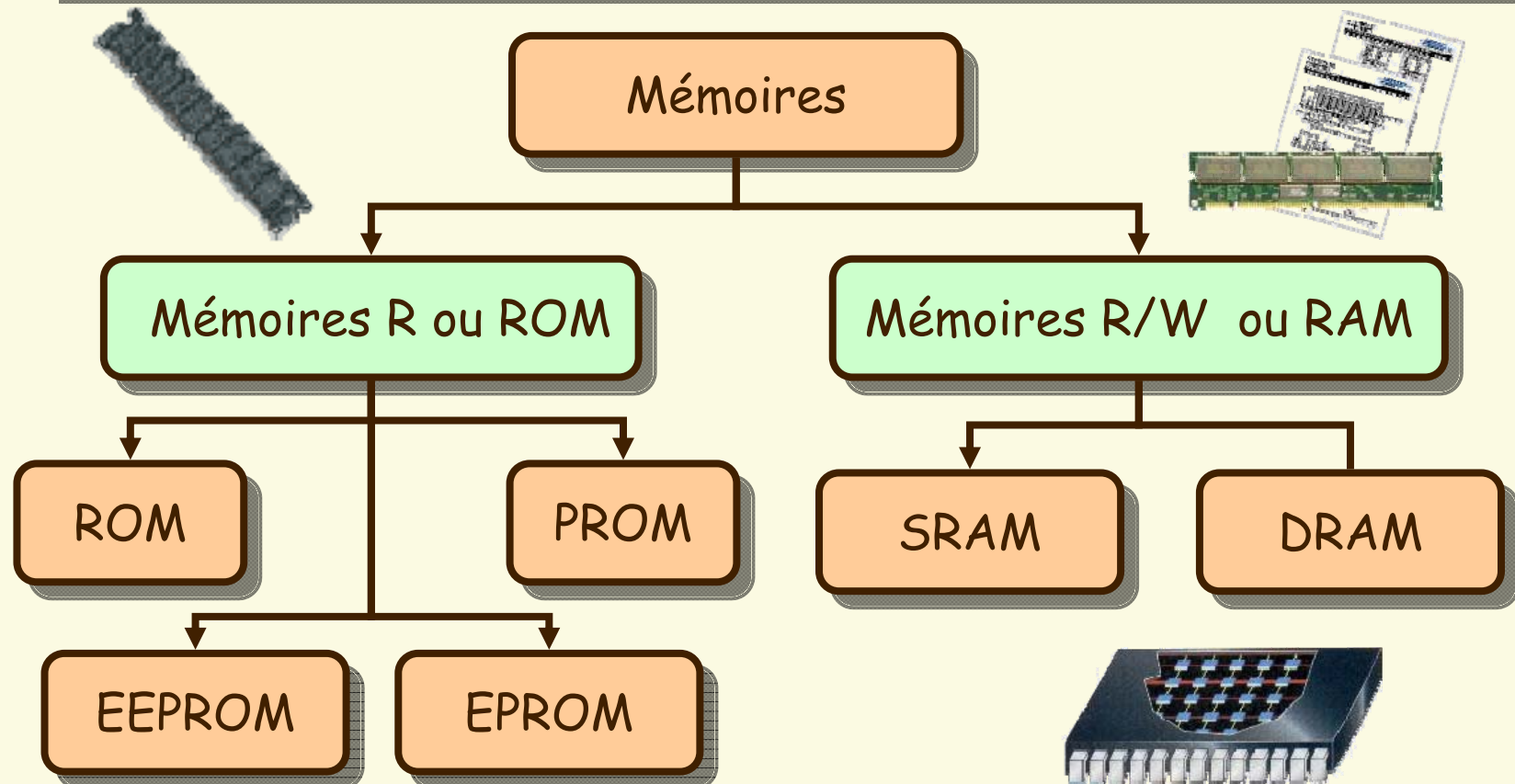
Un marché en constante évolution

ITRS Technology Nodes and Chip Capabilities <sup>2</sup>				
	2004	2007	2010	2018
DRAM Half-Pitch (nanometers)	90	65	45	18
DRAM Memory Size (mega or gigabits)	1G	2G	4G	32G
DRAM Cost/Bit (micro-cents)	2.7	0.96	0.34	0.021
Microprocessor Physical Gate Length (nanometers)	37	25	18	7
Microprocessor Speeds (GHz)	4.2	9.3	15	53

Source : 2003 International Technology Roadmap for Semiconductors

# Éléments de mémorisation

## Classification



# Éléments de mémorisation

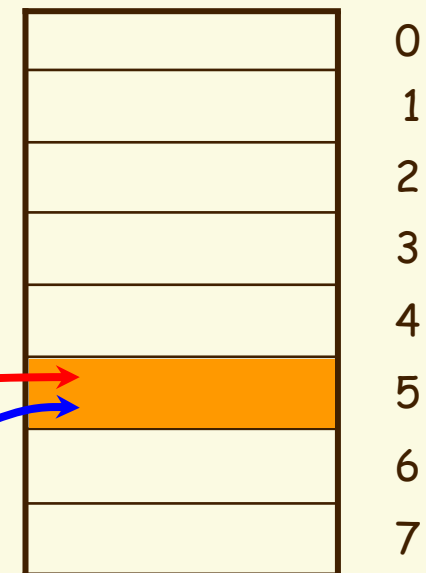
## Mémoire vue de manière simpliste

Une **mémoire** est un tableau à 1 dimension composé de **N mots** de **M bits**

Pour écrire : « je pose mes chaussettes dans le tiroir 5 »

Mémoire[**5**] =

**3**





# Éléments de mémorisation

## Mémoire vue de manière simpliste

Une **mémoire** est un tableau à 1 dimension composé de **N mots** de **M bits**

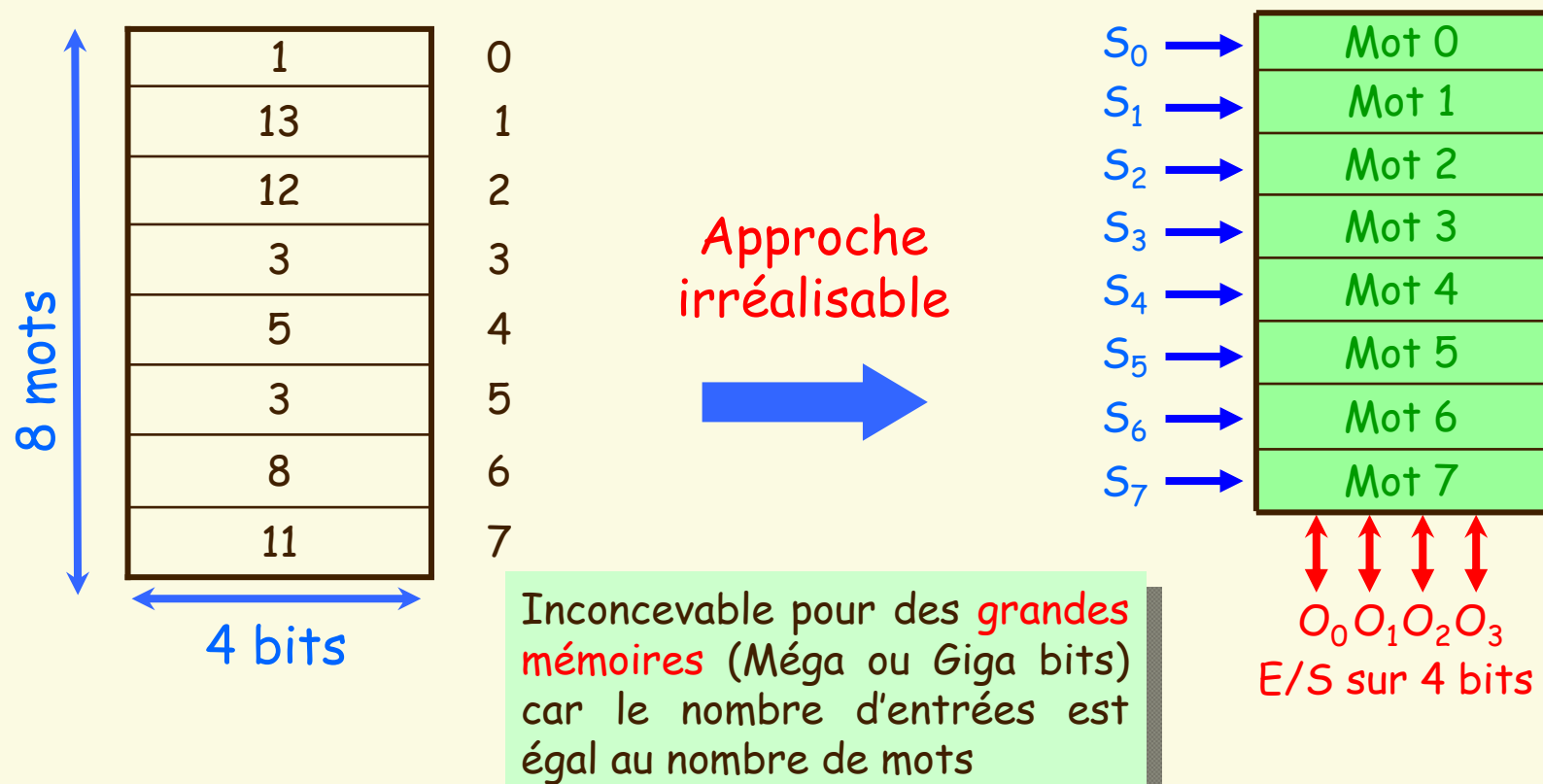
Pour lire : « je prend mes chaussettes dans le tiroir 5 »

Mémoire[**5**] = ?

1	0
13	1
12	2
3	3
5	4
3	5
8	6
11	7

# Éléments de mémorisation

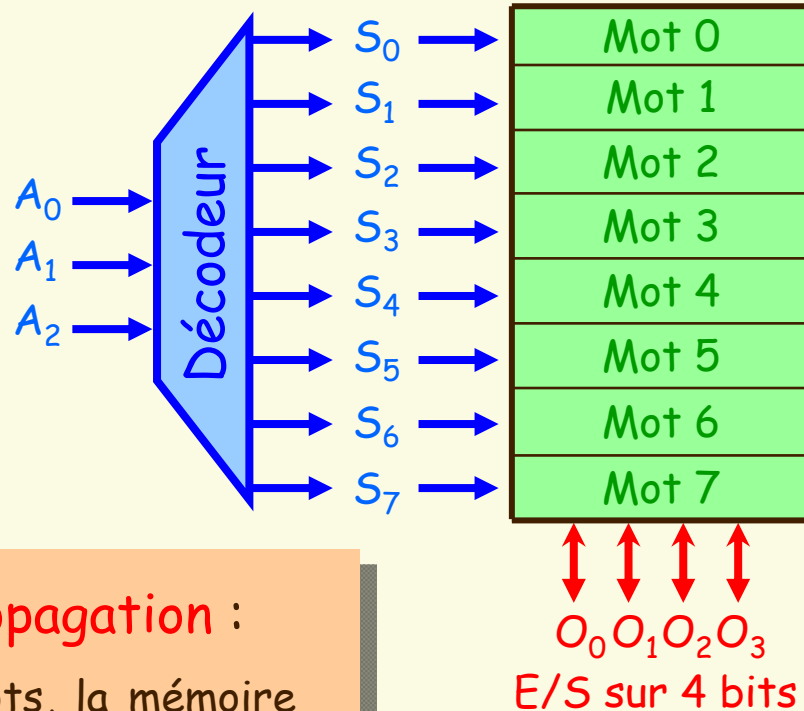
## Mémoire vue de manière simpliste



# Éléments de mémorisation

## Ajout d'un décodeur en entrée

L'ajout d'un décodeur va permettre de réduire le nombre d'entrées de  $2^N$  à  $N$



Problèmes de temps de propagation :

Pour une mémoire de 1 Mega mots, la mémoire est  $2^{20} / 2^3$  plus grande en hauteur qu'en largeur

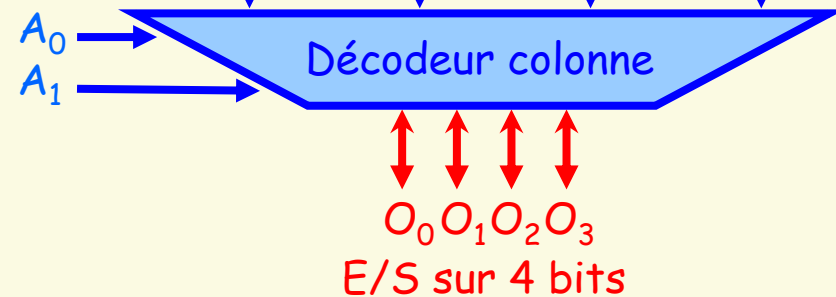
# Éléments de mémorisation

## Plusieurs mots par lignes et ajout d'un décodeur

Sélection de tous les mots d'une même ligne (Ex : Mots 8 à 11)



Le décodeur colonne permet de ne retenir qu'un seul mot sur une ligne donnée (Ex : Mot 9)



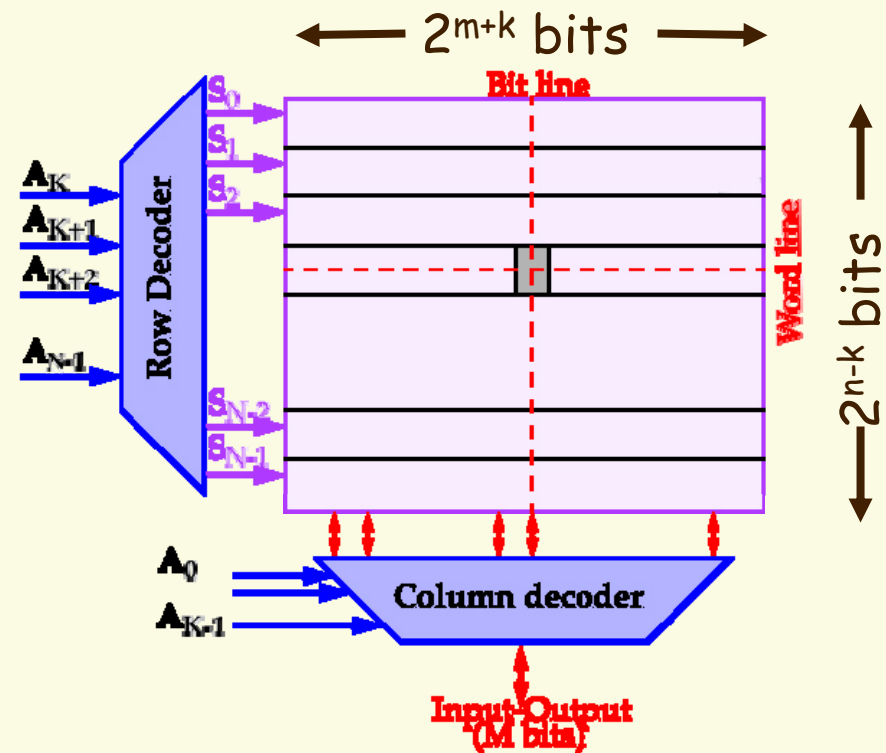


# Éléments de mémorisation

## Bilan : un exemple réaliste

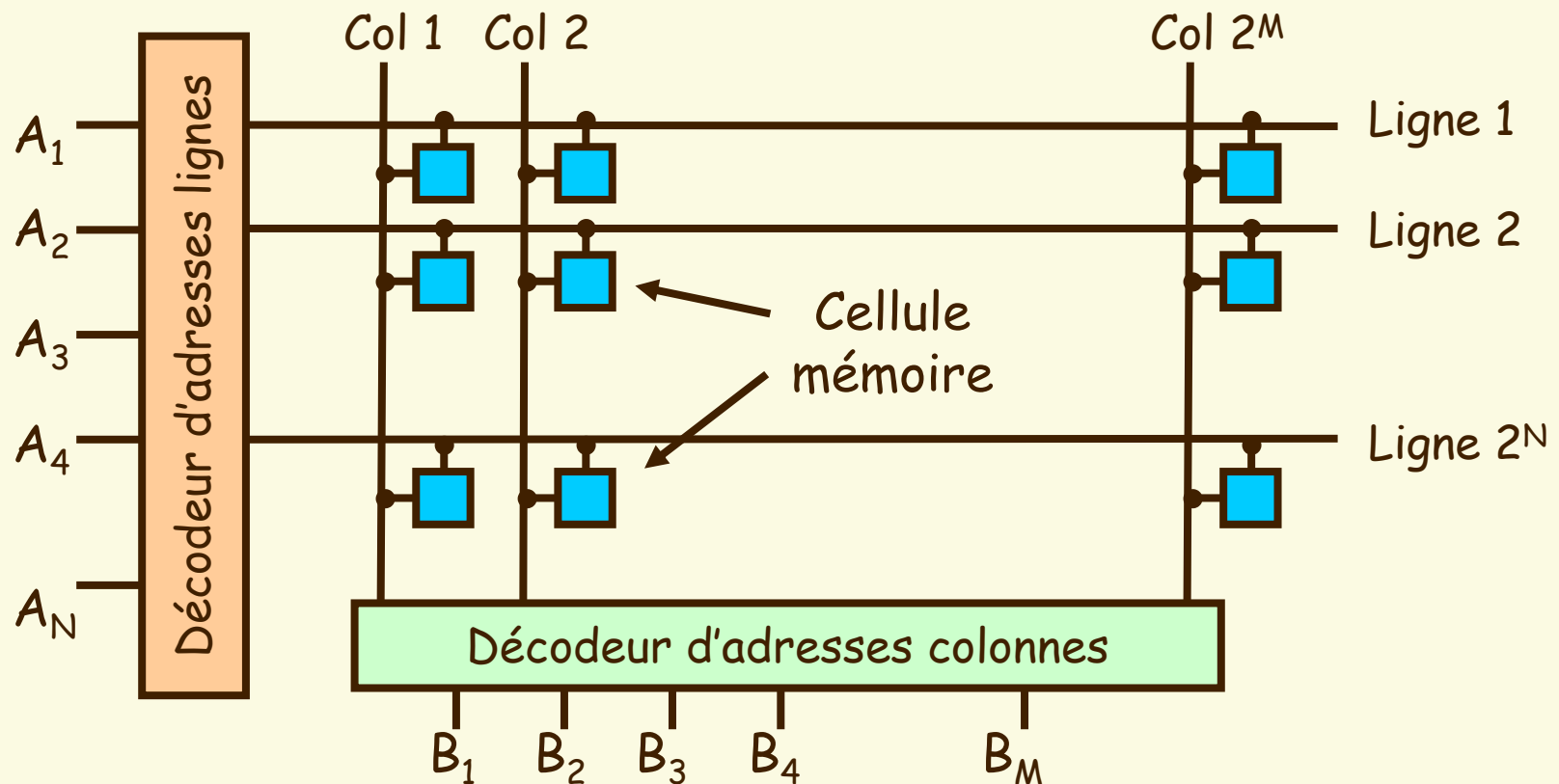
Ex : Mémoire de 1 Mo  
(soit  $2^{20} = 1\,048\,576$   
mots de  $2^3=8$  bits)

La mémoire est donc  
organisée en  $2^{12}$  (4096)  
**lignes** de  $2^{11}$  (2048) bits  
soit **256 mots de 8 bits**  
**par ligne**



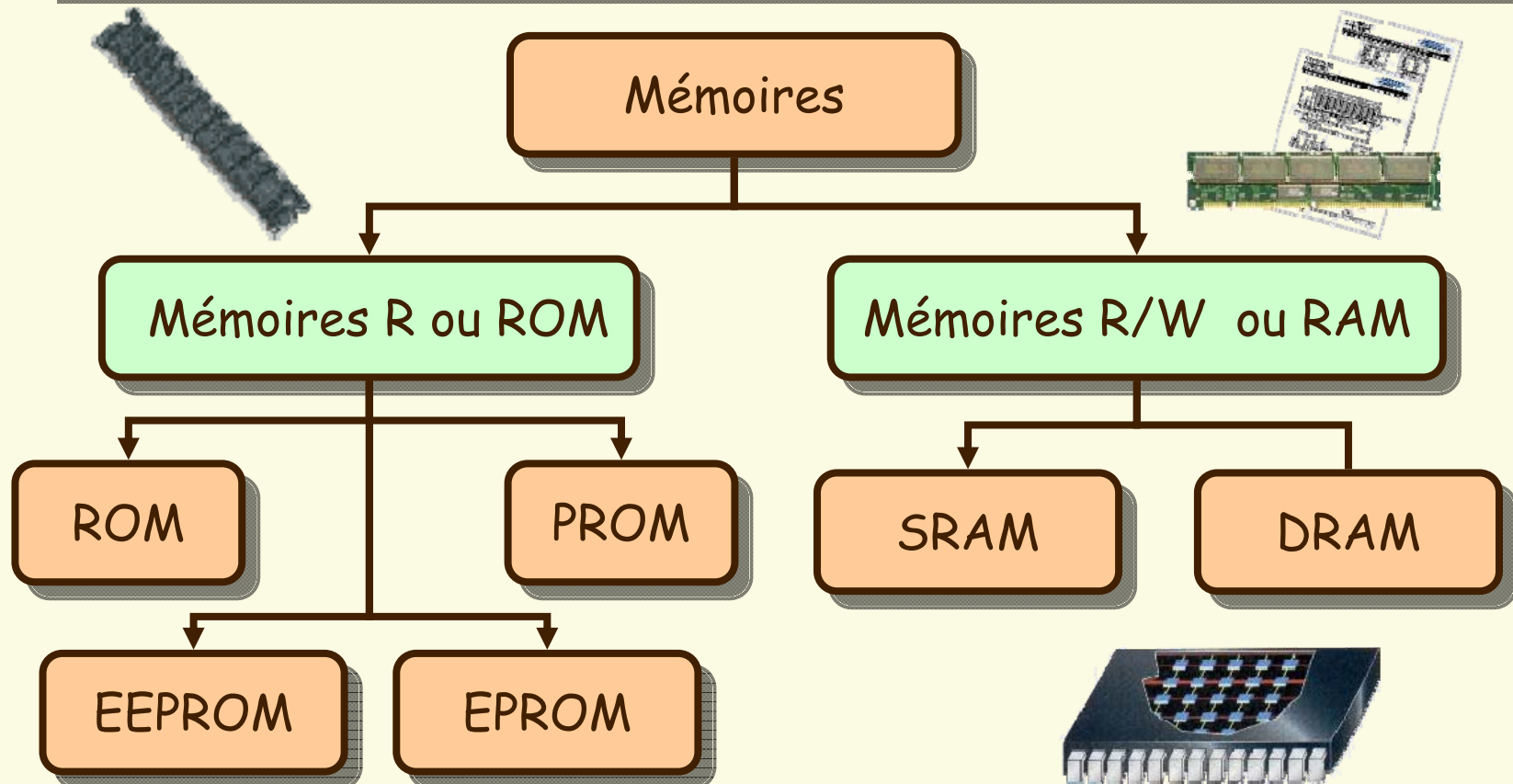
# Éléments de mémorisation

Bilan :



# Éléments de mémorisation

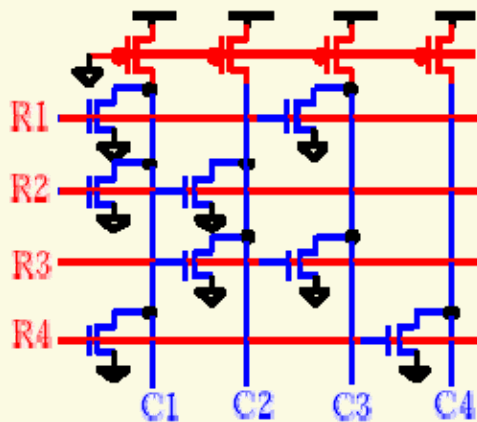
## Classification



# Mémoires de type ROM

Définition : Mémoire à accès lecture uniquement

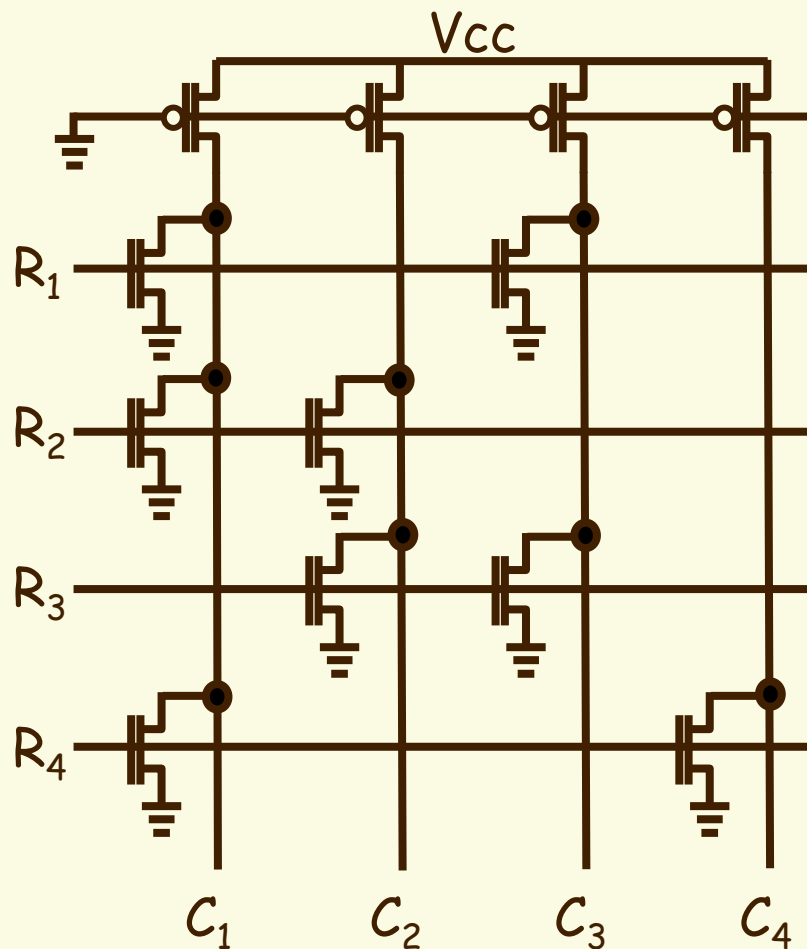
Les ROM peuvent être vues comme des simples réseaux de transistors produisant une valeur logique combinatoire en fonction de leurs entrées



ROM basées sur :  
✓ des NOR,  
✓ des NAND.



# ROM basées sur des NOR



Circuit de type **pseudo NMOS** utilisant un PMOS de charge par colonne

**Mémorisation** d'une valeur à une adresse particulière par la **présence** ou **non** d'un transistor **NMOS**

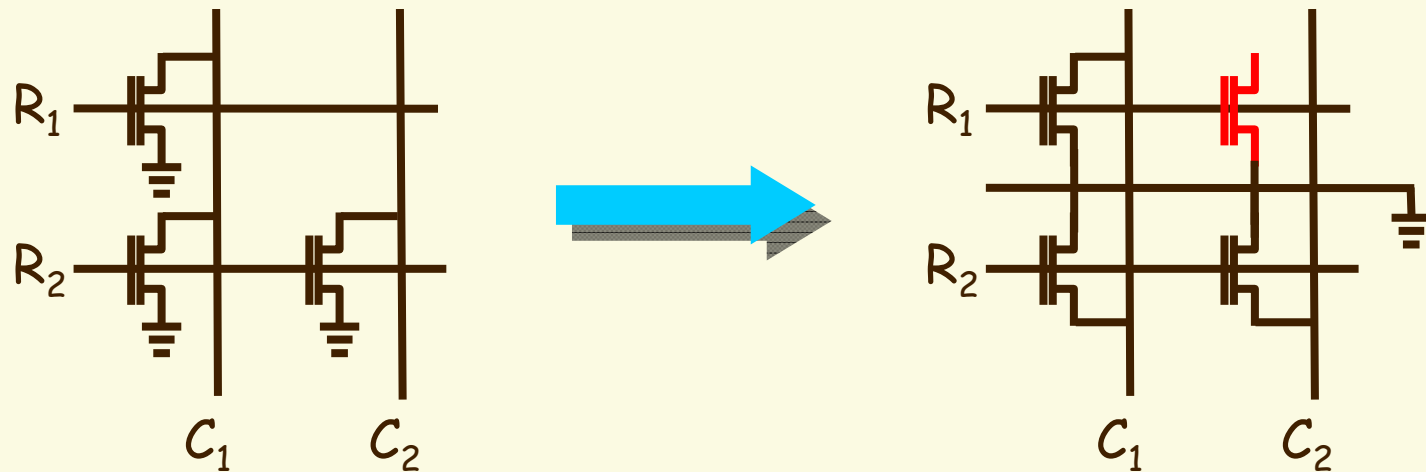
Ex :

R1	R2	R3	R4	C1	C2	C3	C4
1	0	0	0	0	1	0	1
0	1	0	0	0	0	1	1
0	0	1	0	1	0	0	1
0	0	0	1	0	1	1	0

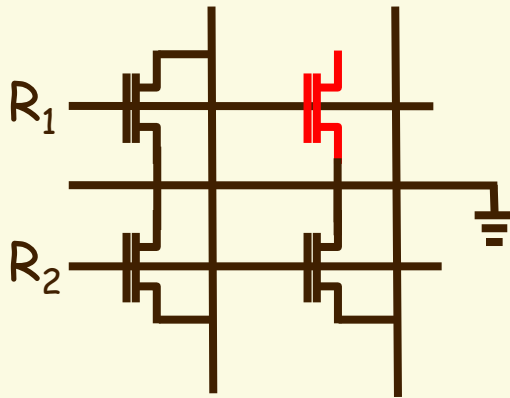
# Version Layout d'une ROM NOR

La ROM est créée à partir d'une **matrice complète** de transistor NMOS.

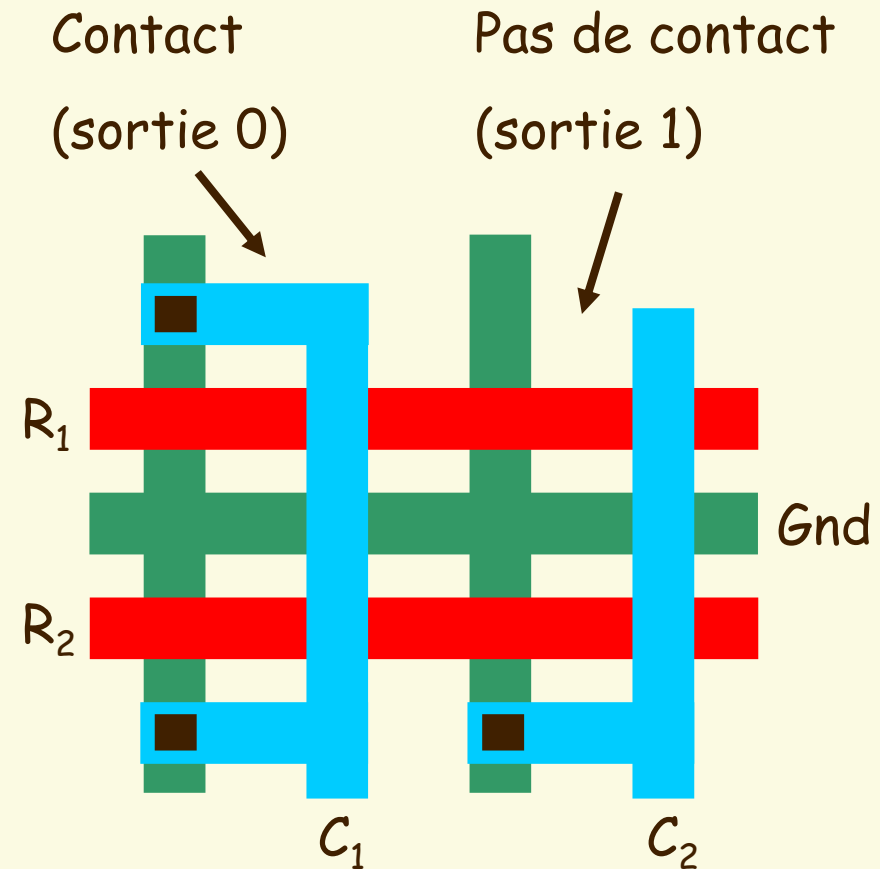
La **personnalisation** se fait au niveau de la métallisation finale par **l'ajout** ou **non** du **drain des NMOS**



# Version Layout d'une ROM NOR

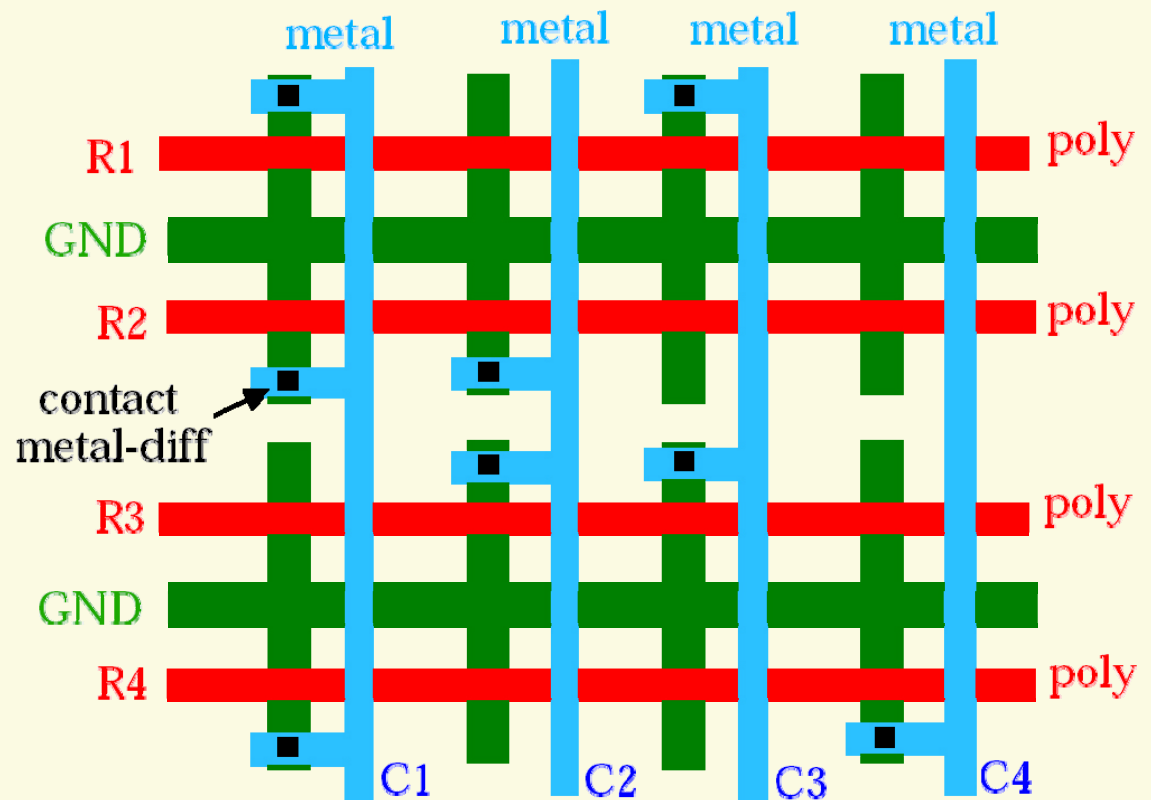
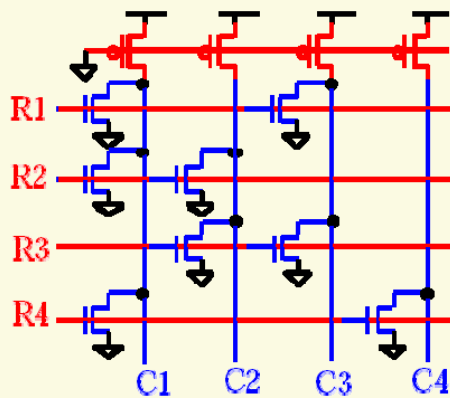


- ✓ Structure régulière,
- ✓ Facile à créer,
- ✓ Faible coût

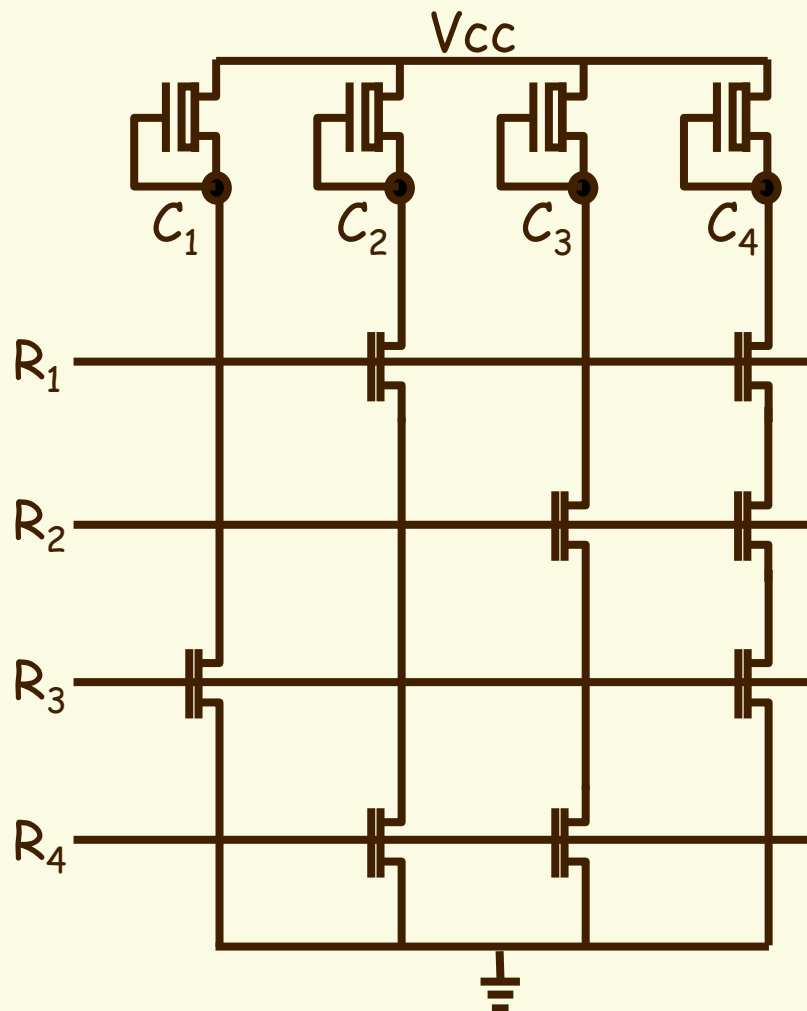


# Version Layout d'une ROM NOR

Un exemple complet :



# ROM basées sur des NAND



Circuit de type **pseudo NMOS** utilisant un **NMOS à déplétion** par colonne

**Mémorisation** d'une valeur à une adresse particulière par la **présence** ou **non** d'un transistor **NMOS**

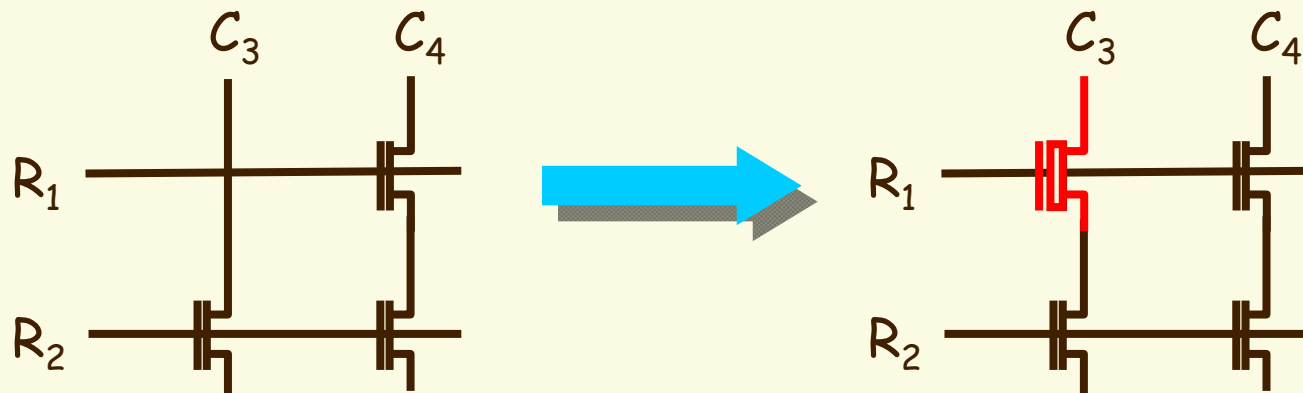
Ex :

R1	R2	R3	R4	C1	C2	C3	C4
0	1	1	1	0	1	0	1
1	0	1	1	0	0	1	1
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	0

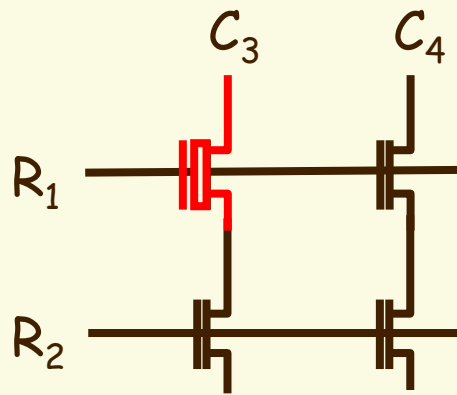
# Version Layout d'une ROM NAND

La ROM est créée à partir d'une **matrice complète** de transistor NMOS.

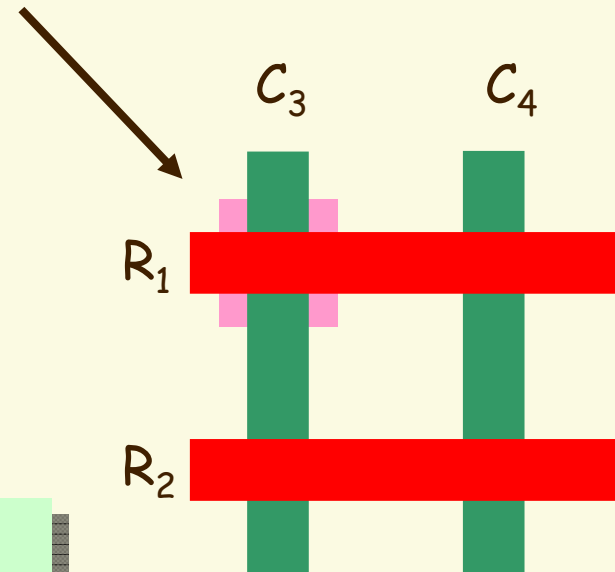
La **personnalisation** se fait au niveau de l'**ajout ou non** d'une **couche enterrée de conduction** entre drain et source des transistors NMOS



# Version Layout d'une ROM NAND



Couche enterrée garantissant la conduction du NMOS

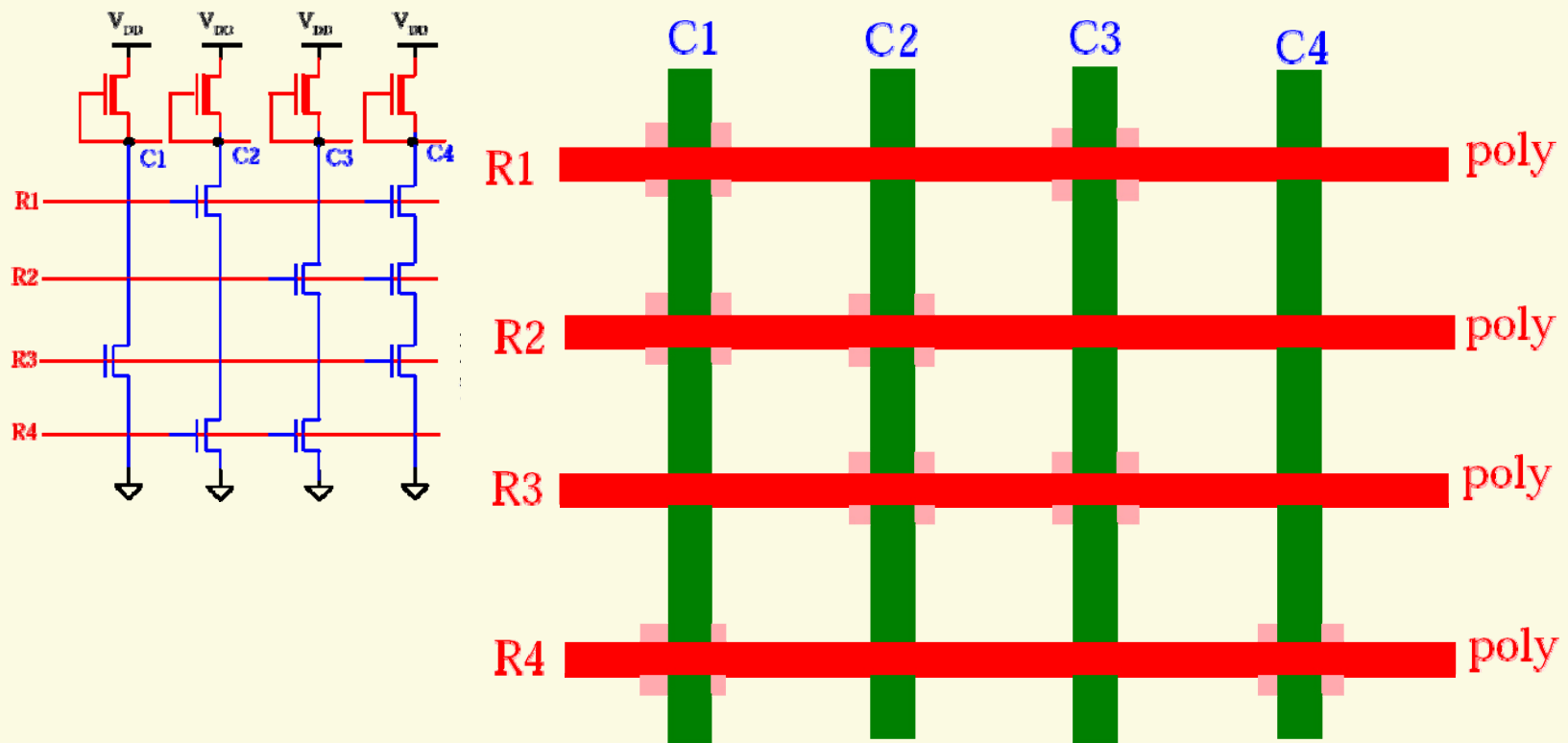


- ✓ Structure régulière très simple,
- ✓ Facile à créer,
- ✓ Faible coût



# Version Layout d'une ROM NAND

Un exemple complet :



# Décodeurs de lignes et de colonnes

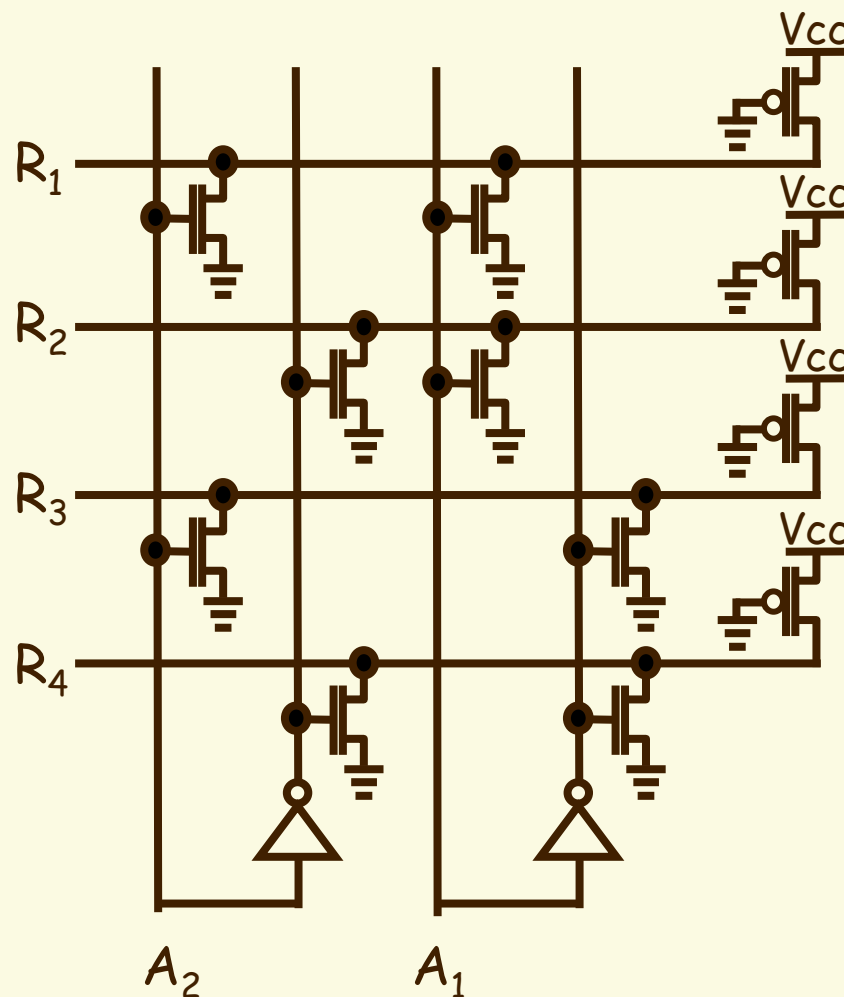
Les **décodeurs** de ligne et de colonne ont pour objectif de **sélectionner** une des cellules de la mémoire et de **lire** son contenu.

Un décodeur de lignes à **N entrées** pour une **ROM** de type **NOR** sélectionne donc une des  **$2^N$  lignes** en mettant à  $V_{cc}$  la ligne désirée



$A_1$	$A_2$	$R_1$	$R_2$	$R_3$	$R_4$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

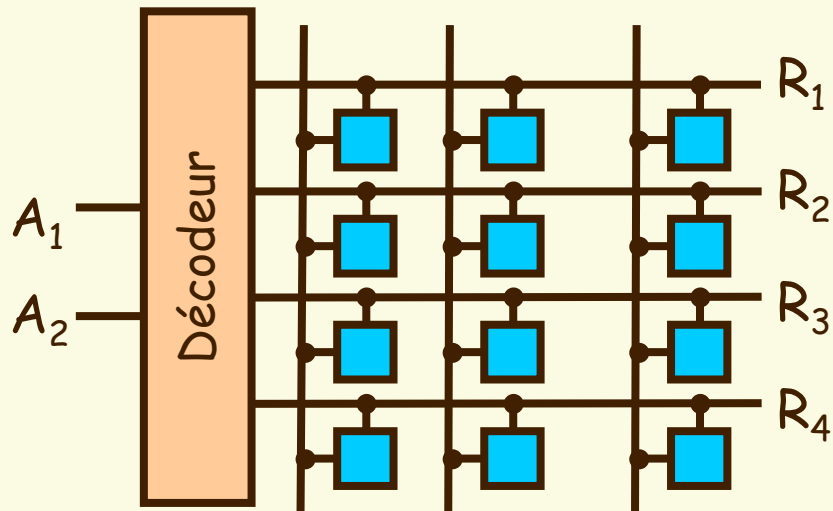
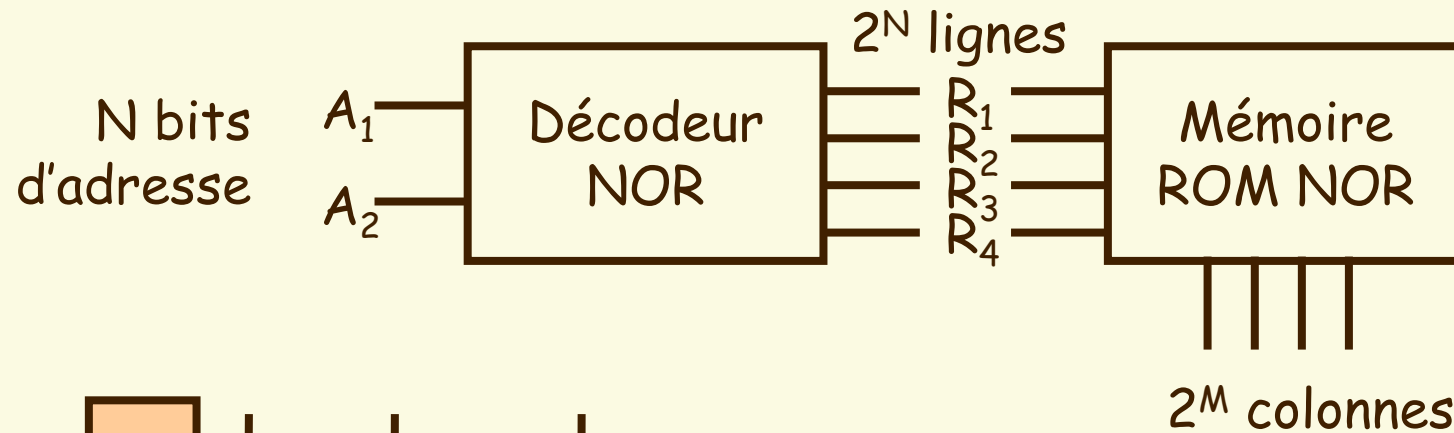
# Décodeurs de lignes



Les **décodeurs** de ligne NOR peuvent être réalisés de manière similaire aux ROM.

$A_1$	$A_2$	$R_1$	$R_2$	$R_3$	$R_4$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

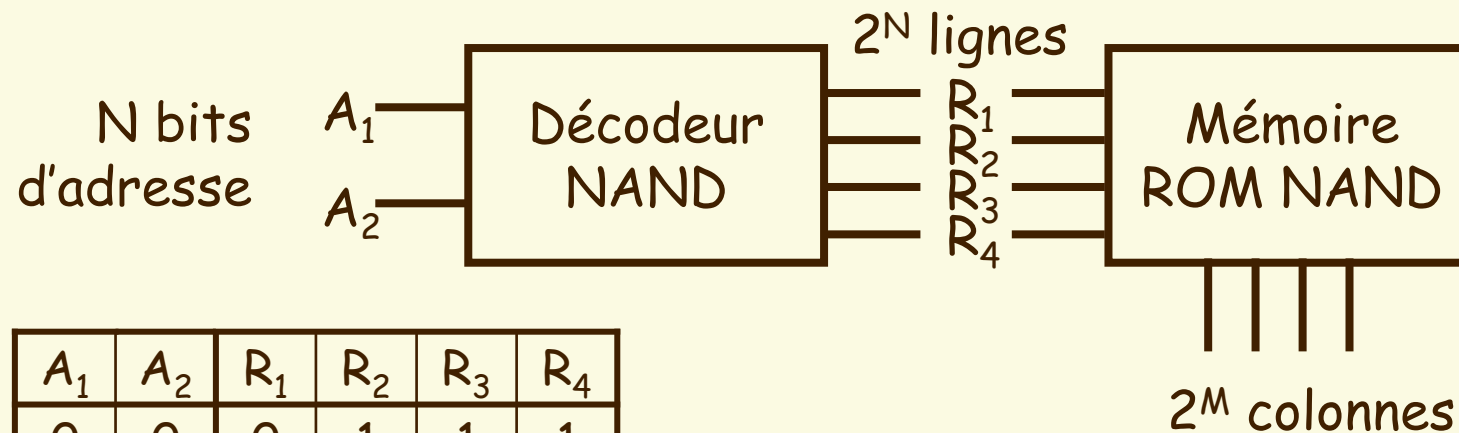
# Décodeurs de lignes



Structure régulière  
très simple à intégrer

# Décodeurs de lignes

Possibilité de réaliser la **ROM** et son **décodeur** de lignes en **NAND**



$A_1$	$A_2$	$R_1$	$R_2$	$R_3$	$R_4$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

**Sélection** d'une ligne  
par mise à 0 de  $R_i$

# Décodeurs de colonnes

Le **décodeur** de colonne a pour objectif de **sélectionner** une des colonnes de la mémoire et de **router** son contenu vers la sortie.

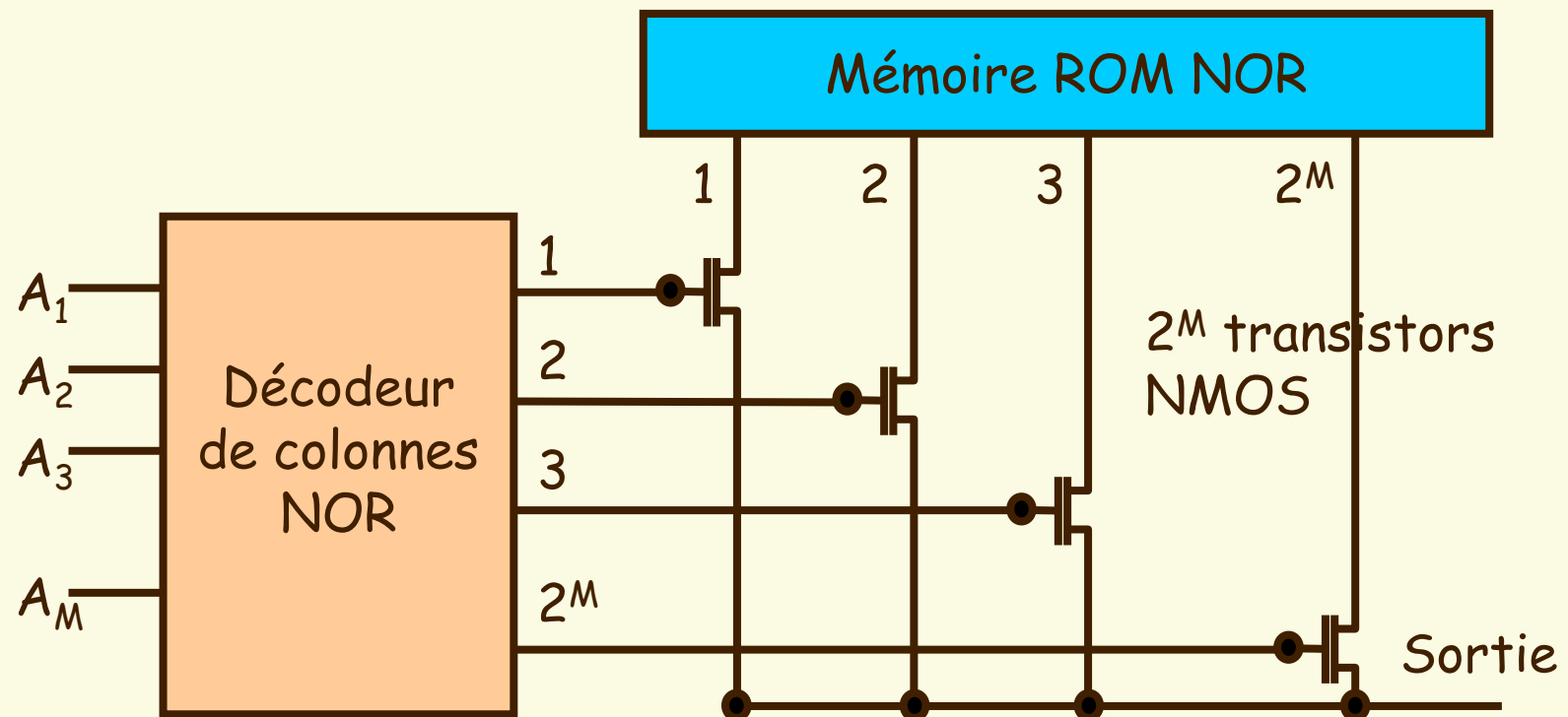
Un décodeur de colonnes à **M entrées** pour une **ROM** de type **NOR** sélectionne donc une des  **$2^M$  colonnes** en mettant à Vcc la colonne désirée



Il peut être réalisé de **manière similaire** au décodeur de lignes (NOR ou NAND)

# Décodeurs de colonnes

Le **décodeur** de colonne a également pour objectif de **router** le contenu d'une cellule vers la sortie.





## Une autre solution

L'inconvénient majeur du précédent décodeur est son grand nombre de transistors :  $2^M (M+1)$

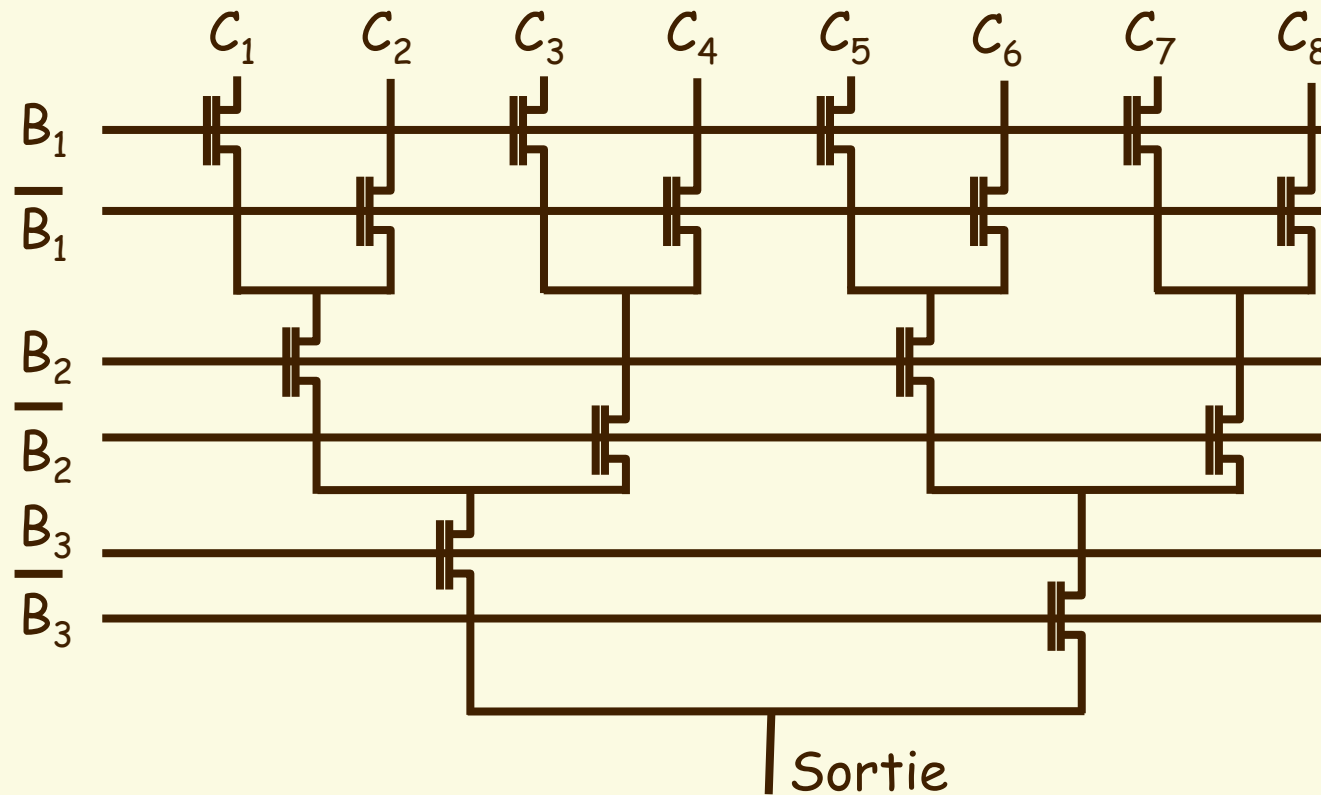
Il est possible de réaliser le décodeur et le routeur de données en une seule fois sous la forme d'un arbre binaire car les tailles de mémoires sont des puissances de 2

Avantage : réduit le nombre de transistors

Inconvénients : plus de transistors à traverser pour router la donnée résultat

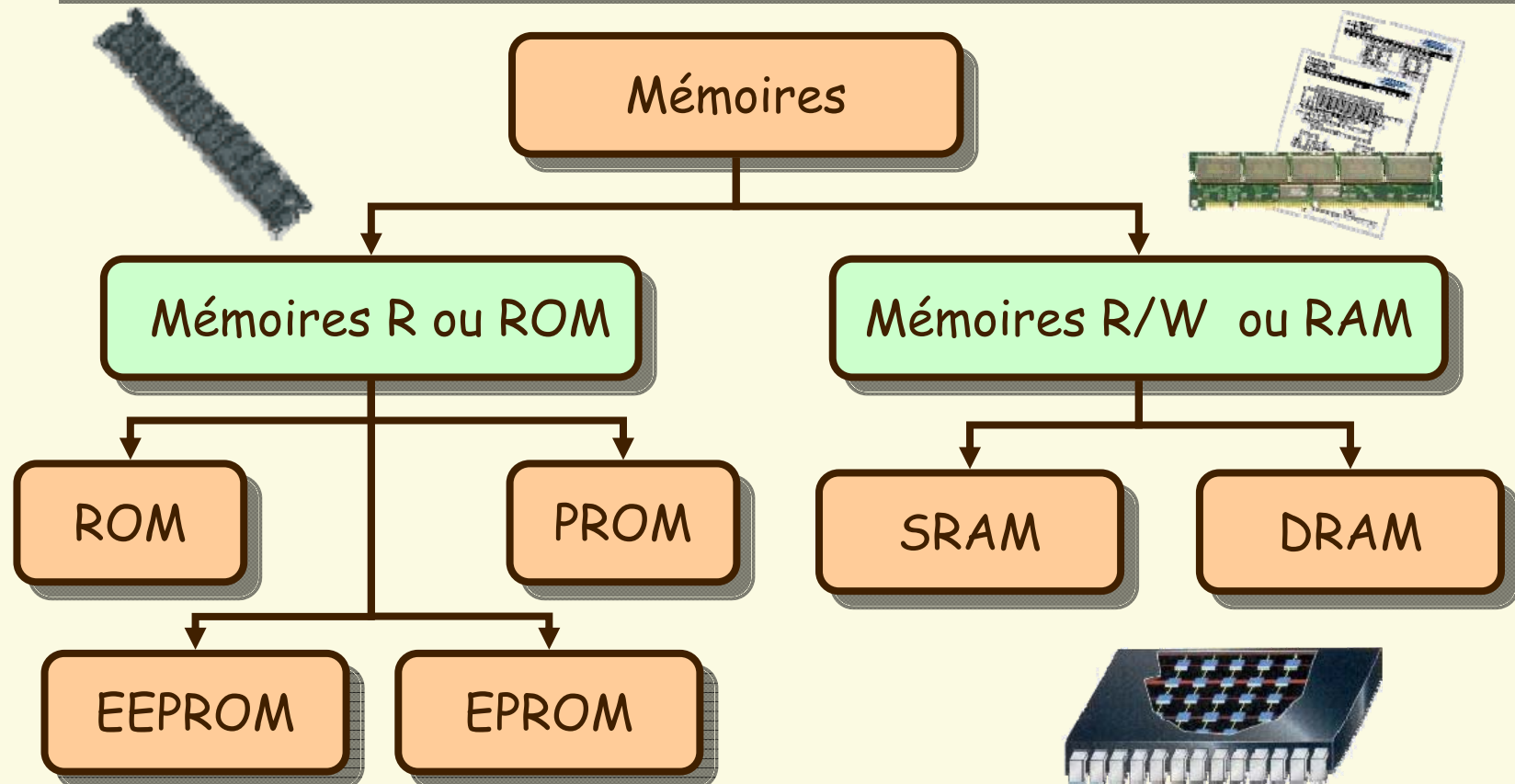
# Une autre solution

Un exemple avec  $M = 3$



# Éléments de mémorisation

## Classification



# Mémoires de type PROM

Définition : Mémoire **ROM** dont le **contenu** est **programmable**

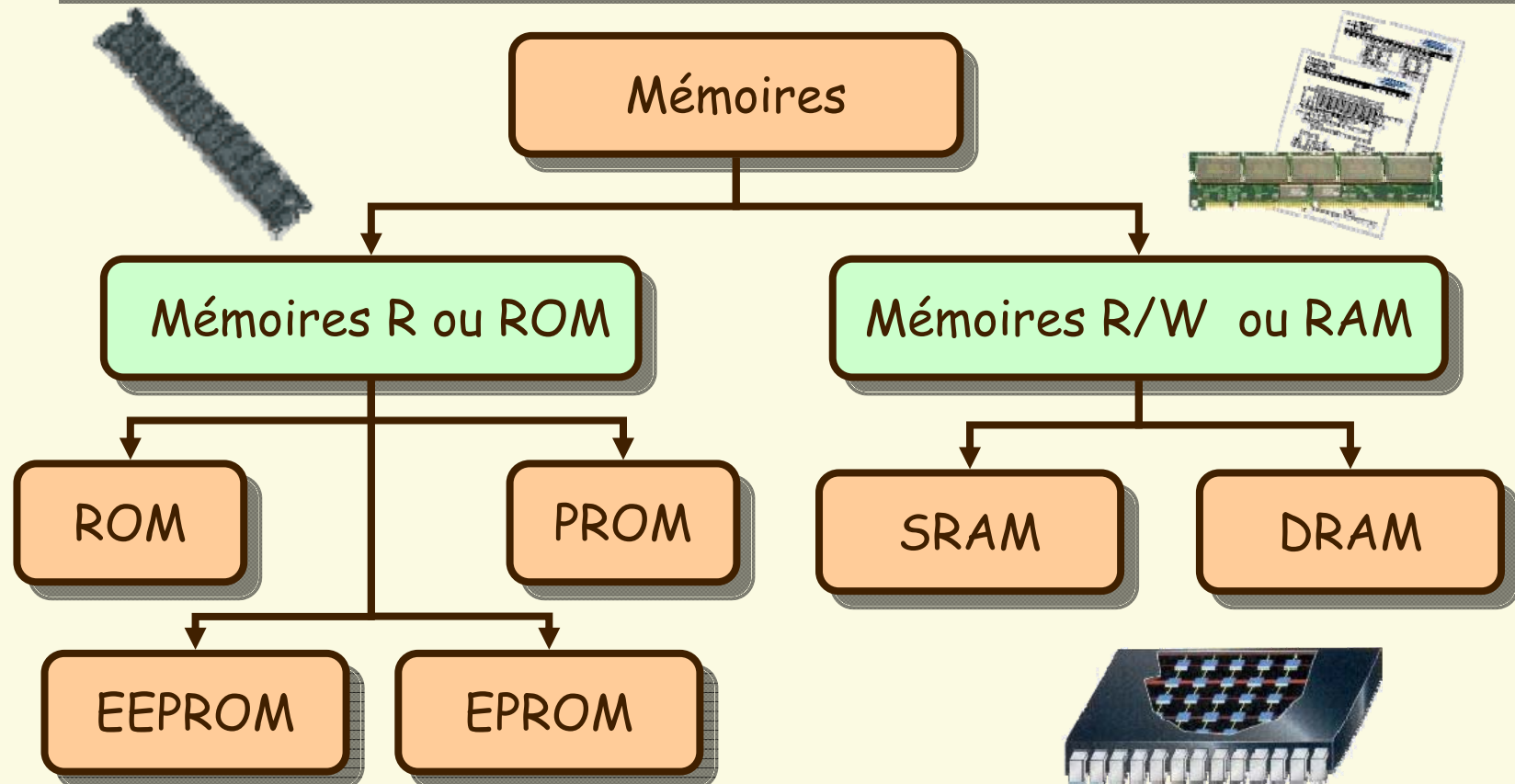
Chaque transistor de la PROM est accompagné d'un **fusible** **monté en série**.

Principe : Une impulsion de programmation claque le fusible ( $\approx 17,5$  à  $20V$  pendant  $0,6$  à  $1,2\mu s$ )

La **programmation** d'une PROM est **irréversible**

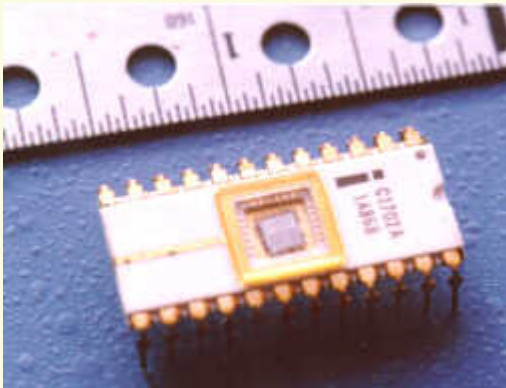
# Éléments de mémorisation

## Classification



# Mémoires de type EPROM

Définition : Mémoire **ROM** dont le contenu est **programmable plusieurs fois** (E = Erasable)

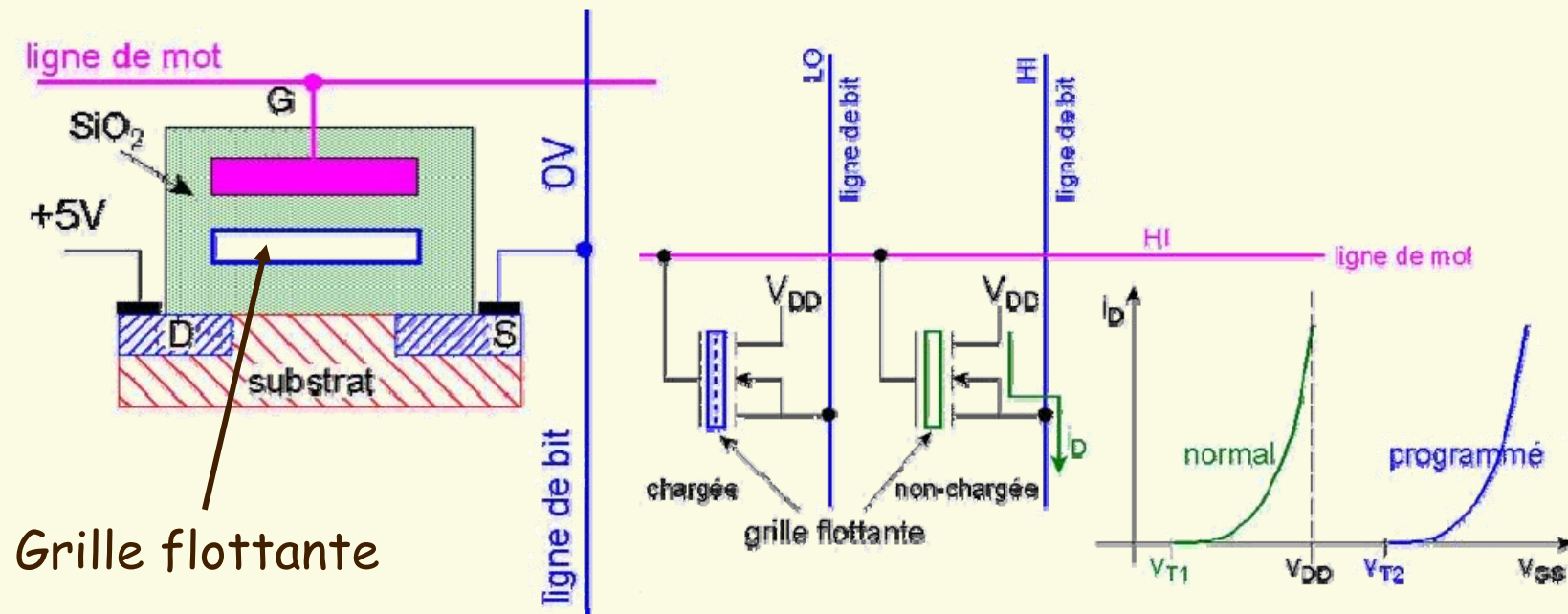


Une EPROM est une PROM **effaçable** par un **rayonnement ultraviolet intense**

L'effacement se fait en plaçant l'EPROM sous des UV durant un temps compris entre 5 et 30 mn.

# Mémoires de type EPROM

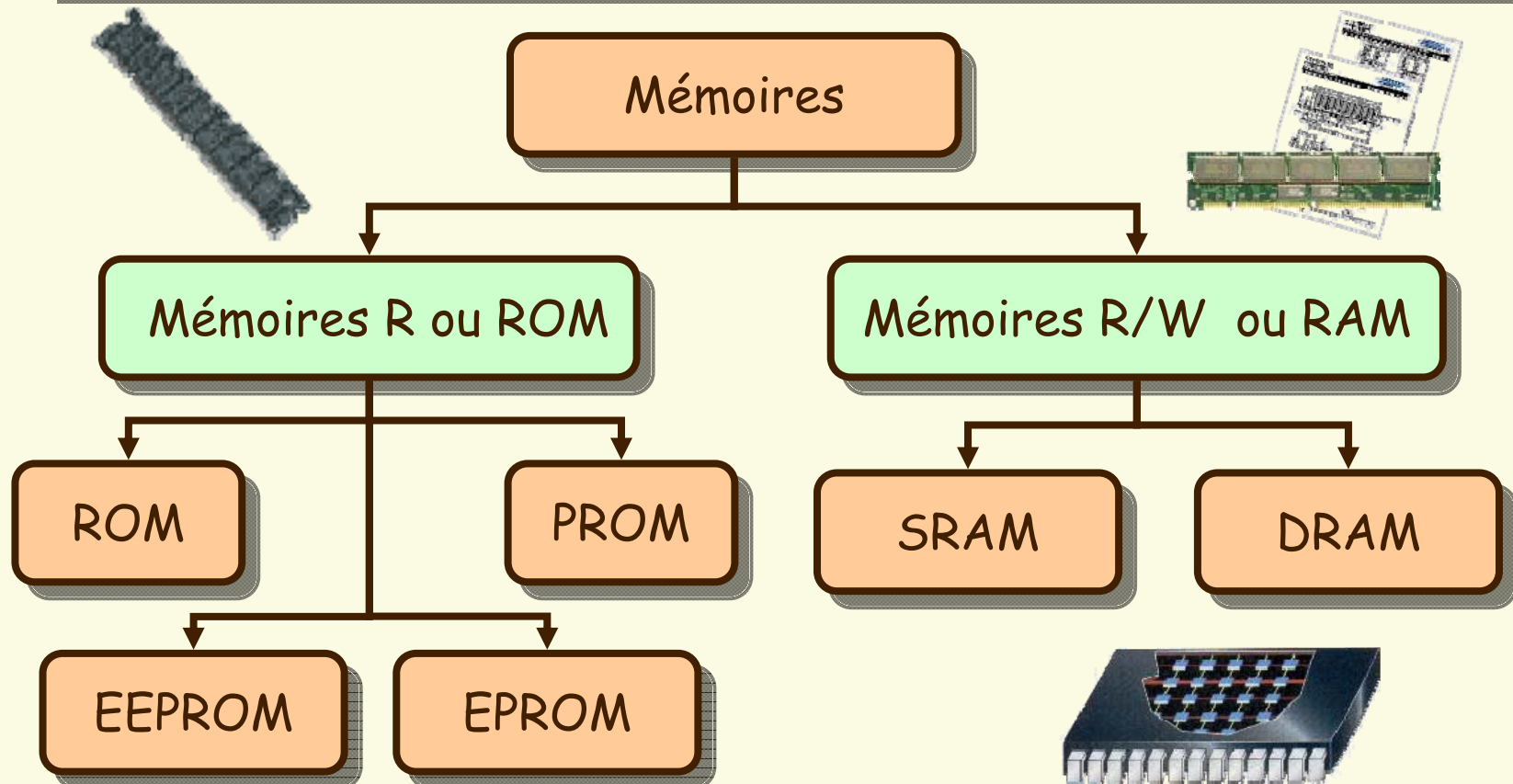
Les transistors utilisés sont réalisés en **technologie FAMOS** (Floating gate Avalanche injection MOS) : la grille n'est reliée à aucun conducteur





# Éléments de mémorisation

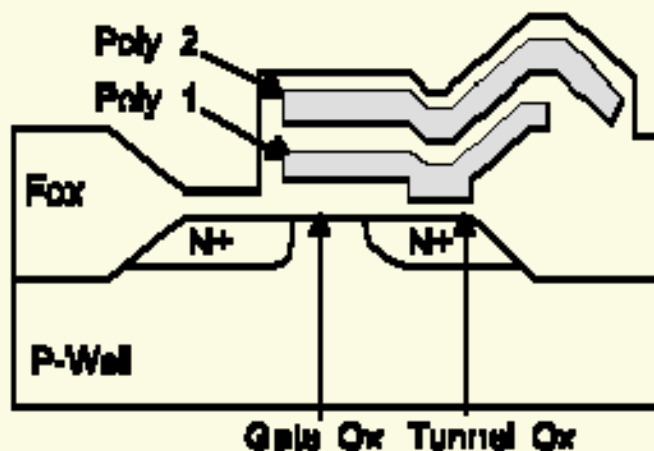
## Classification



# Mémoires de type EEPROM

Définition : Mémoire **ROM** dont le contenu est **programmable** plusieurs fois

Une EEPROM est une PROM **effaçable** de manière **électrique** (EE : Electric Erasable)

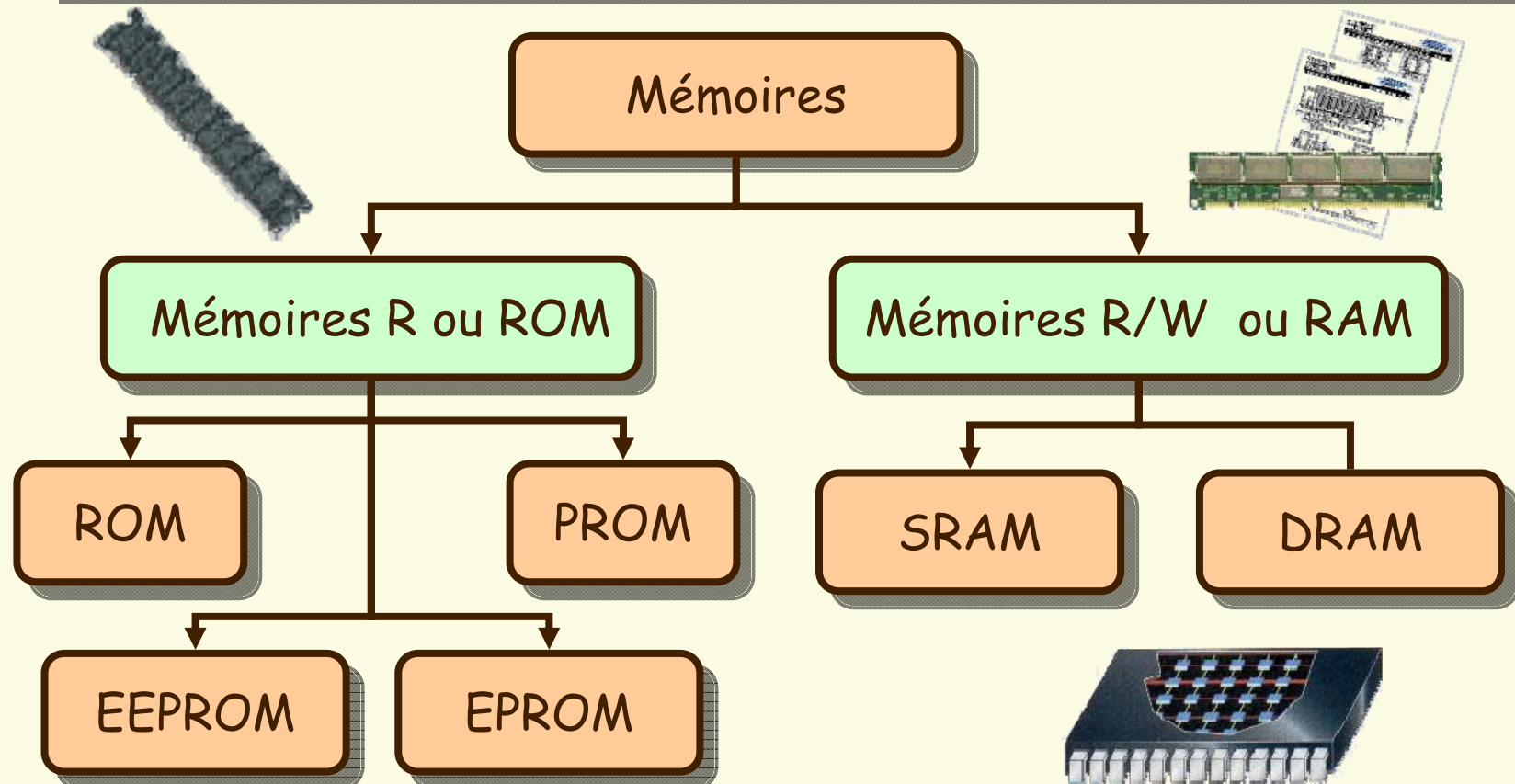


Une **impulsion** de programmation charge la grille flottante (Techno **FLOTOX**) (20V pendant 20ms).

**L'effacement** se fait en appliquant pendant 10 ms une tension de 20V.

# Éléments de mémorisation

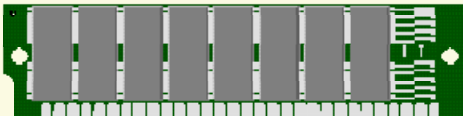
## Classification



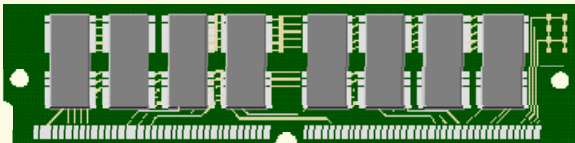
# Mémoires de type RAM

Définition : Mémoire à accès lecture/écriture

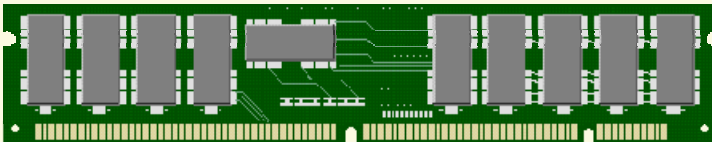
30-pin SIMM (3.5 x .75")



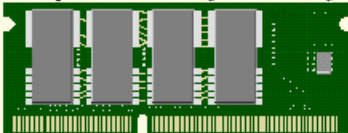
72-pin SIMM (4.25 x 1")



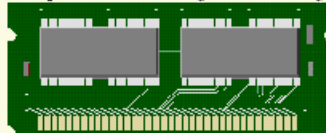
168-pin DIMM (5.375 x 1")



144-pin SODIMM (2.625 x 1")



72-pin SODIMM (2.875 x 1")



A la différence des ROM qui conservent indéfiniment les données, les mémoires RAM nécessitent d'être alimentées pour produire une valeur en sortie

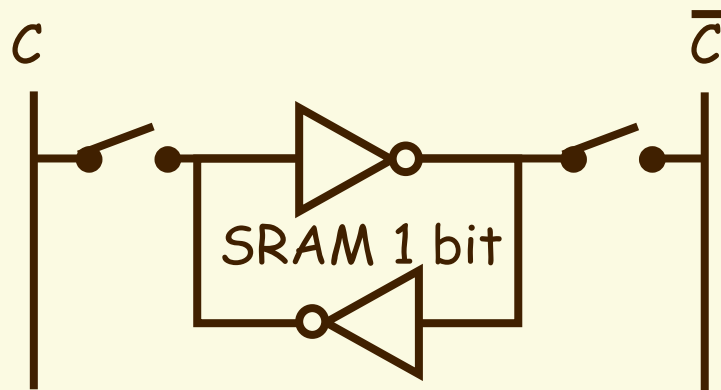
2 grandes familles :

- ✓ SRAM,
- ✓ DRAM

# Mémoires de type SRAM

Définition : Mémoire RAM de type statique

Par **statique**, on entend une **mémoire** dont le **contenu** est **conservé** tant que **l'alimentation électrique** est assurée

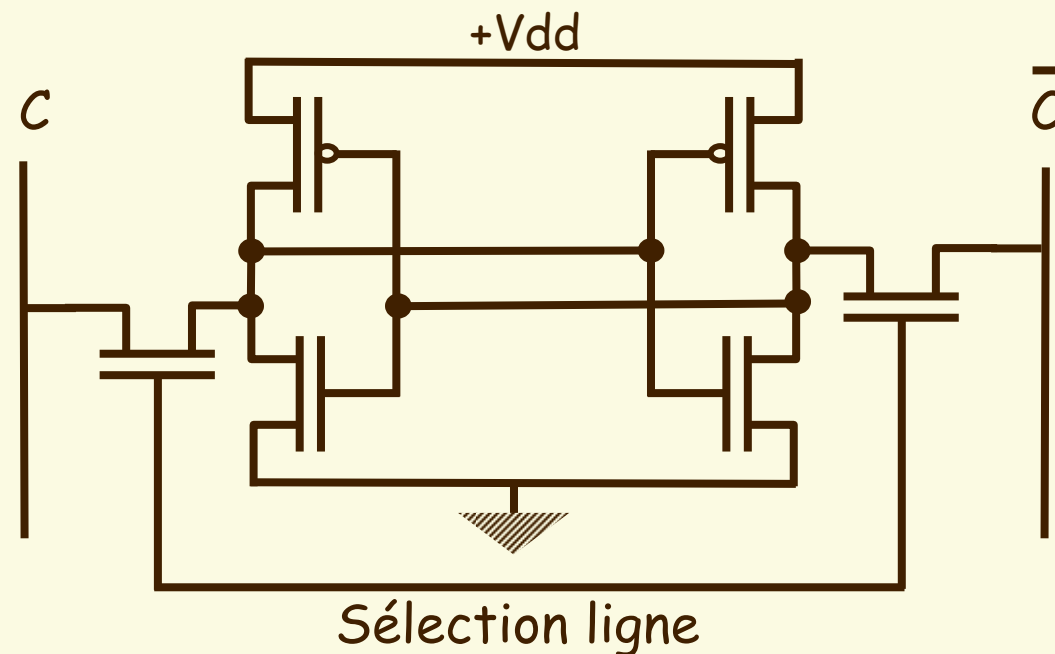


La **mémorisation** d'une cellule 1 bit est assurée par un **système bistable**

La **commande** de la mémoire est assurée par **2 interrupteurs**

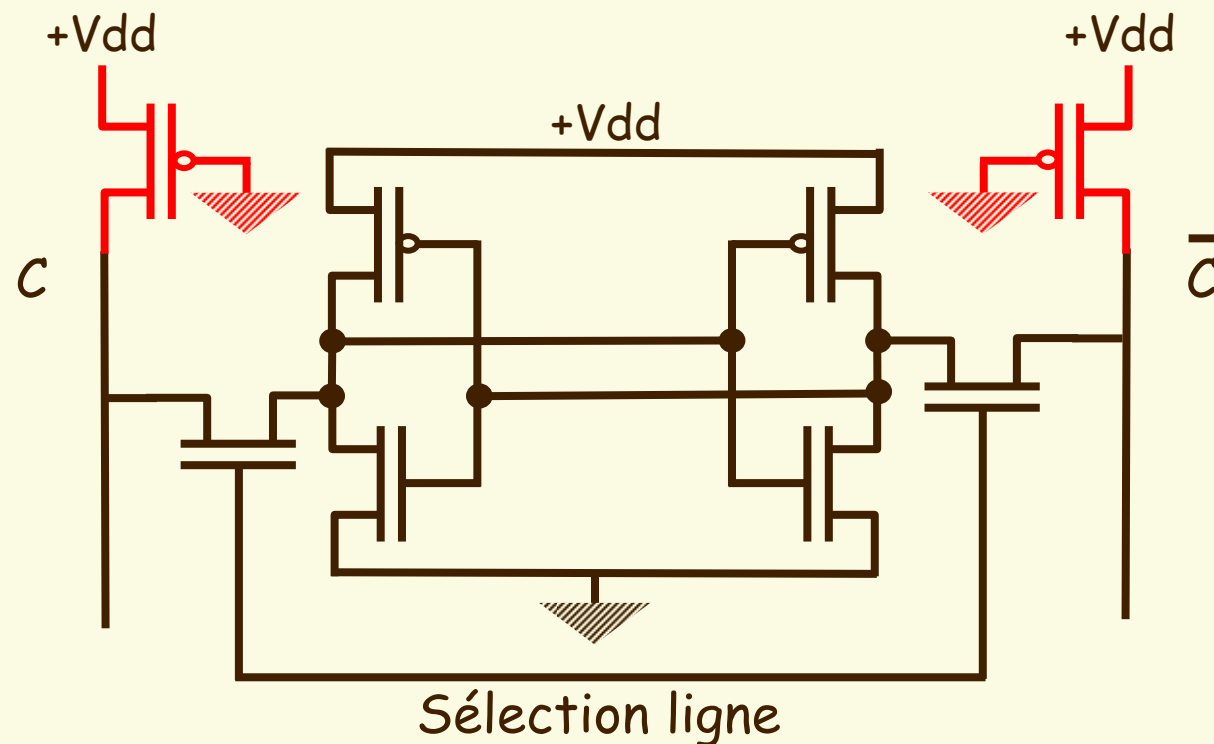
# Mémoires de type SRAM CMOS

Une cellule 1 bit de **SRAM CMOS** comprend donc **6 transistors** (4 NMOS + 2 PMOS)

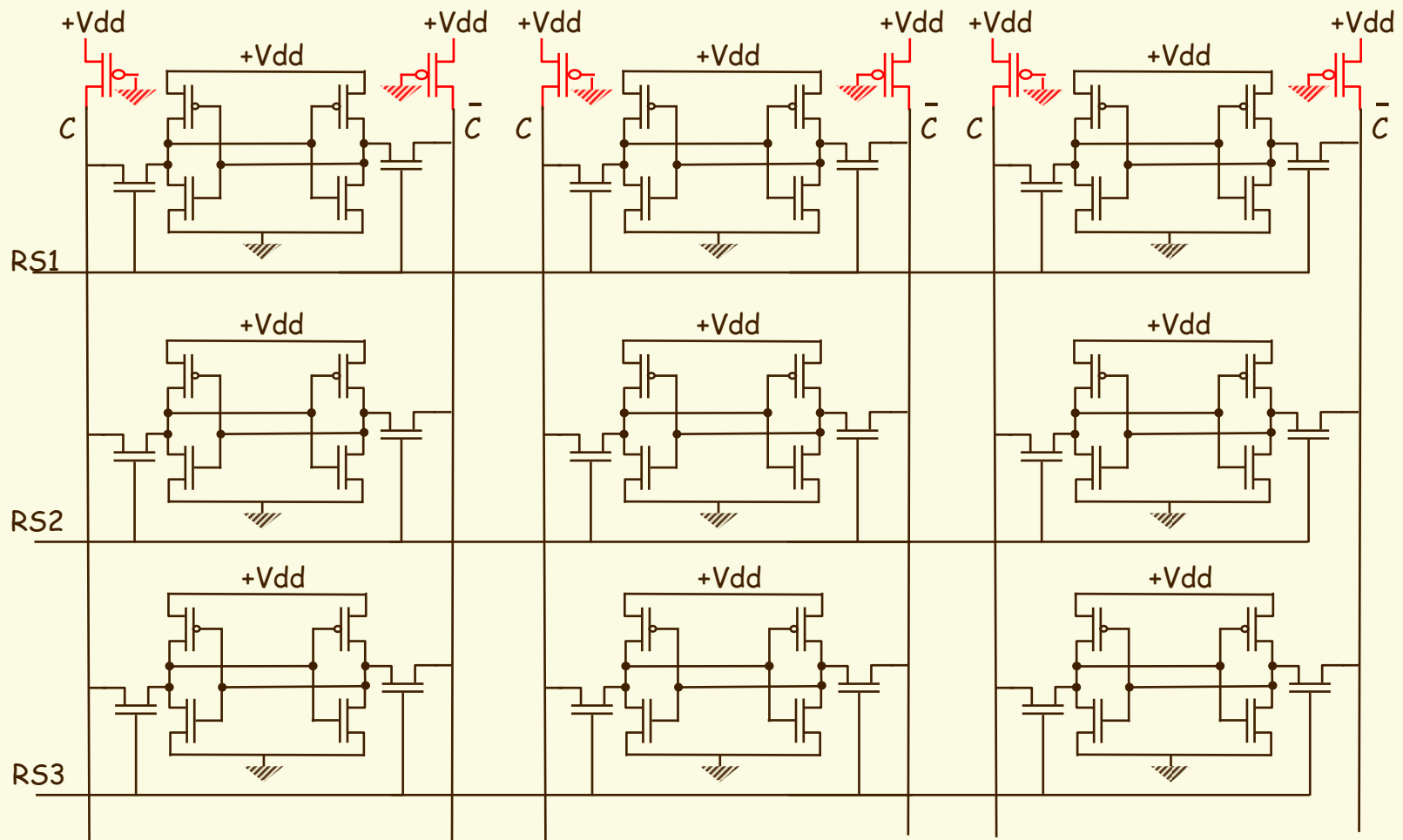


# Mémoires de type SRAM CMOS

En réalité, on **rajoute** pour **chaque colonne** de la mémoire 2 transistors PMOS de **précharge à 1**



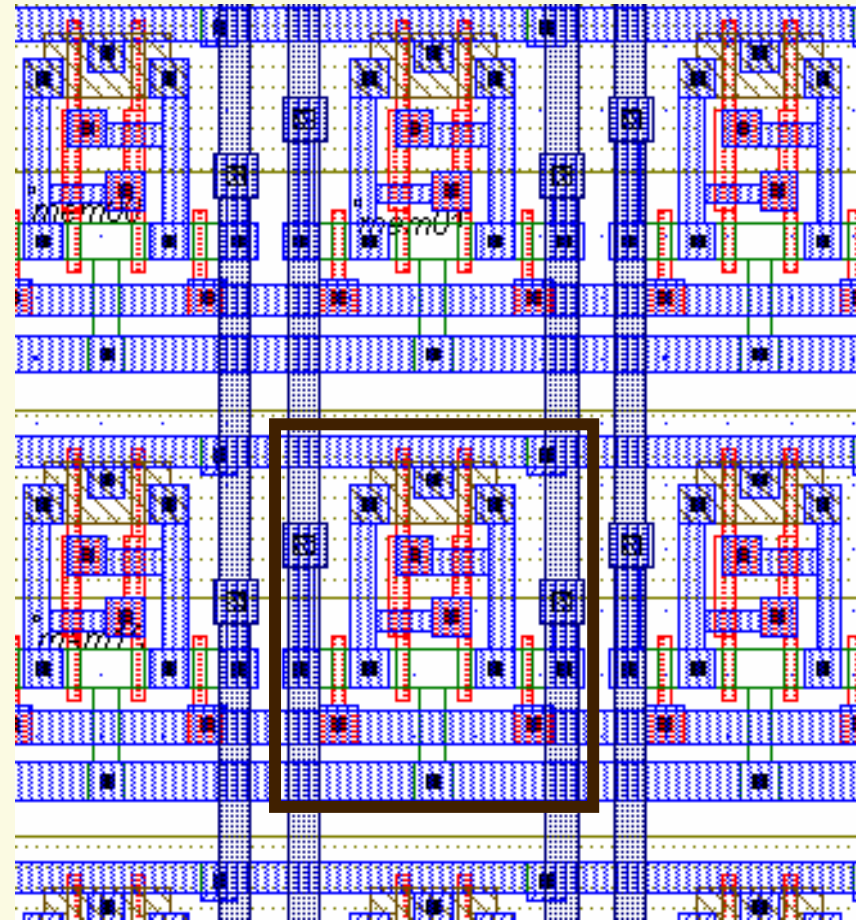
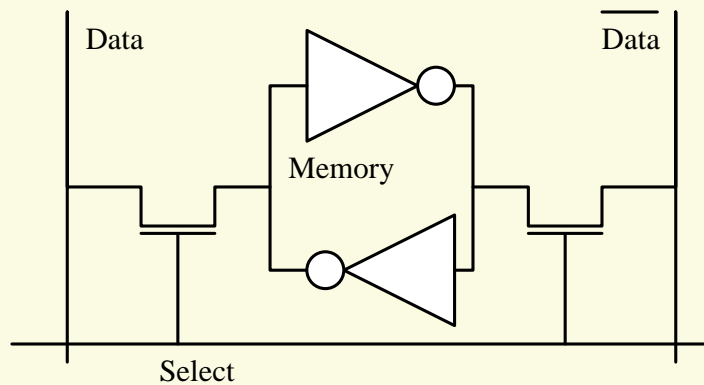
# Mémoires de type SRAM CMOS





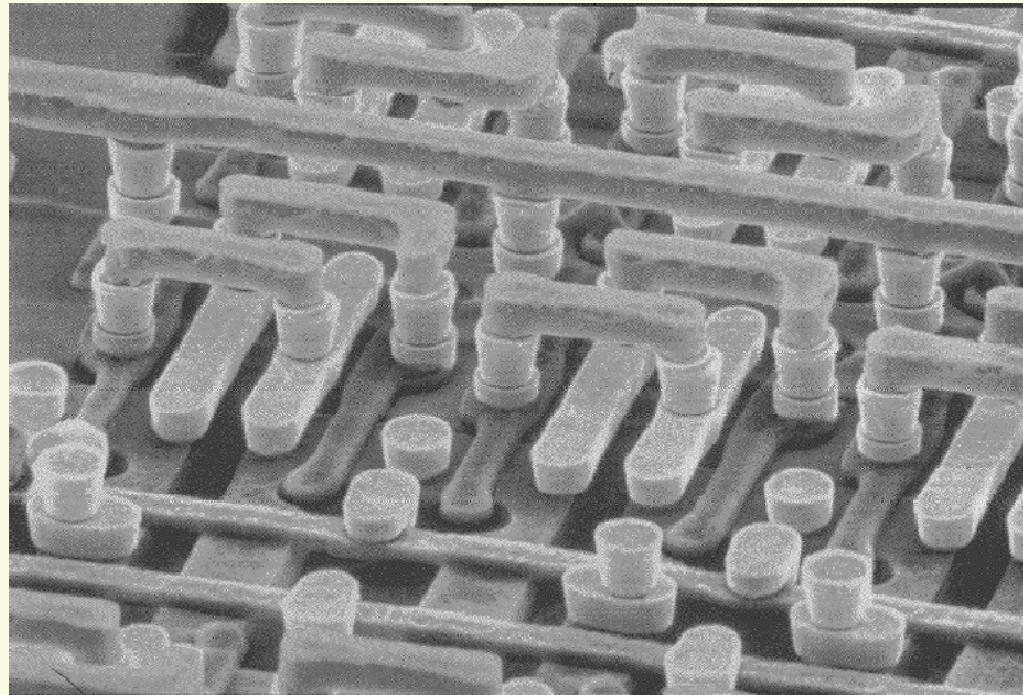
# Mémoires de type SRAM CMOS

Version layout



# Mémoires de type SRAM CMOS

Zoom



Exemple de la structure tridimensionnelle réalisée  
photo SEM du câblage d'une mémoire statique après attaque chimique de l'oxyde

# Mémoires de type SRAM CMOS

Si aucune ligne n'est sélectionnée, les 2 transistors M3 et M4 sont bloqués : les cellules sont en phase de maintien

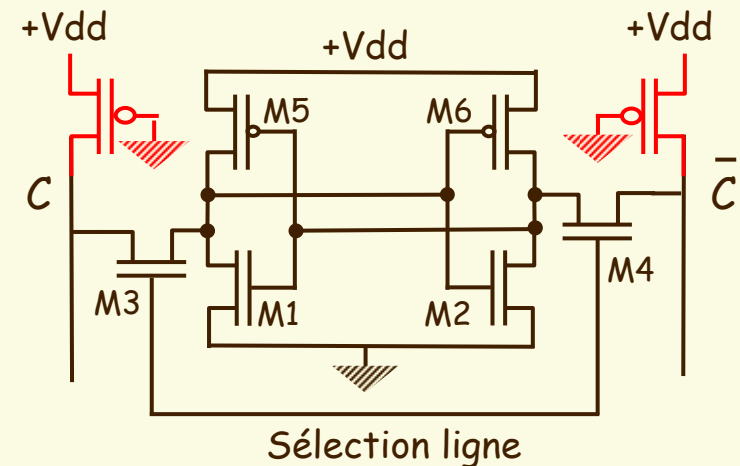
Si aucune ligne n'est sélectionnée, les 2 transistors de précharge forcent donc C et  $\bar{C}$  à 5 V (niveau logique 1)

En réalité, on peut écrire :

$$V_{\text{NOT-C}} = V_C = V_{\text{DD}} - (V_{\text{T0}} + \gamma(\sqrt{|2\phi_F| + V_C} - \sqrt{|2\phi_F|}))$$

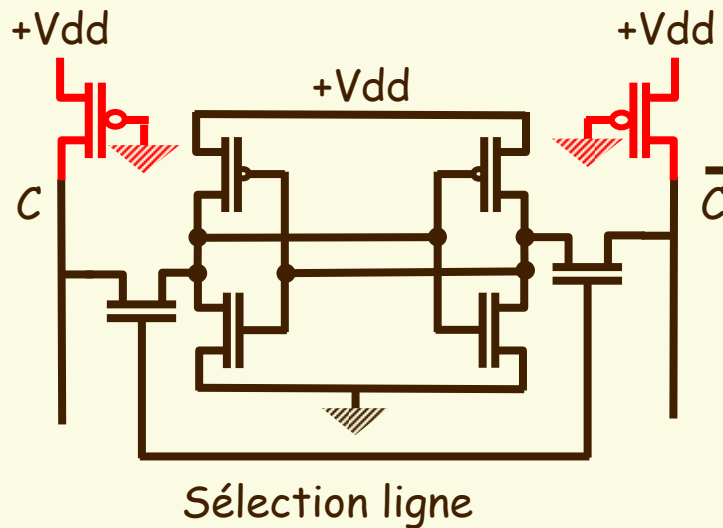
$$V_C = V_{\text{NOT-C}} = 3.5 \text{ V}$$

$$\text{pour } V_{\text{DD}} = 5 \text{ V}, V_{\text{T0}} = 1 \text{ V}, |2\phi_F| = 0.6 \text{ V}, \gamma = 0.4 \text{ V}^{1/2}.$$



# Mémoires de type SRAM CMOS

## 6 cas de fonctionnement :



Sélection ligne non validée :

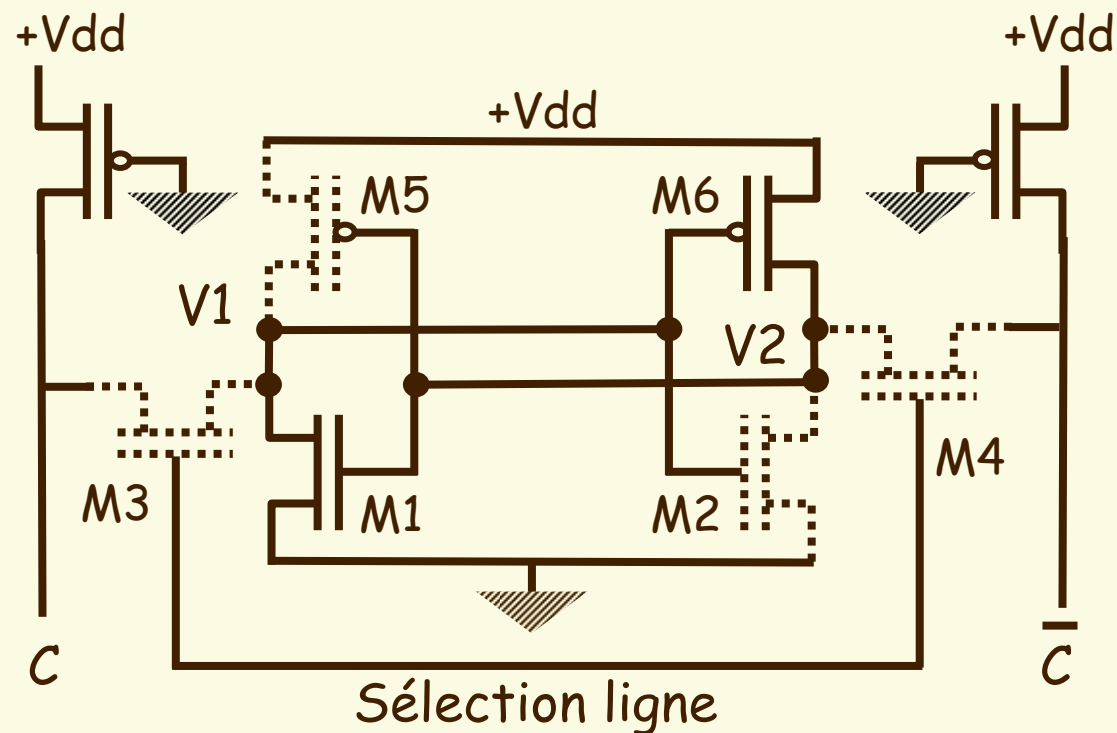
- ✓ Maintien d'un 0
- ✓ Maintien d'un 1

## Sélection ligne validée

- ✓ Lecture 0
- ✓ Lecture 1
- ✓ Ecriture 0
- ✓ Ecriture 1

# Maintien d'un 0

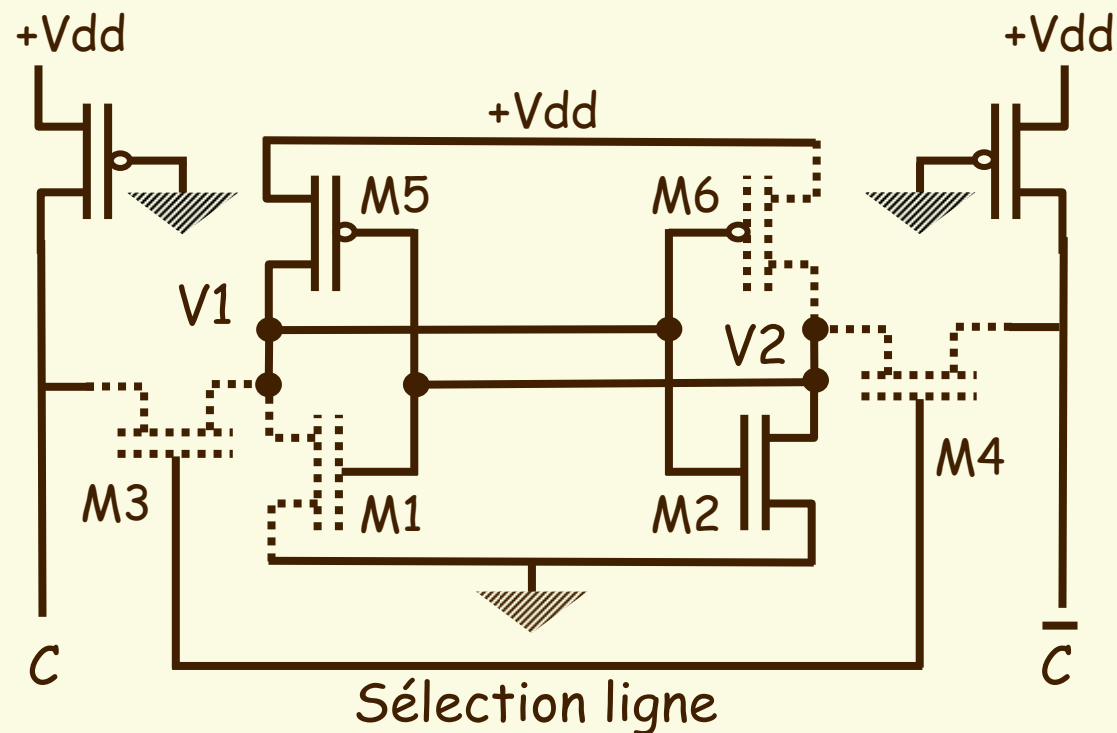
**M3, M4 non conducteurs** car Sélection = 0  
Le système bistable maintient la valeur 0



M2 bloqué  
M6 passant  
M5 bloqué  
M1 passant  
  
 $V2 = 1, V1 = 0$

# Maintien d'un 1

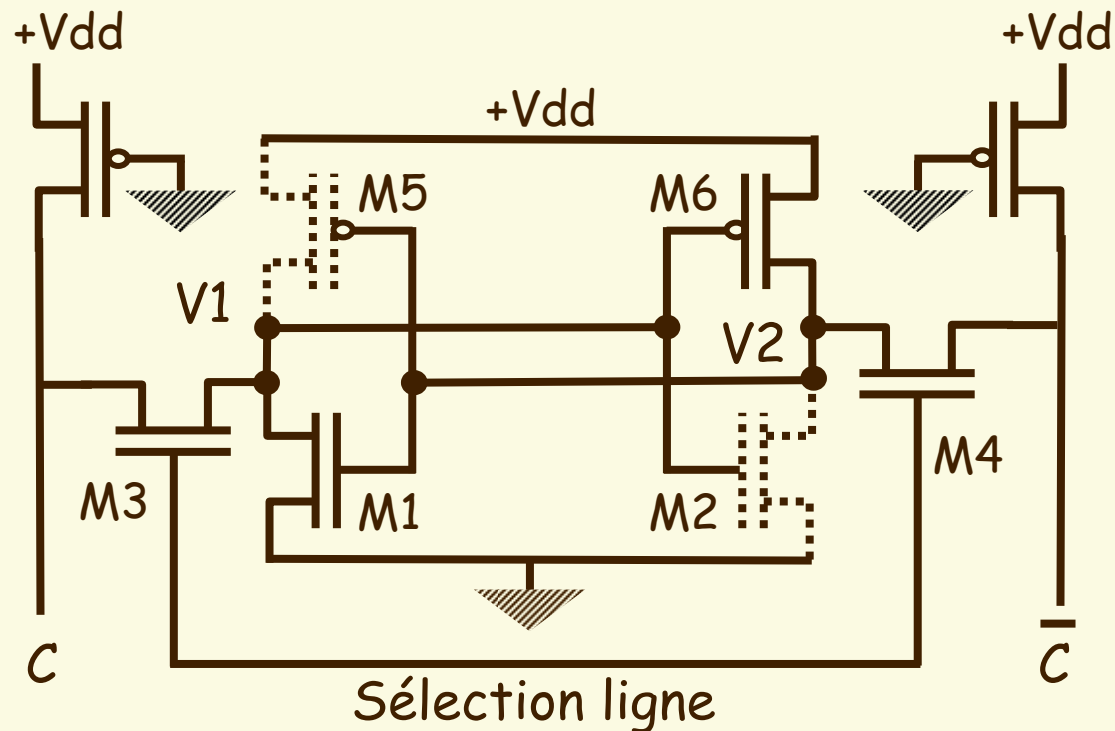
**M3, M4 non conducteurs** car Sélection = 0  
Le système bistable maintient la valeur 1



M6 bloqué  
M2 passant  
M1 bloqué  
M5 passant  
  
 $V2 = 0, V1 = 1$

# Lecture d'un 0

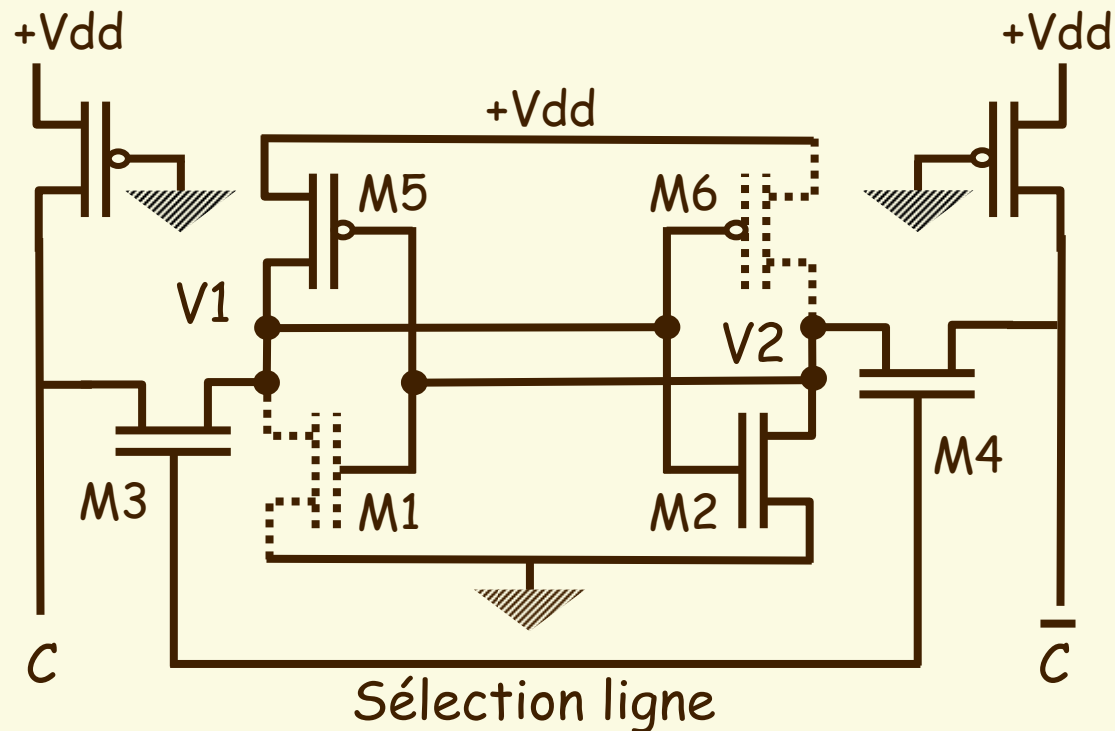
**M3, M4 conducteurs** car Sélection = 1  
On garde  $\overline{C} = 1$  et on obtient  $C = 0$



$\overline{C} = 1 = V2$  donc :  
M1 passant  
M5 bloqué  
D'où  $V1 = 0 = C$   
M6 passant  
M2 bloqué

# Lecture d'un 1

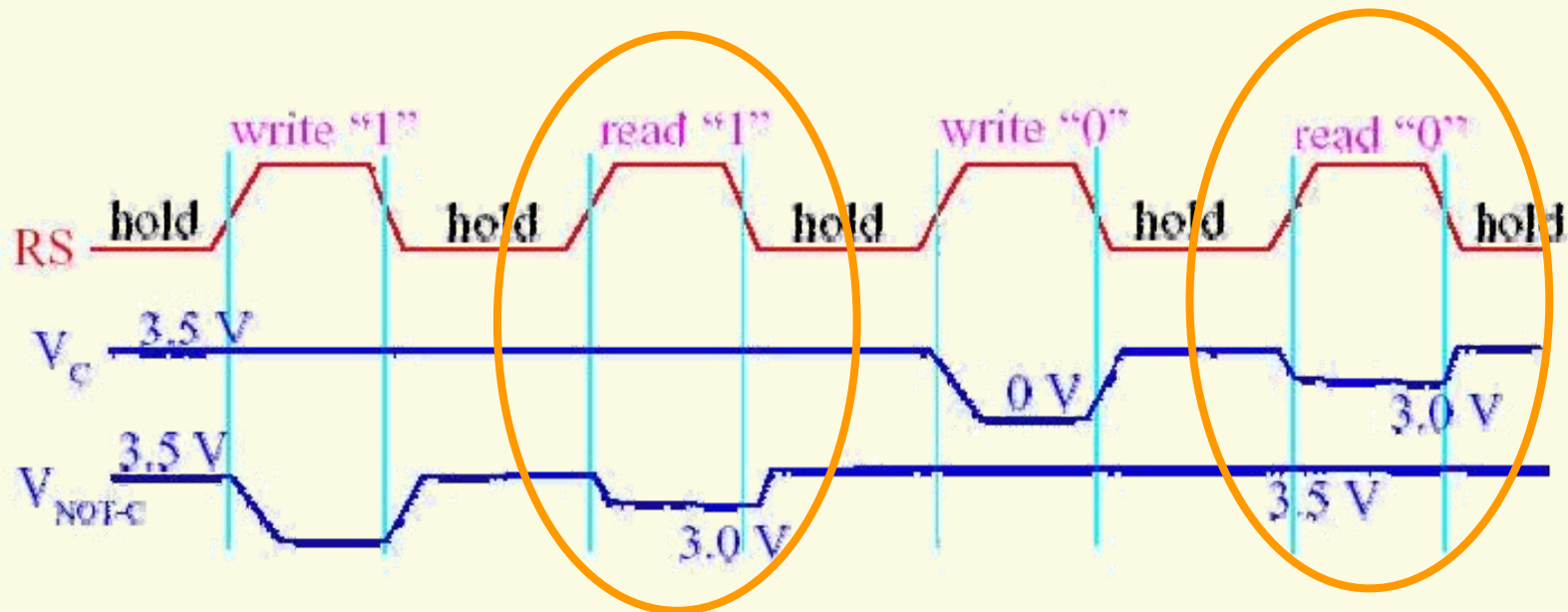
**M3, M4 conducteurs** car Sélection = 1  
On garde  $C = 1$  et on obtient  $\overline{C} = 0$



$C = 1 = V1$  donc :  
M2 passant  
M6 bloqué  
D'où  $V2 = 0 = \overline{C}$   
M1 passant  
M5 bloqué



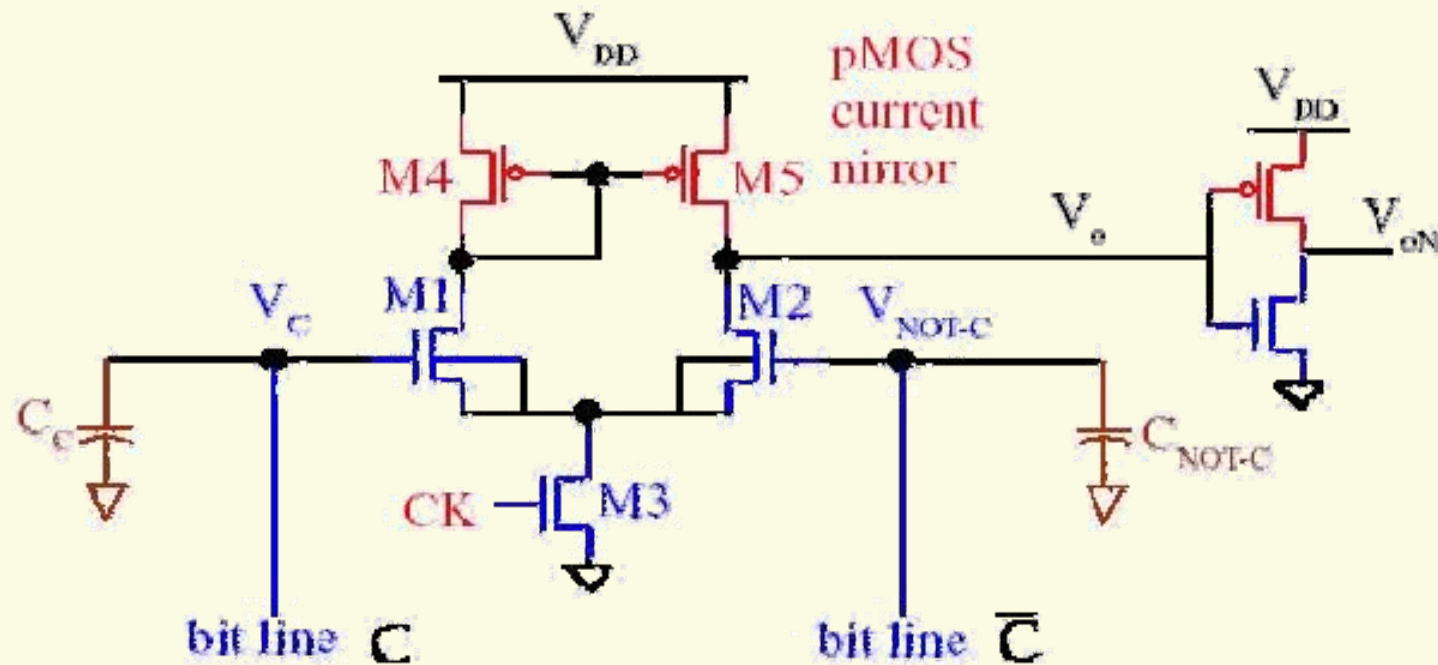
# Chronogramme typique



La mise à 0 de C ou de  $\bar{C}$  par la cellule entraîne seulement une faible chute de potentiel.

C'est cette différence de potentiel qui va être ensuite amplifiée et donner en sortie un 0 ou un 1

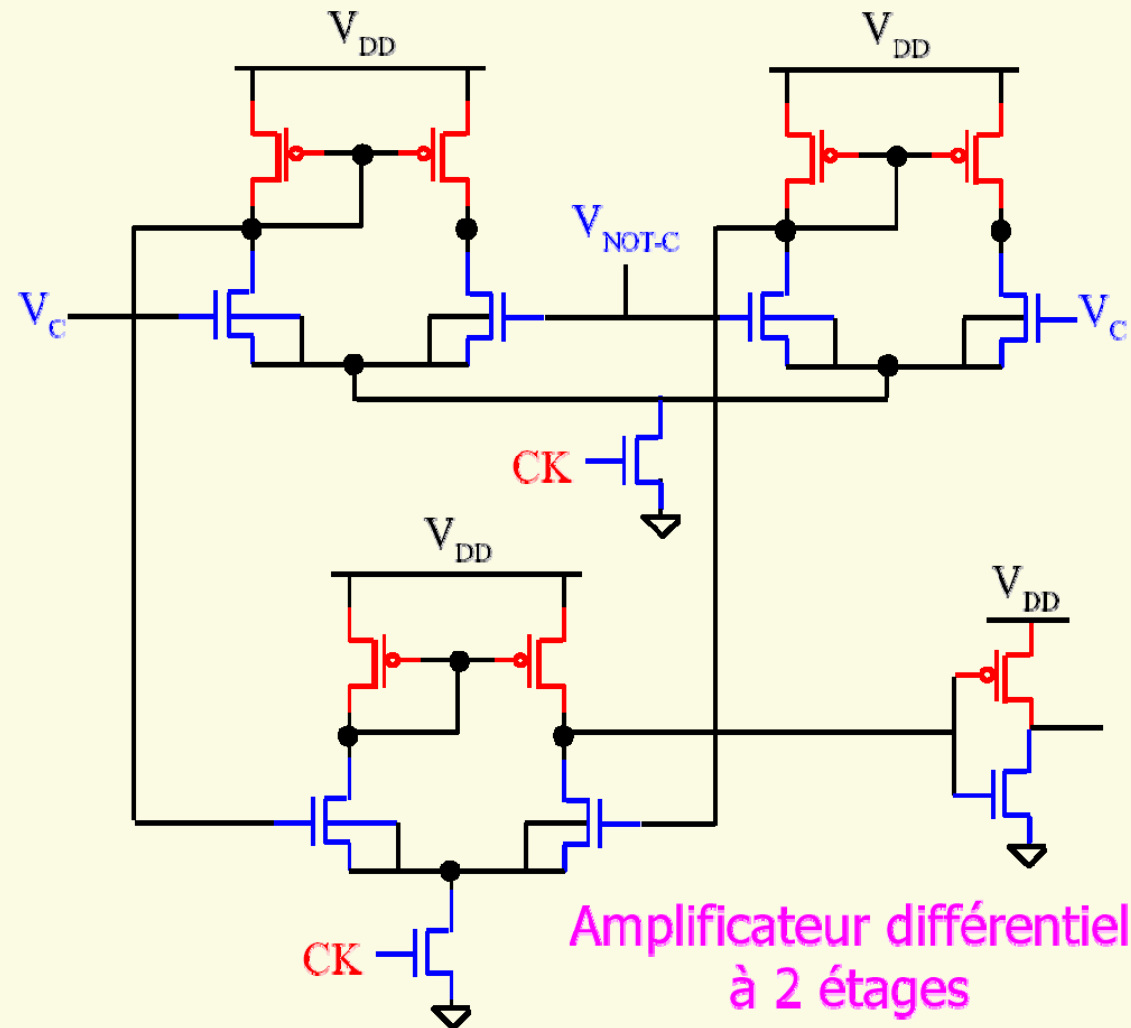
# Un exemple d'amplificateur



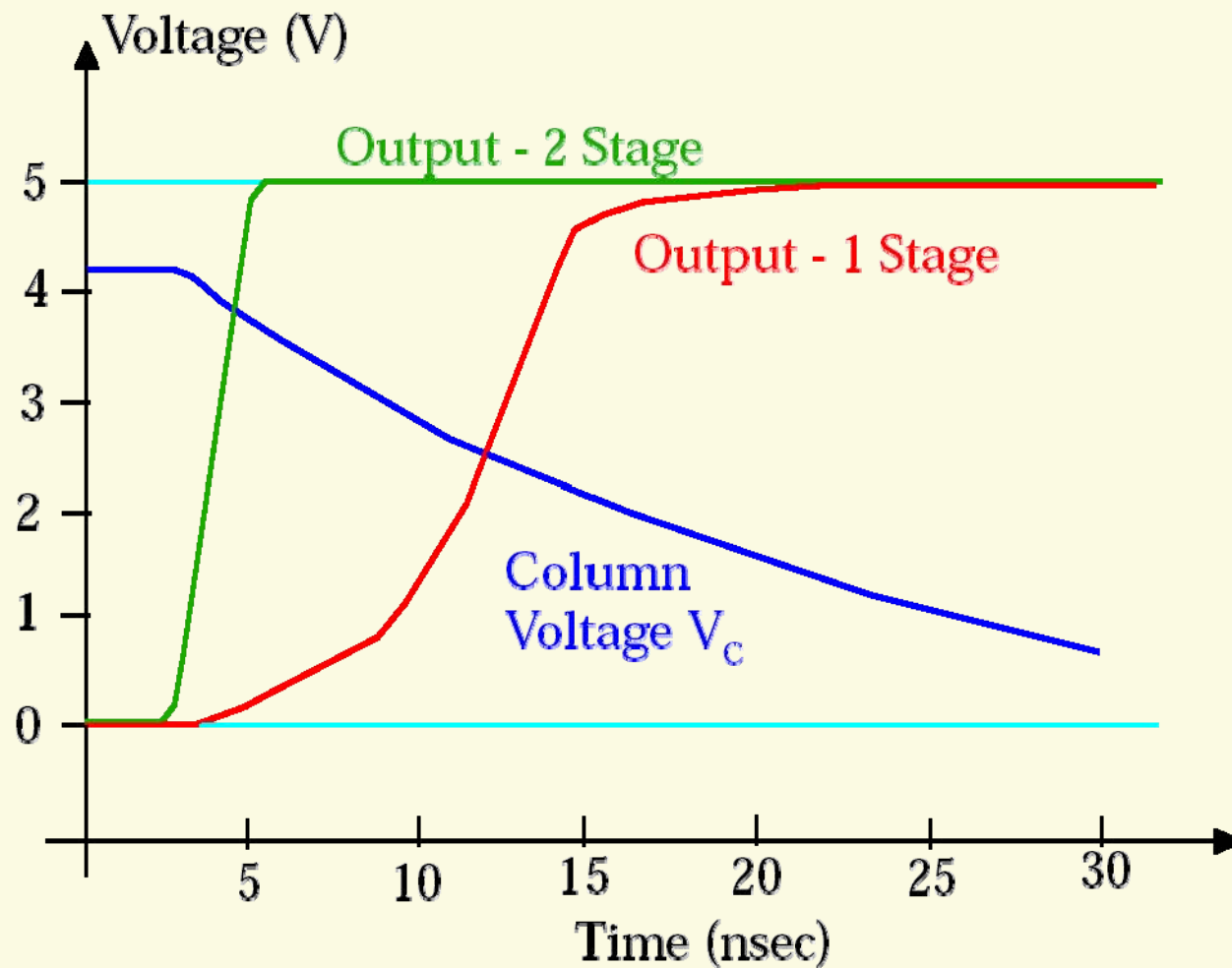
$C < \bar{C}$  : M1 bloqué,  $V_0 = 0$  et  $V_{ON} = 1$

$C > \bar{C}$  : M2 bloqué,  $V_0 = 1$  et  $V_{ON} = 0$

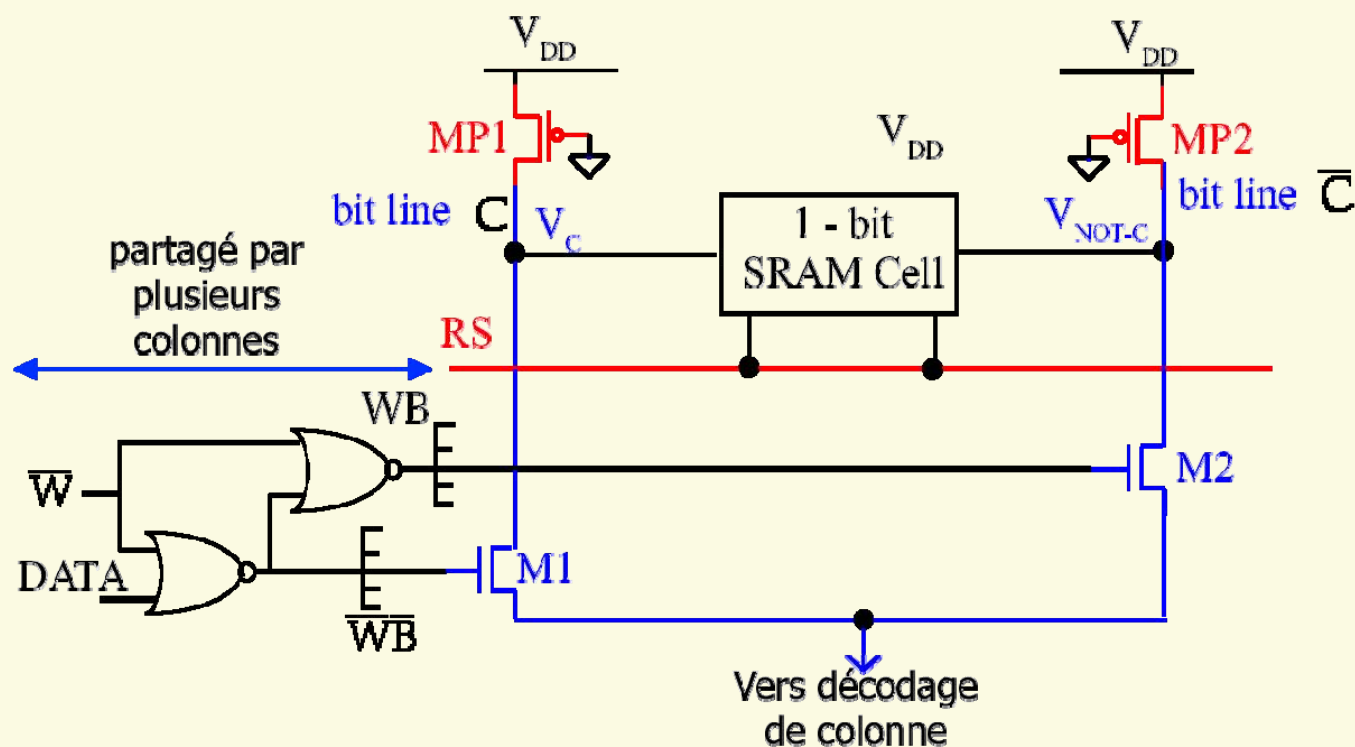
## Un deuxième exemple d'amplificateur



# Réponse typique des amplificateurs



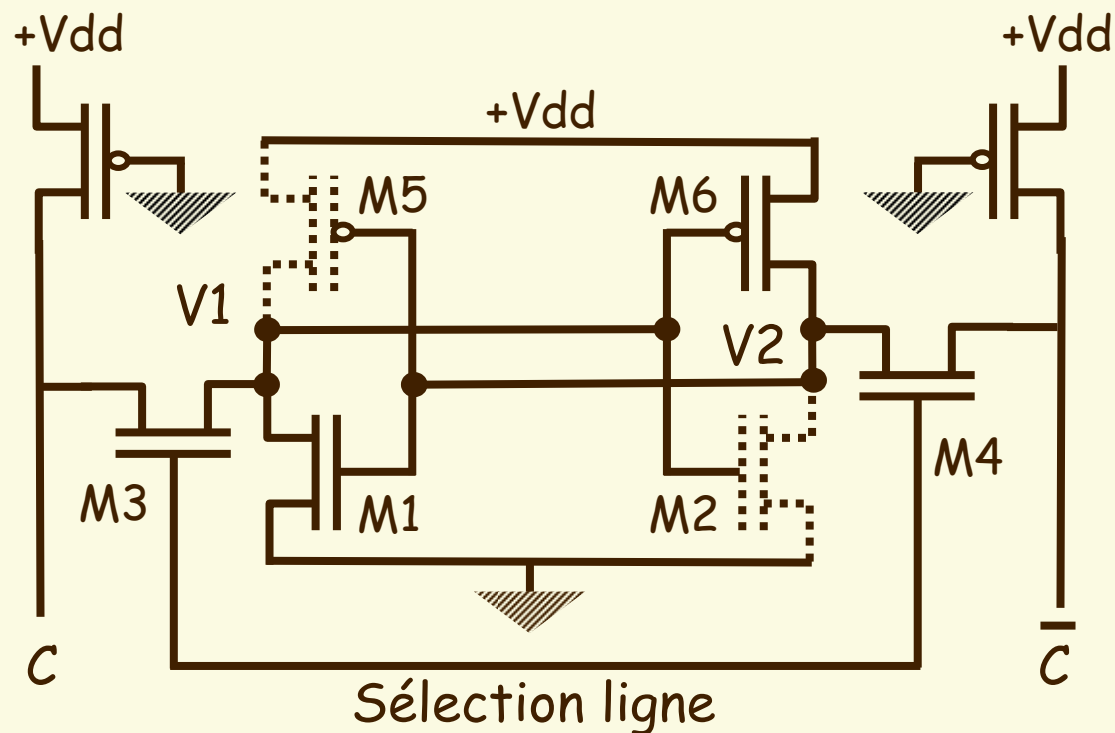
## Principe de l'écriture dans une cellule



NOT-W	DATA	NOT-WB	WB	OPERATION
0	1	0	1	M1 OFF, M2 ON $\rightarrow V_{\text{NOT-C}}$ LOW
0	0	1	0	M1 ON, M2 OFF $\rightarrow V_{\text{C}}$ LOW
1	X	0	0	M1 OFF, M2 OFF $\rightarrow V_{\text{C}} = V_{\text{NOT-C}}$ HIGH

# Ecriture d'un 0

**M3, M4 conducteurs** car Sélection = 1  
On force  $C = 0$



$C = 0 = V1$  donc :

M6 passant

M2 bloqué

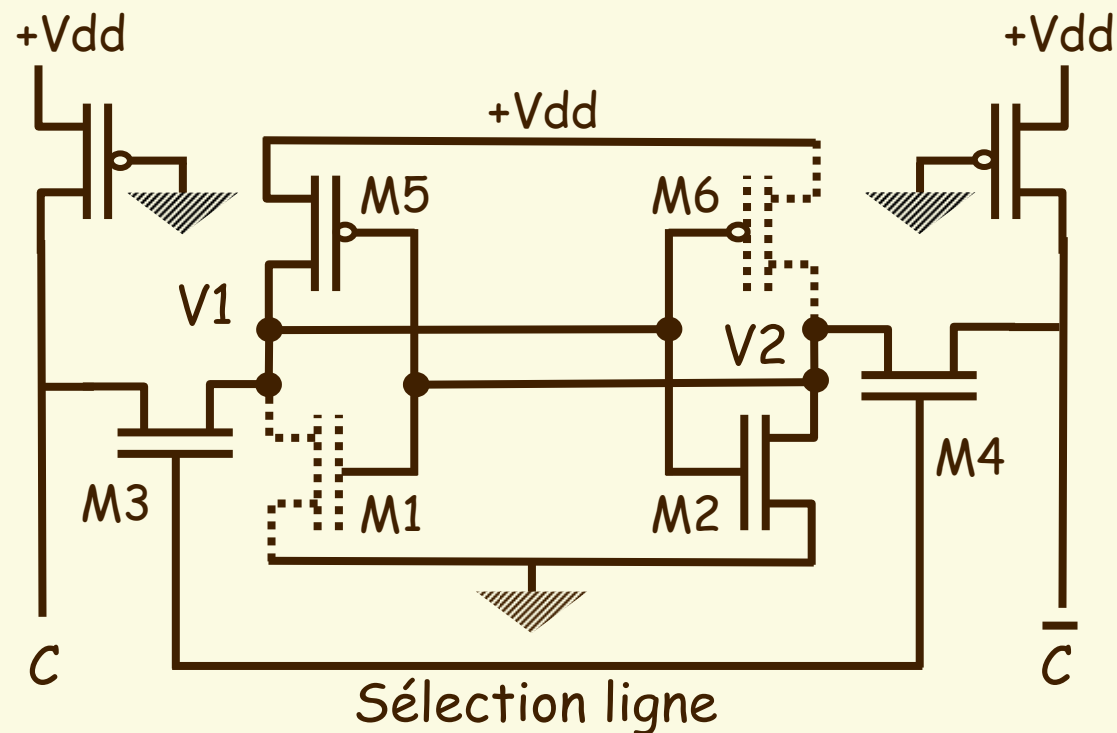
D'où  $V2 = 1 = \overline{C}$

M1 passant

M5 bloqué

# Ecriture d'un 1

M3, M4 conducteurs car Sélection = 1  
On force  $\bar{C} = 0$



$\overline{C} = 0 = V_2$  donc :

## M5 passant

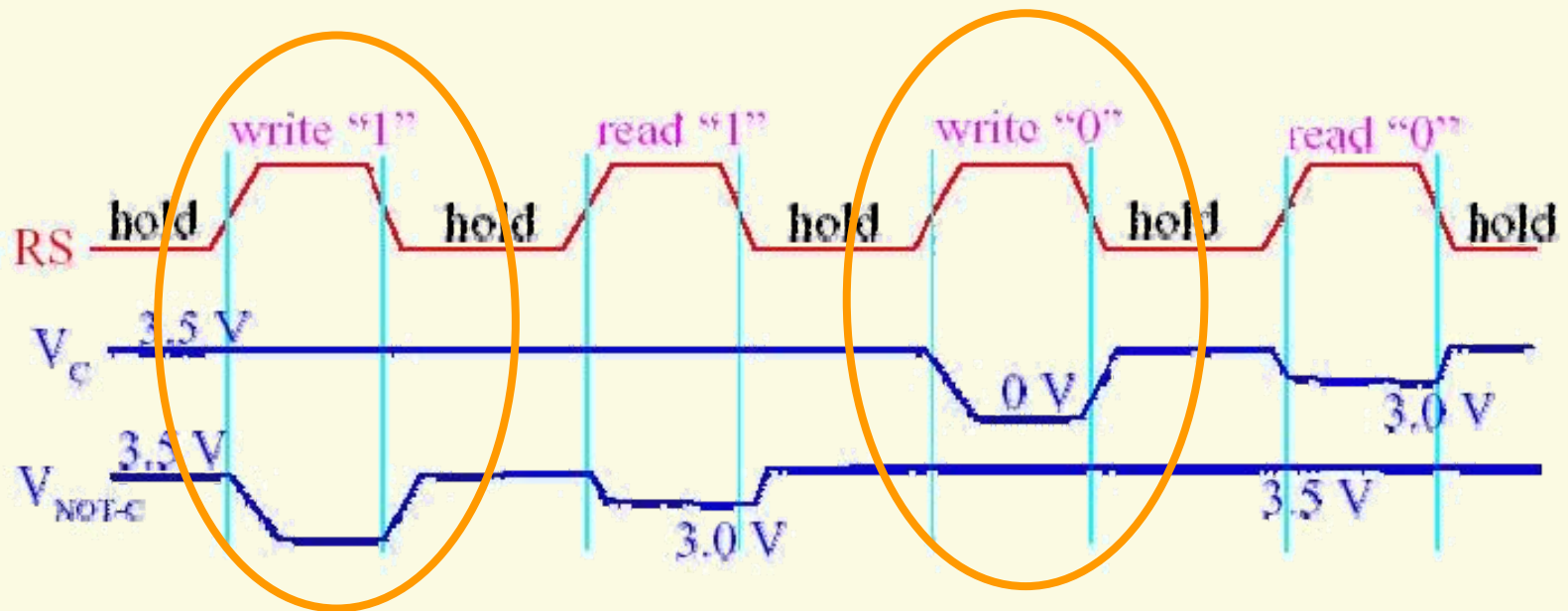
## M1 bloqué

D'où  $V1 = 1 = C$

## M2 passant

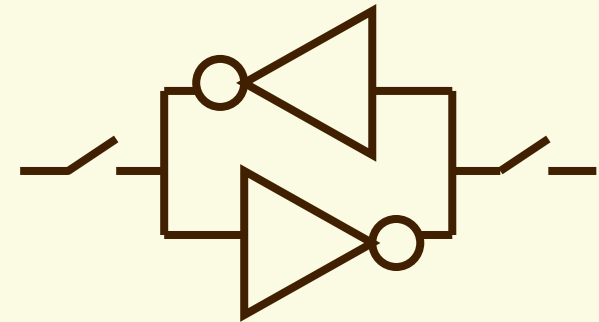
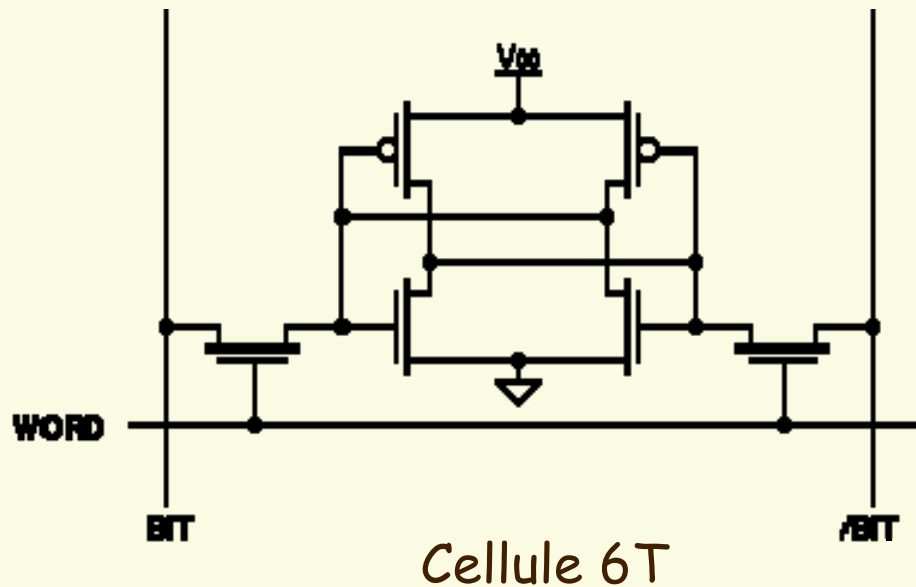
## M6 bloqué

# Chronogramme typique





# Bilan sur les mémoires SRAM



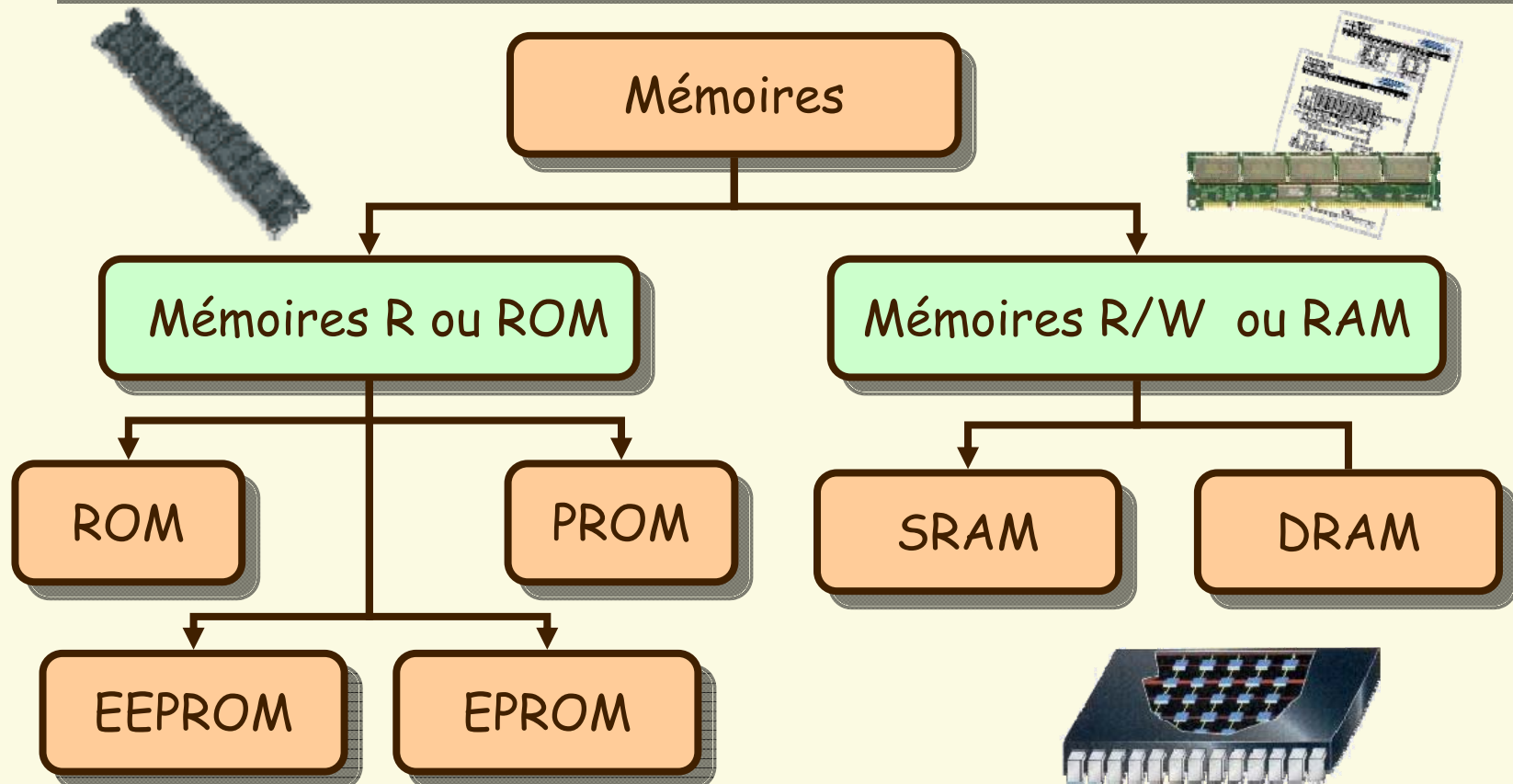
**Avantages** : Rapide, lecture non destructrice, statique

**Inconvénients** : 6T, volatile

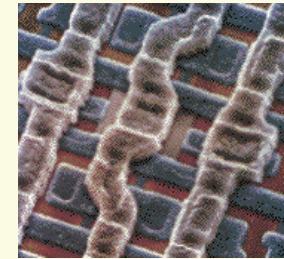
10 à 30 ns de temps d'accès (qq ns pour les petites)

# Éléments de mémorisation

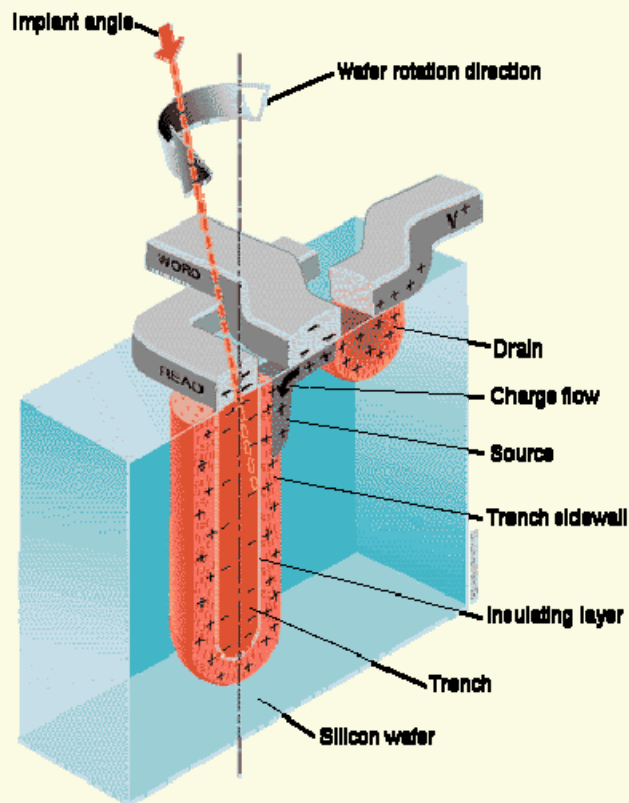
## Classification



# Mémoires de type DRAM



Définition : Mémoire RAM de type dynamique

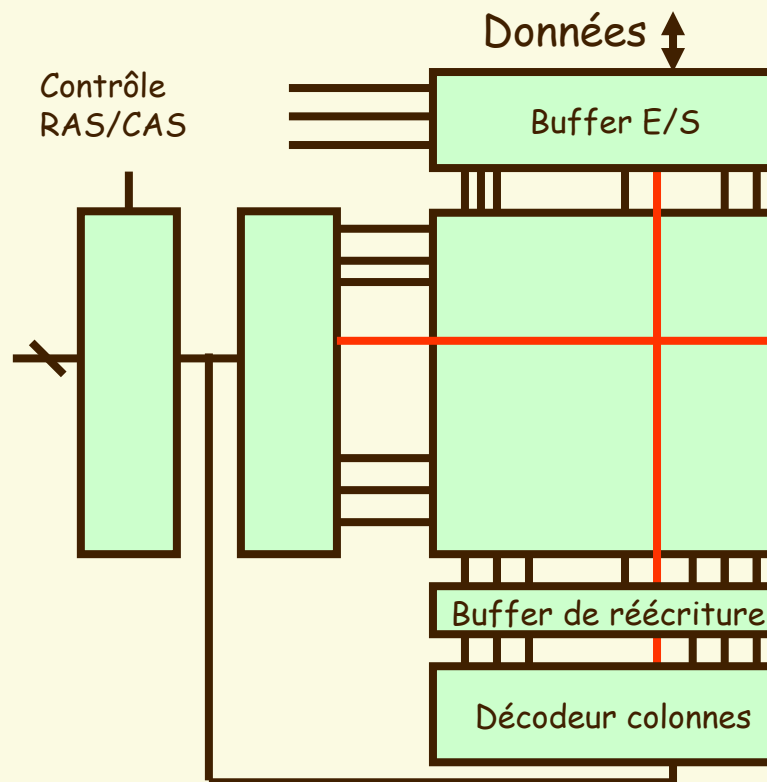


Par **dynamique**, on entend une **mémoire** dont le **contenu** doit être rafraîchi périodiquement.

**Simplification** extrême de la cellule 1 bit : **1 transistor**

# Mémoires de type DRAM

De nombreuses **évolutions** : FPM, EDO, SDRAM, DDR, RDRAM



**Fast Page Mode** : Buffer = Latch

**Enhanced Data Out** : Buffer = flipflop

**SDRAM** : Synchrone 1 front

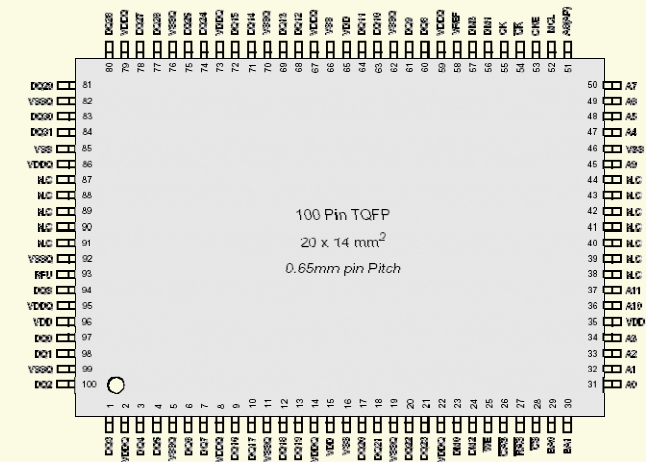
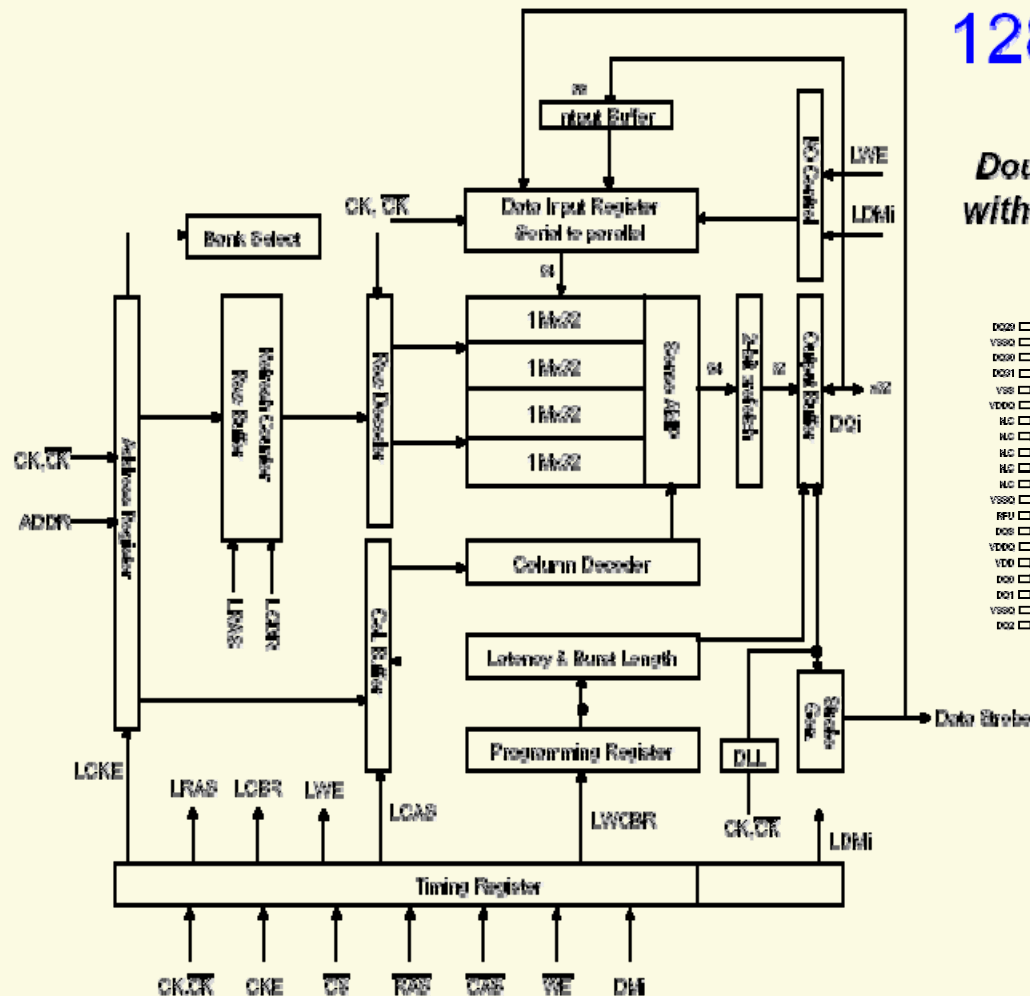
**DDR-RAM** : Synchrone 2 fronts

**RAMBUS/RDRAM** : Modif bus (no TTL)

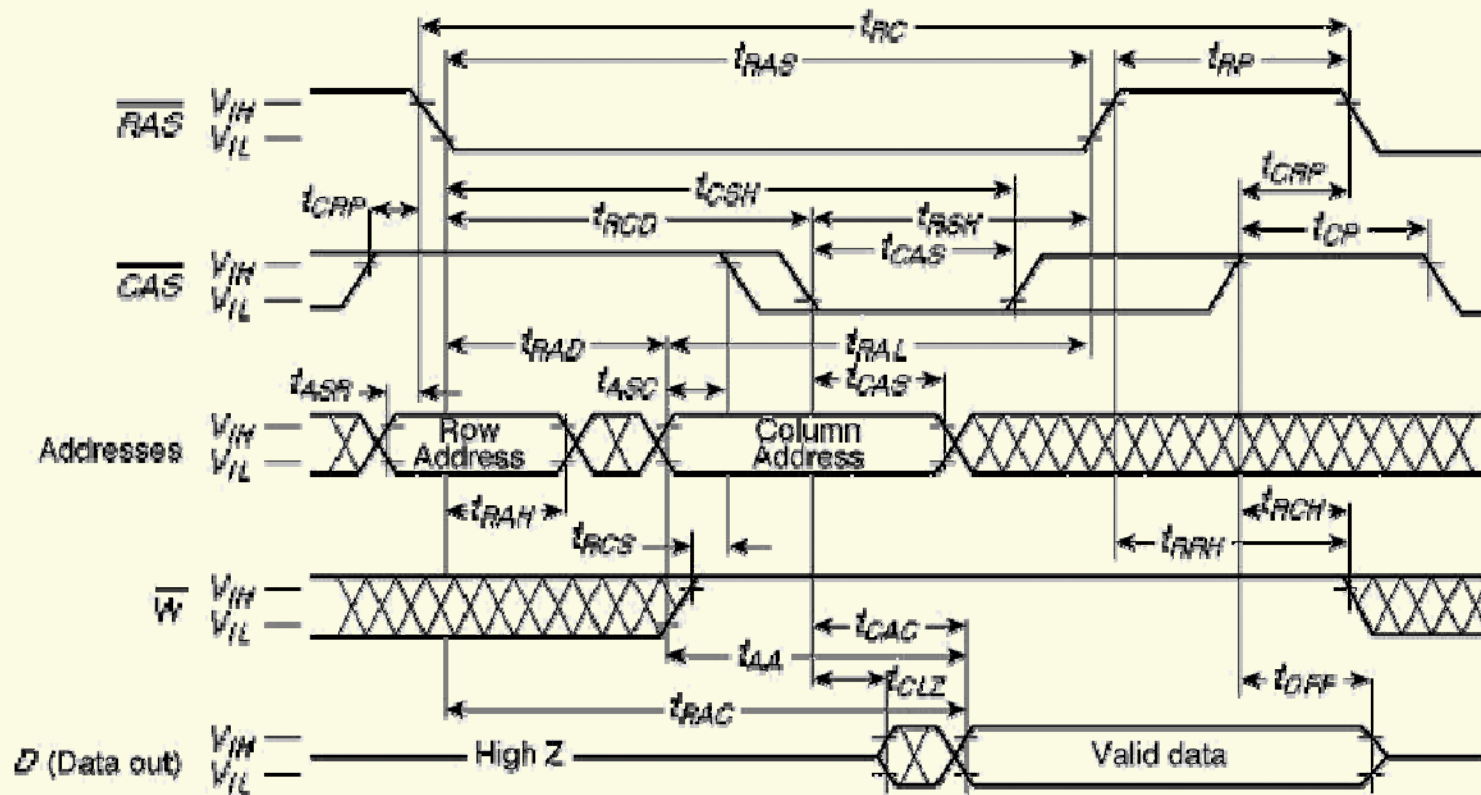
## Exemple : DDR-RAM 128 Mbits

## 128Mbit DDR SDRAM

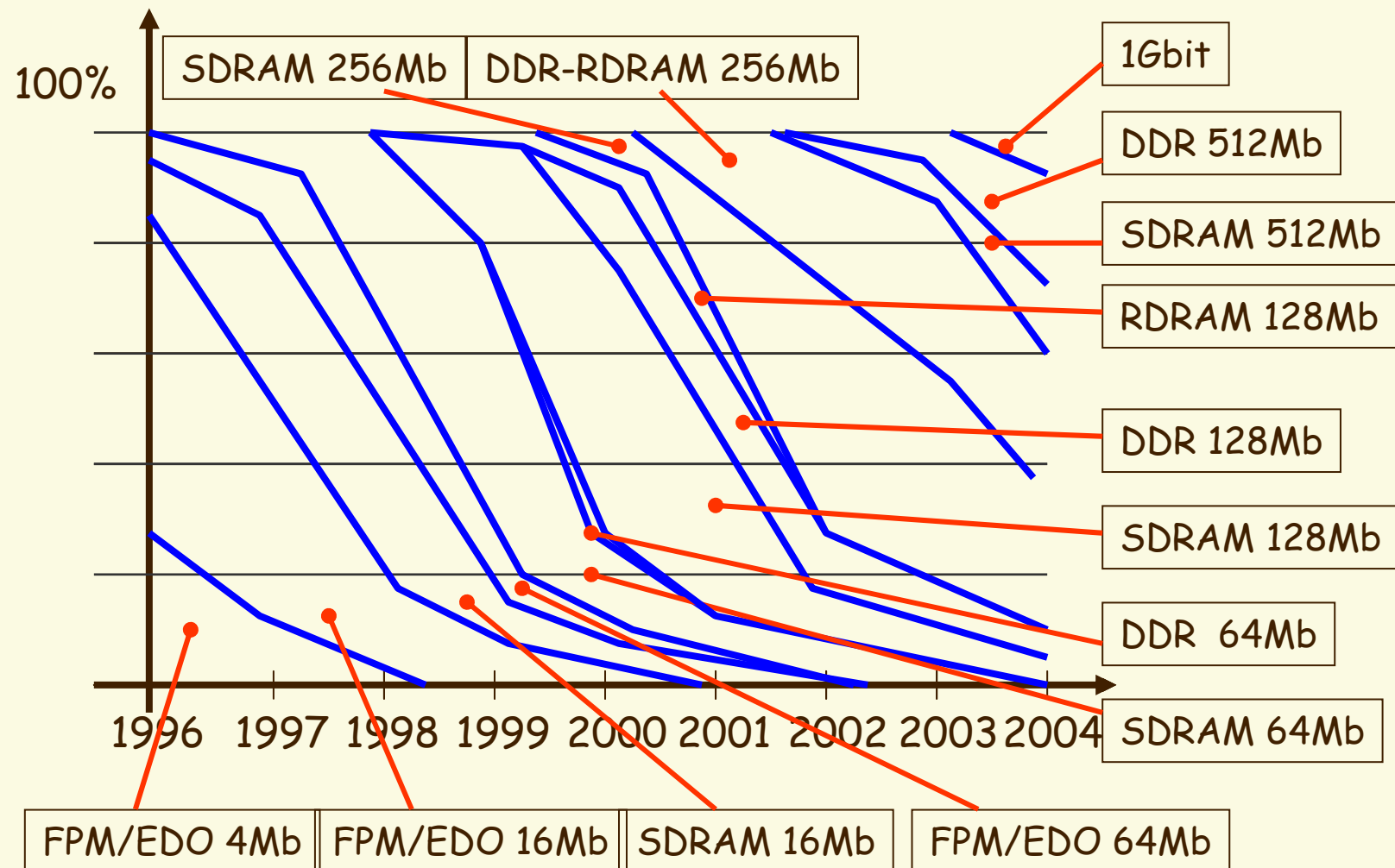
**1M x 32Bit x 4 Banks**  
**Double Data Rate Synchronous DRAM**  
**with Bi-directional Data Strobe and DLL**



# Chronogramme typique d'une DRAM



# Marché des DRAM (20 G\$ en 1999)



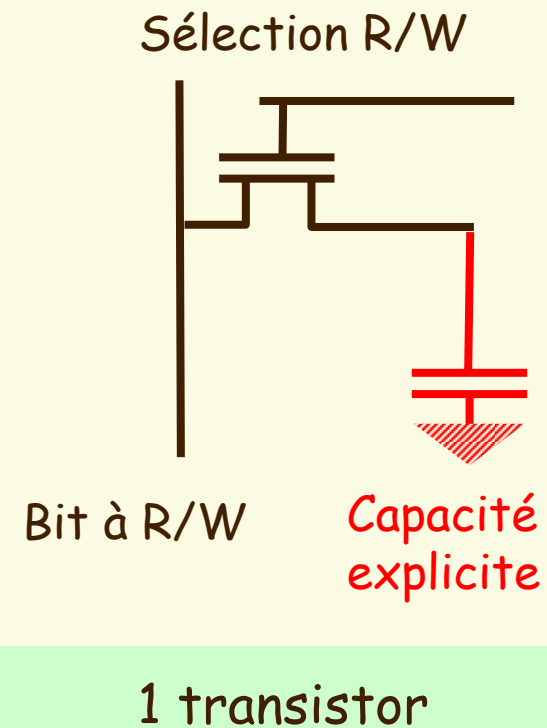
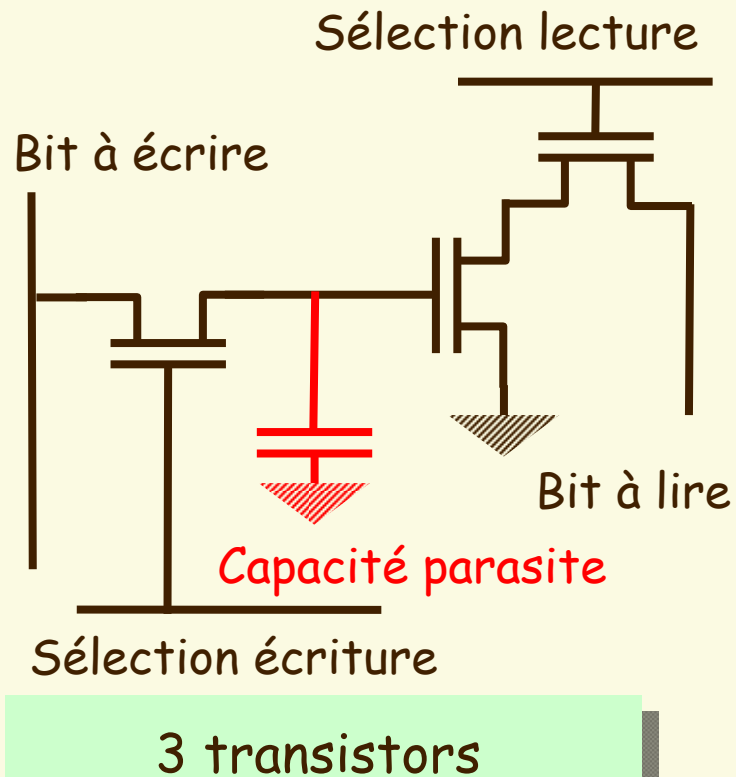
# Performances des DRAM

DYNAMIC RAM (DRAM) MEMORY TECHNOLOGIES					
Type	First Used	Clock Rate	Bus <sup>***</sup> Width	Peak Bandwidth	Voltage
FPM (60, 70ns)	1990	25MHz	64 bits	200 MBps	5v
EDO (50, 60, 70ns)	1994	40MHz	64 bits	320 MBps	5v
SDRAM (66MHz)	1996	66MHz	64 bits	528 MBps	3.3v
SDRAM (100MHz)	1998	100MHz	64 bits	800 MBps	3.3v
SDRAM (133MHz)	1999	133MHz	64 bits	1.1 GBps	3.3v
RDRAM (Direct Rambus)	1999	400MHz (x2)	16 bits	1.6 GBps	2.5v
DDR SDRAM (100MHz)	2000	100MHz (x2)	64 bits	1.6 GBps	3.3v
DDR SDRAM (133MHz)	2000	133MHz (x2)	64 bits	2.1 GBps	3.3v
SLDRAM	---	400MHz (x2)	16 bits	1.6 GBps	2.5v
<sup>***</sup> Maximum PC module data width. 64 bits started with 75MHz Pentium. The specifications in the top half of this chart were obtained from Kingston Technology Company, Fountain Valley, CA ( <a href="http://www.kingston.com">www.kingston.com</a> ).					



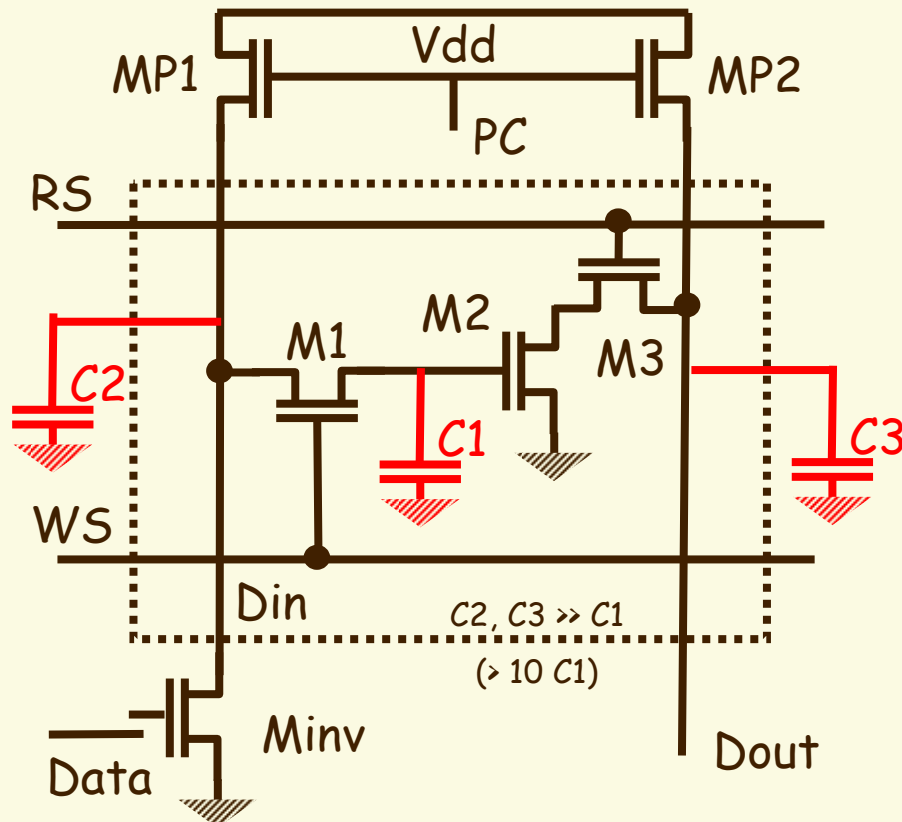
# Mémoires de type DRAM

## Différentes complexités de conception



## Circuit complet pour une DRAM 3T

## Rajout de transistors de précharge

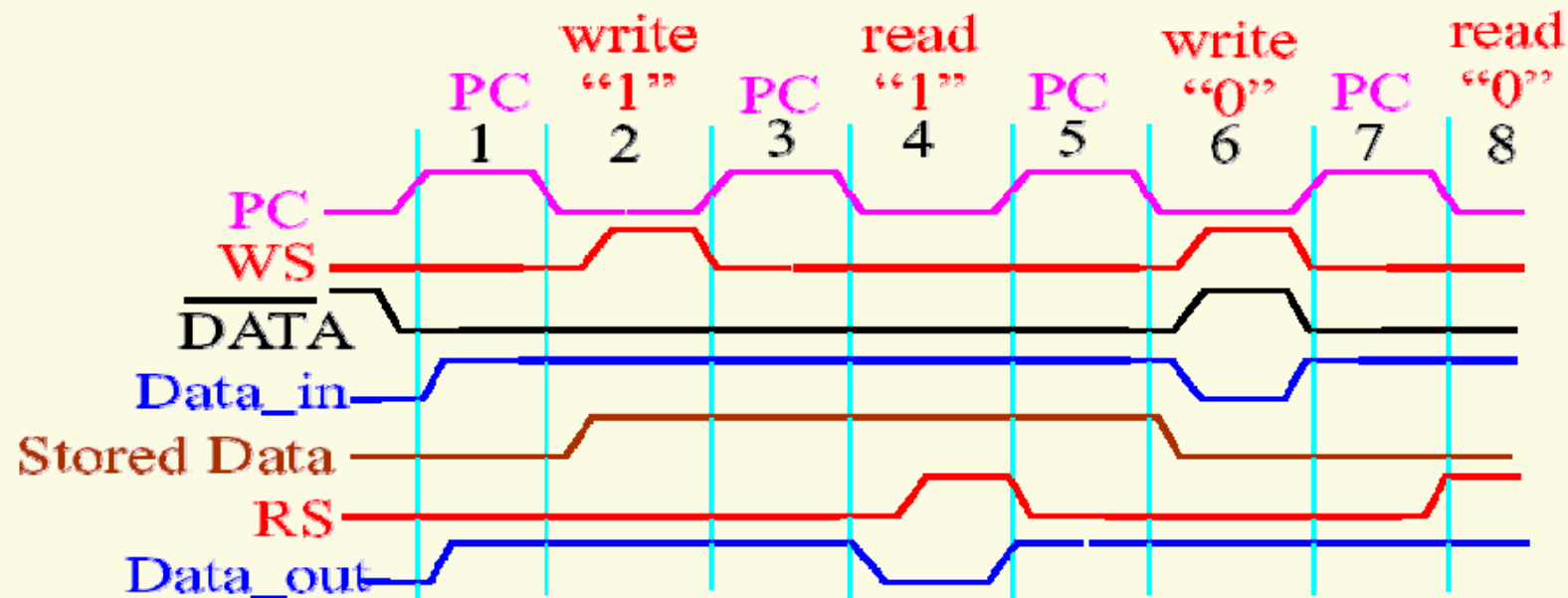
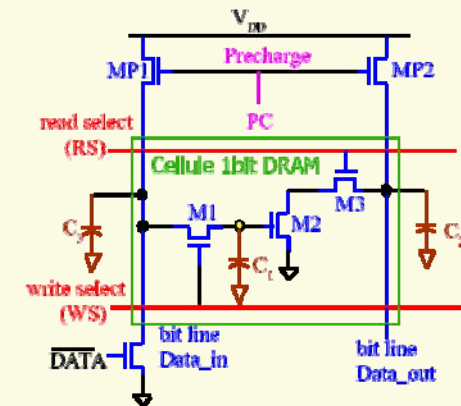


MP1, MP2 pour la  
précharge de la cellule  
lorsque  $PC = 1$

Un transistor M<sub>inv</sub>  
pour inverser la valeur  
de DATA en phase  
d'écriture

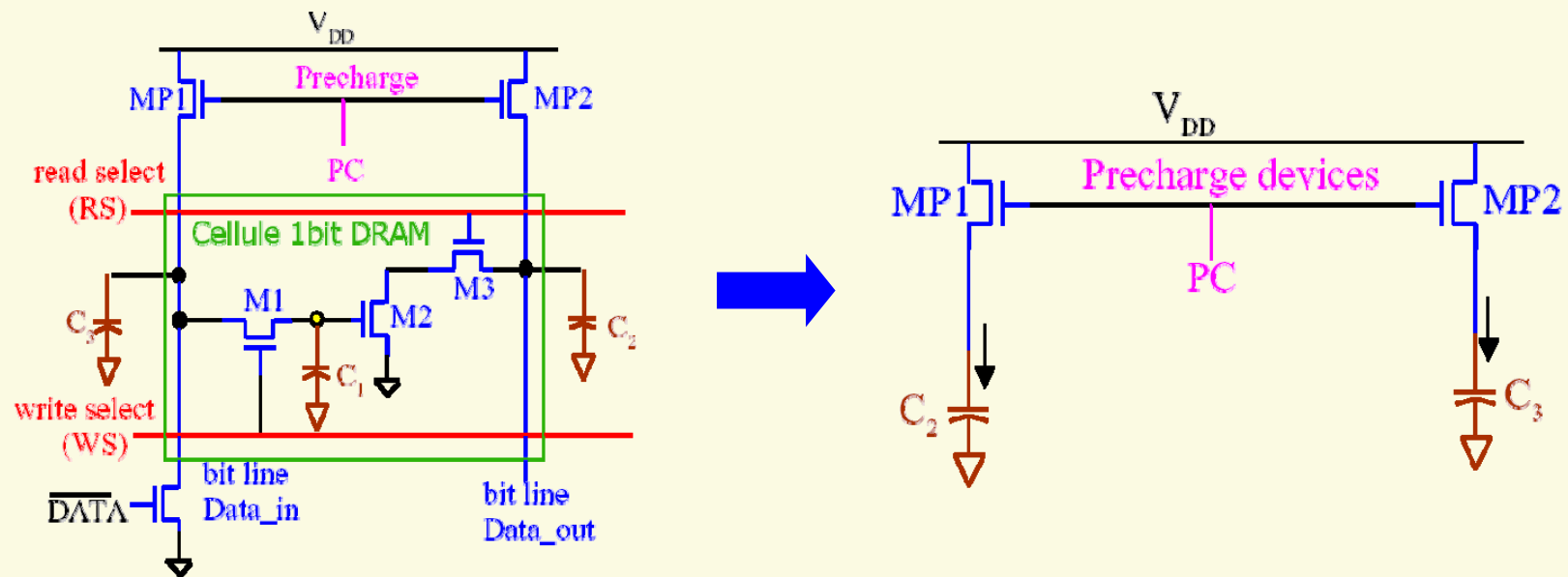
# Circuit complet pour une DRAM 3T

Un chronogramme typique



# Phase de précharge

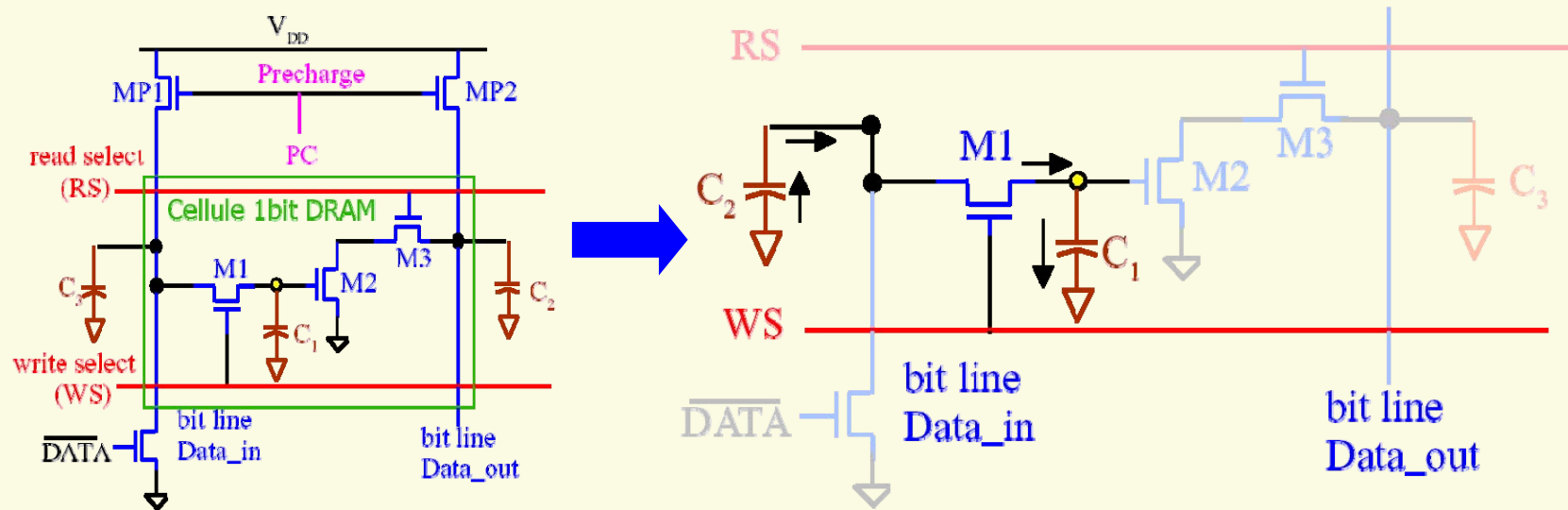
PC = 1 : MP1 et MP2 conduisent



La phase de précharge avant chaque cycle de R/W a pour objectif de charger les 2 capacités parasites C2 et C3

# Ecriture d'un 1

$WS = 1, RS = 0$  : mémorisation d'un 1 dans  $C1$



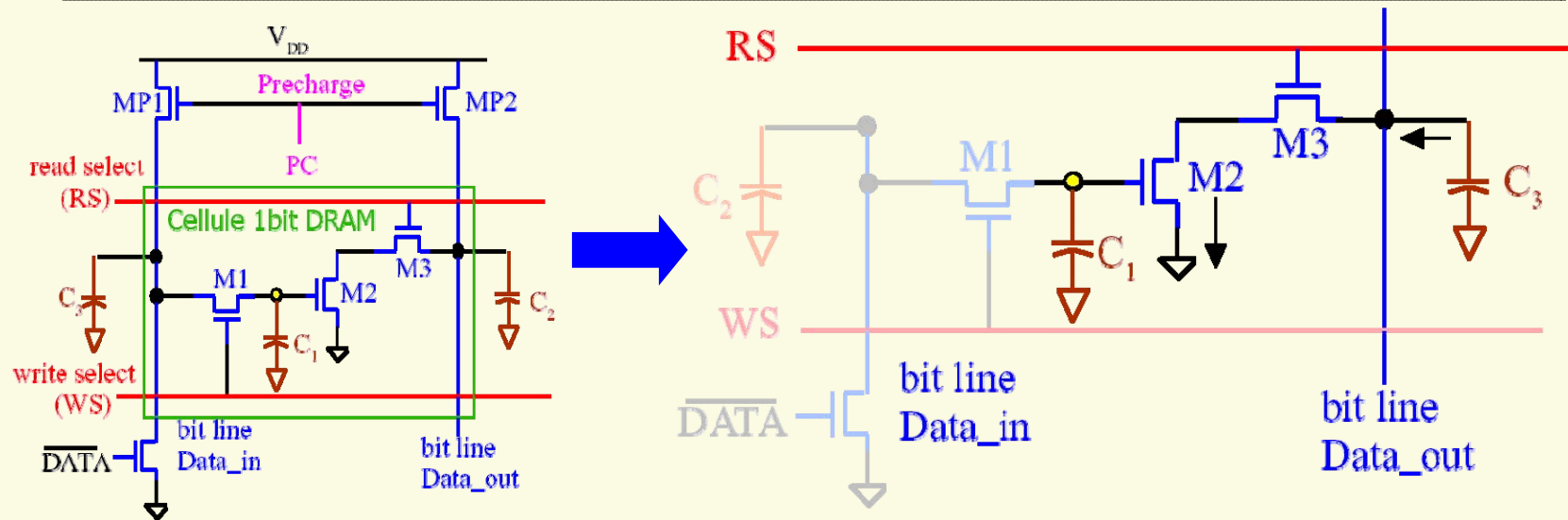
Data = 0 donc Minv bloqué

WS = 1 donc M1 passant et RS = 0 donc M3 bloqué

$C_2$  se décharge à travers M1 et donc  $C_1$  se charge

# Lecture d'un 1

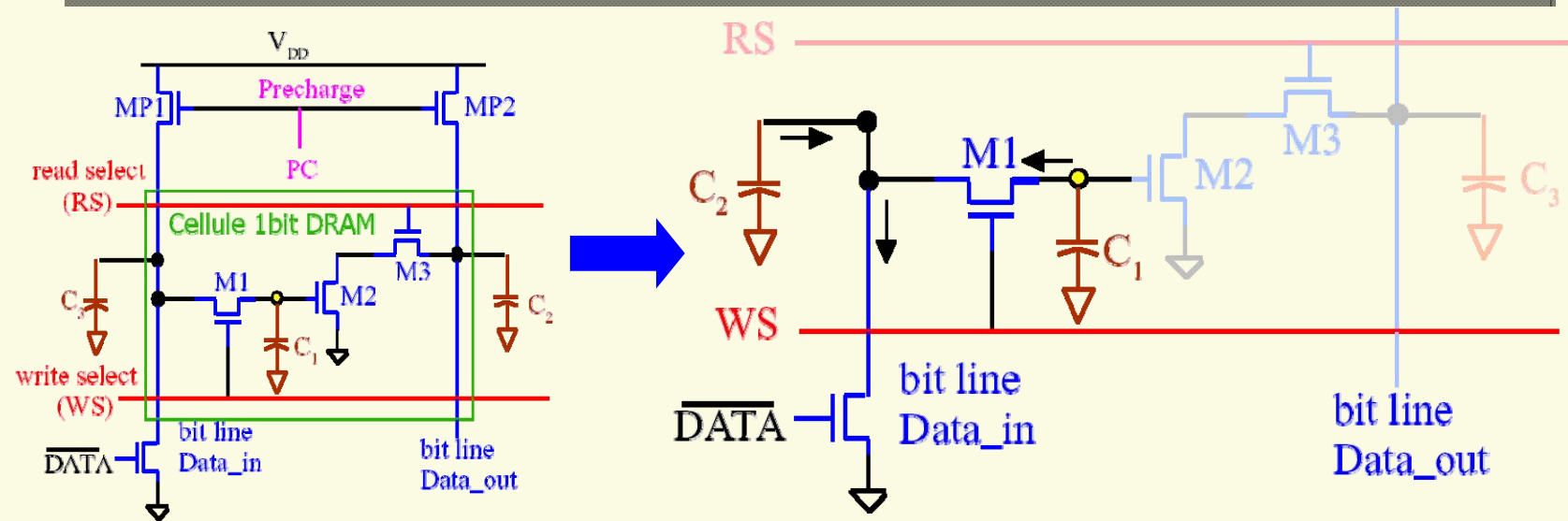
$WS = 0, RS = 1$  :  $Data\_out = 0$



$WS = 0$  donc M1 bloqué et  $RS = 1$  donc M3 passant  
C1 est chargé donc M2 est passant  
C3 se décharge donc à travers M3 et M2

# Ecriture d'un 0

$WS = 1, RS = 0$  : mémorisation d'un 0 dans  $C1$



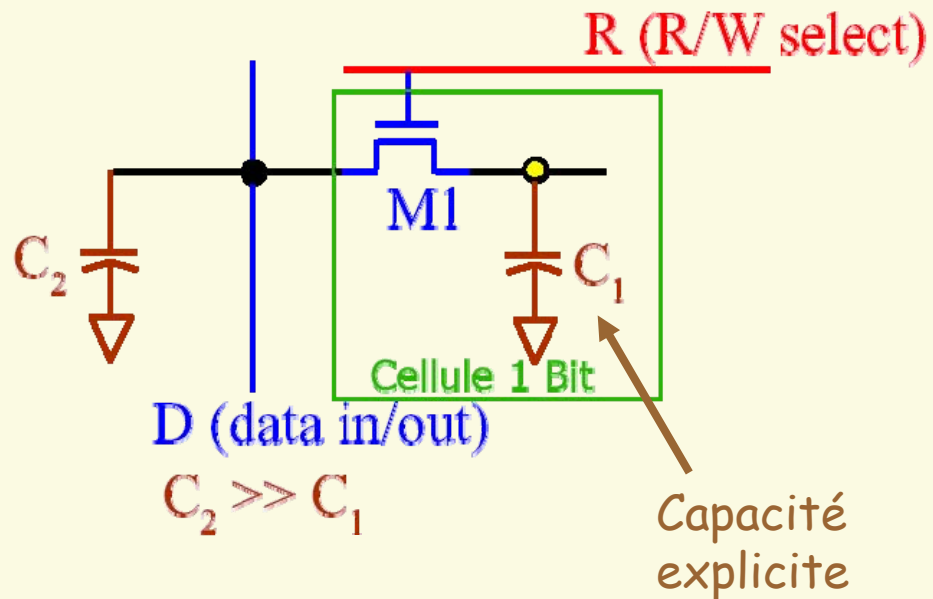
Data = 1 donc Minv passant et Data\_in = 0  
WS = 1 donc M1 passant et RS = 0 donc M3 bloqué  
C2 et C1 se déchargent à travers M1 et Minv





# Mémoires DRAM 1 T

1 Transistor et 1 capacité

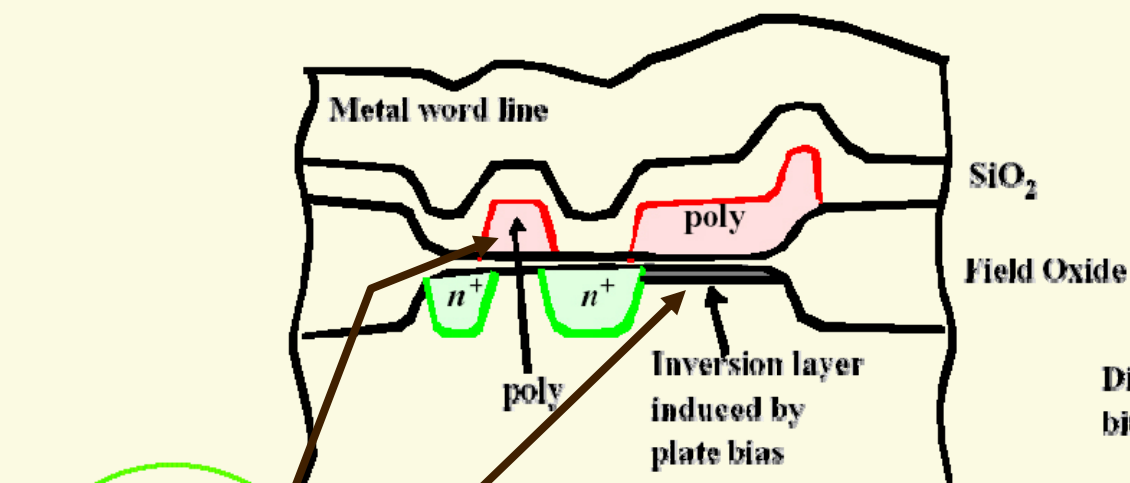
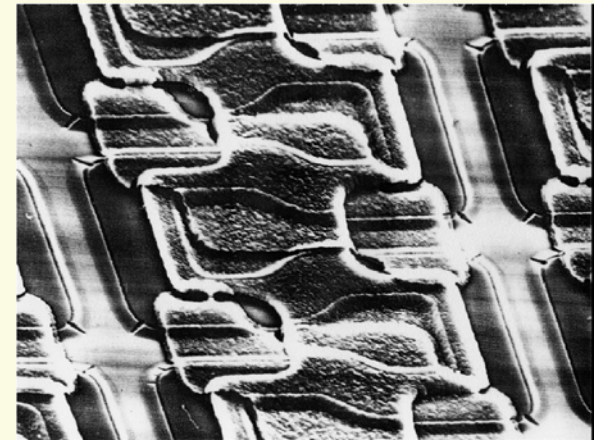


La capacité explicite  $C_1$  stocke la donnée

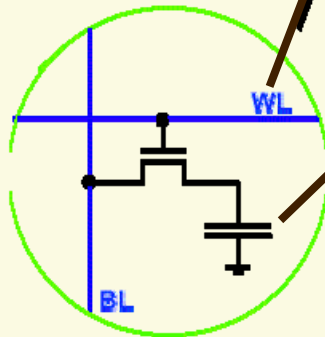
$C_1$  de valeur comprise entre 30 et 100 fF

# Mémoires DRAM 1 T

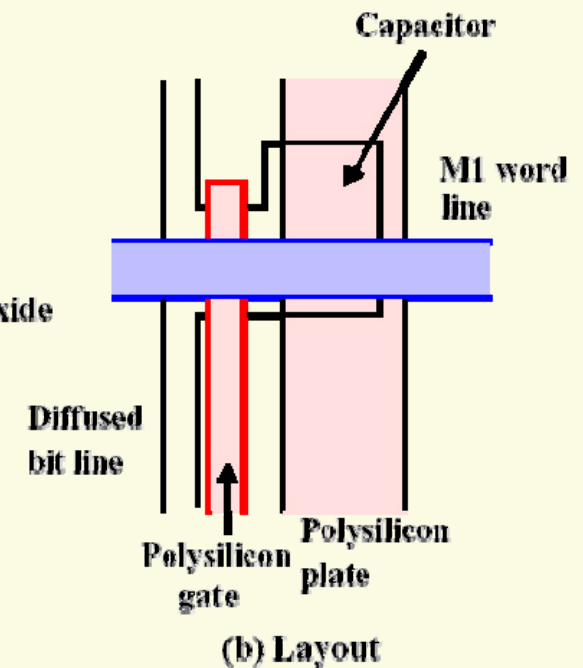
1 Transistor et 1 capacité



(a) Cross-section



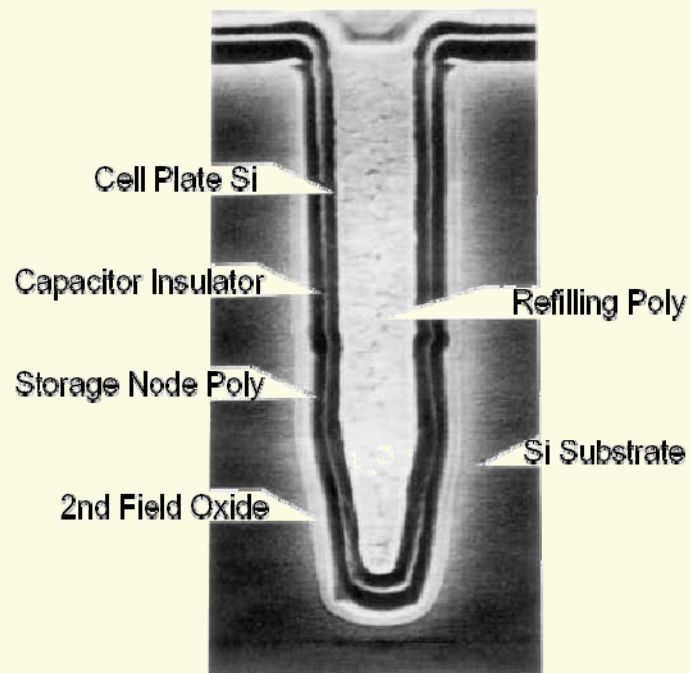
Capacité Poly/métal  
 $100 \text{ fF}/\mu\text{m}^2$  en techno  $0.25\mu\text{m}$



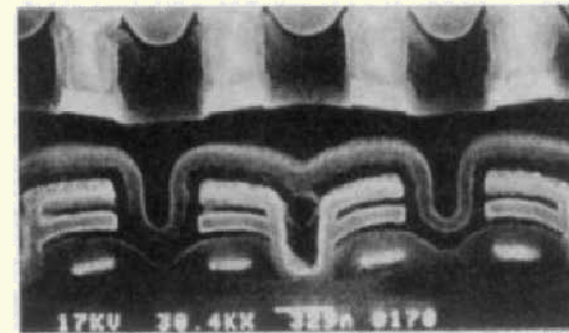
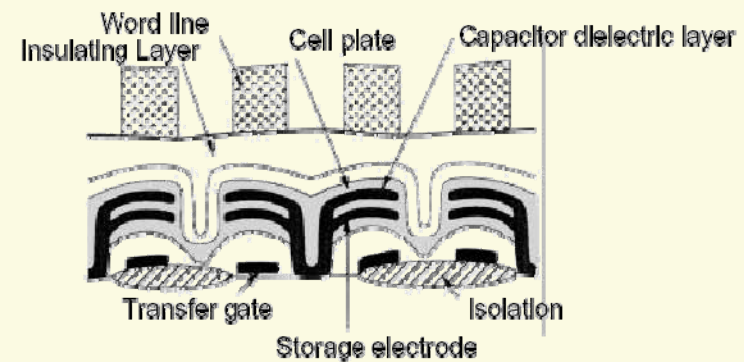
(b) Layout

# Mémoires DRAM 1 T

Technologies avancées :



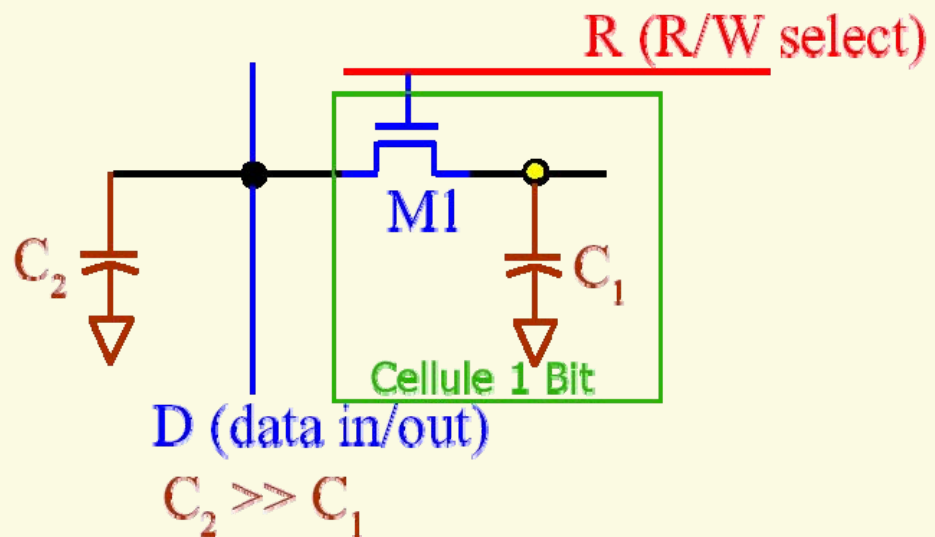
Trench Cell



Stacked-capacitor Cell

# Ecriture

Selection R/W = 1 donc M1 passant

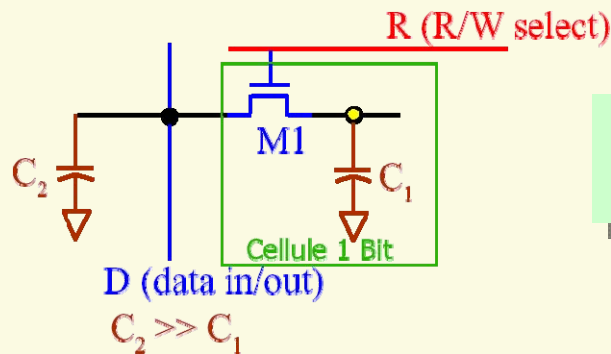


Si data = 1  
Charge de  $C_1$

Si data = 0  
Décharge de  $C_1$

# Lecture

Selection R/W = 1 donc M1 passant



Précharge de C2 avant lecture

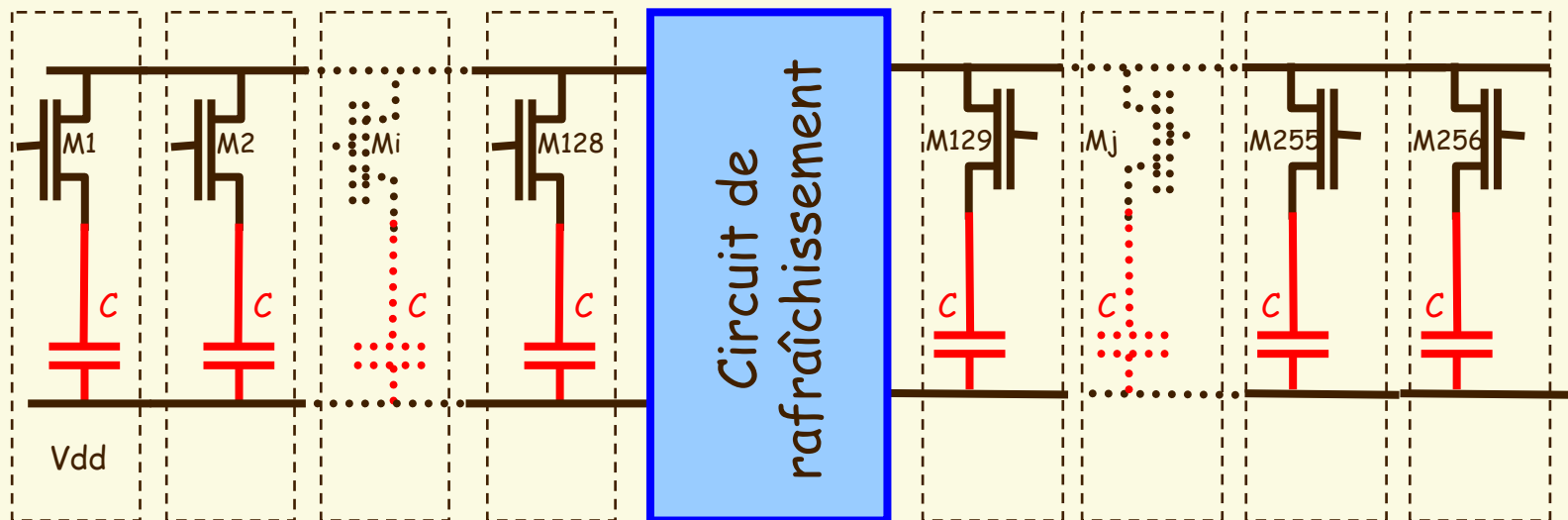
Cas 1 : C1 non chargée donc C2 se décharge dans C1 et donc la valeur de D diminue légèrement (~250 mv)

Cas 2 : C1 chargée donc C2 ne se décharge pas dans C1 et donc la valeur de D reste identique

Dans tous les cas, il y a perte de la valeur stockée dans C1

# Rafraîchissement

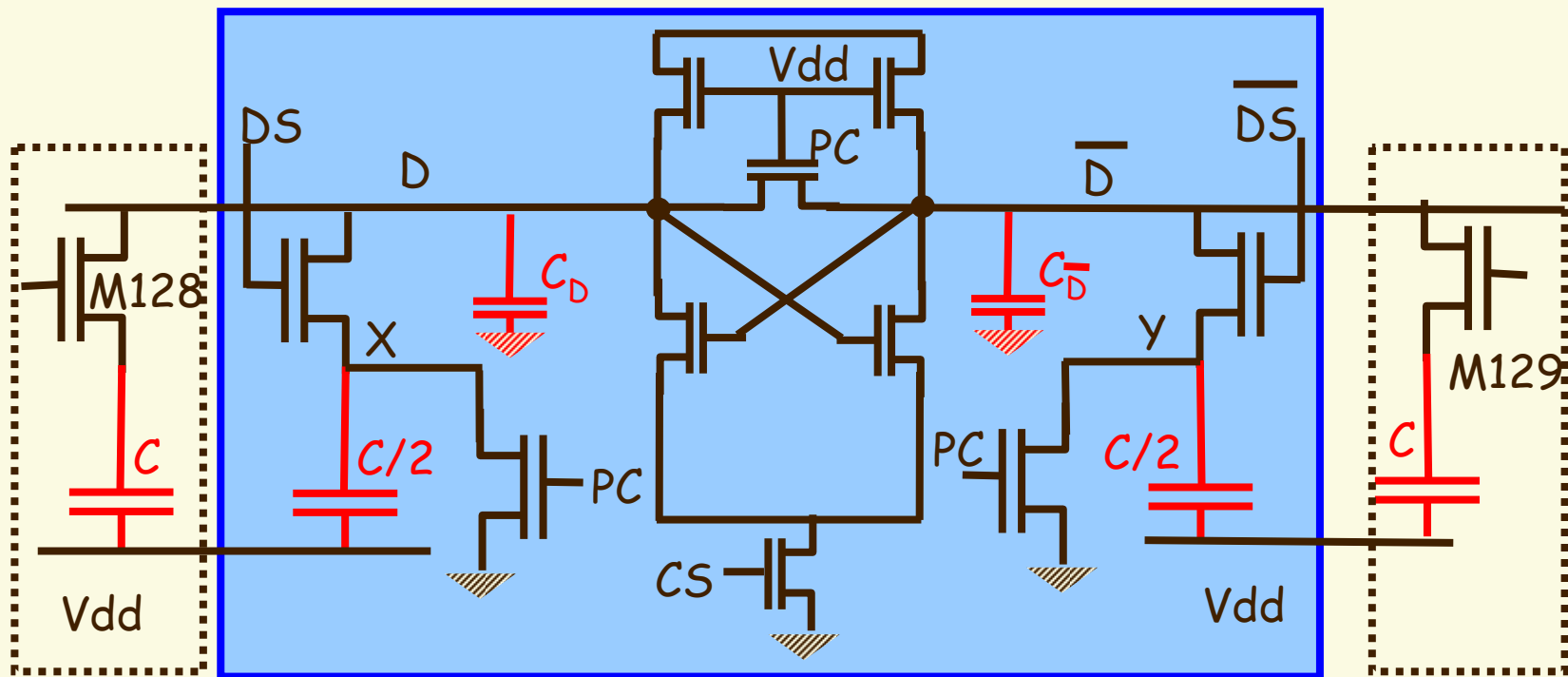
Rajout d'un circuit supplémentaire pour rafraîchir la mémoire en cas de lecture destructive.



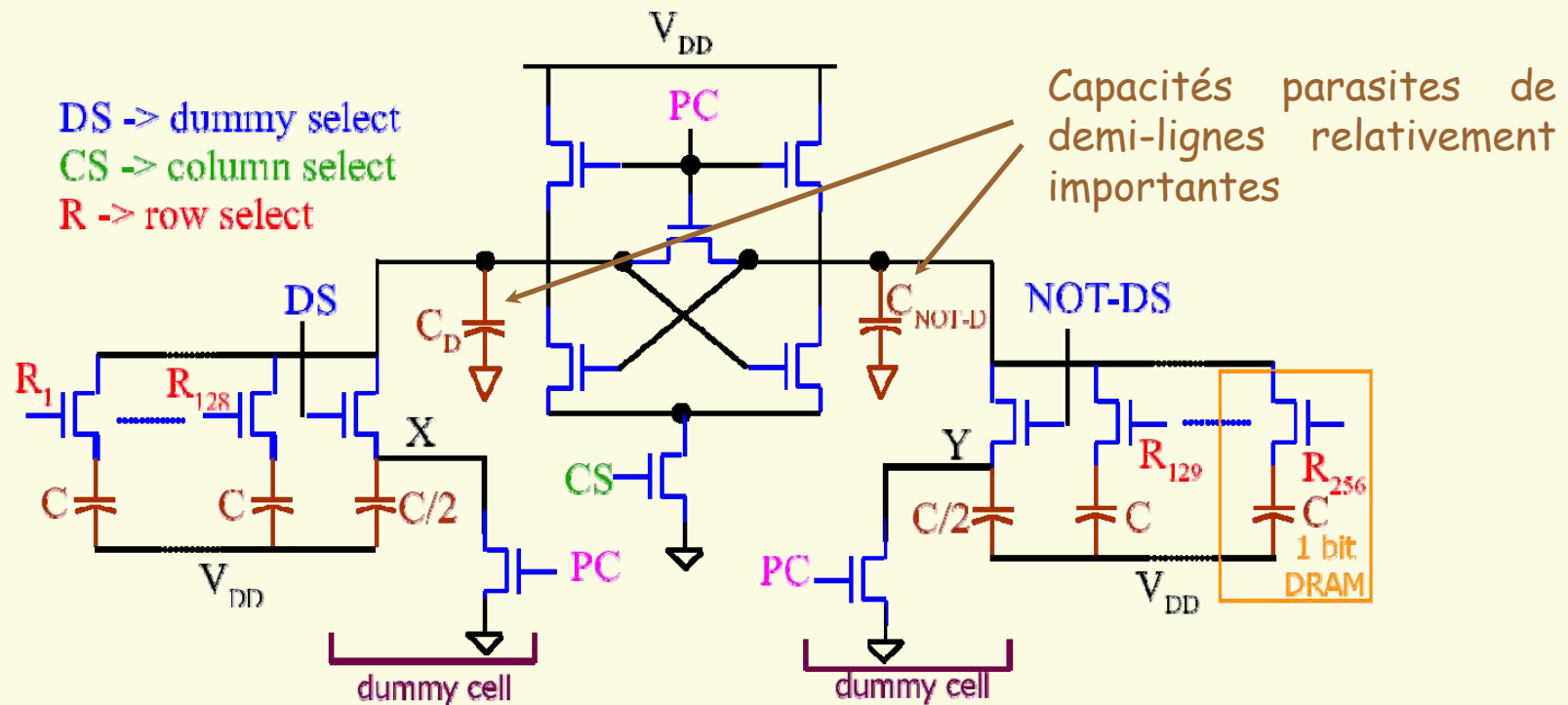
Un circuit par ligne et placé au milieu la ligne pour avoir un équilibre au niveau des capacités des cellules mémoires

# Rafrâichissement

Rajout d'un circuit supplémentaire pour rafraîchir la mémoire en cas de lecture destructrice.



# Rafraîchissement



Rajout de 2 cellules supplémentaires inutile (« dummy cell ») de capacité égale à la moitié d'une cellule mémoire

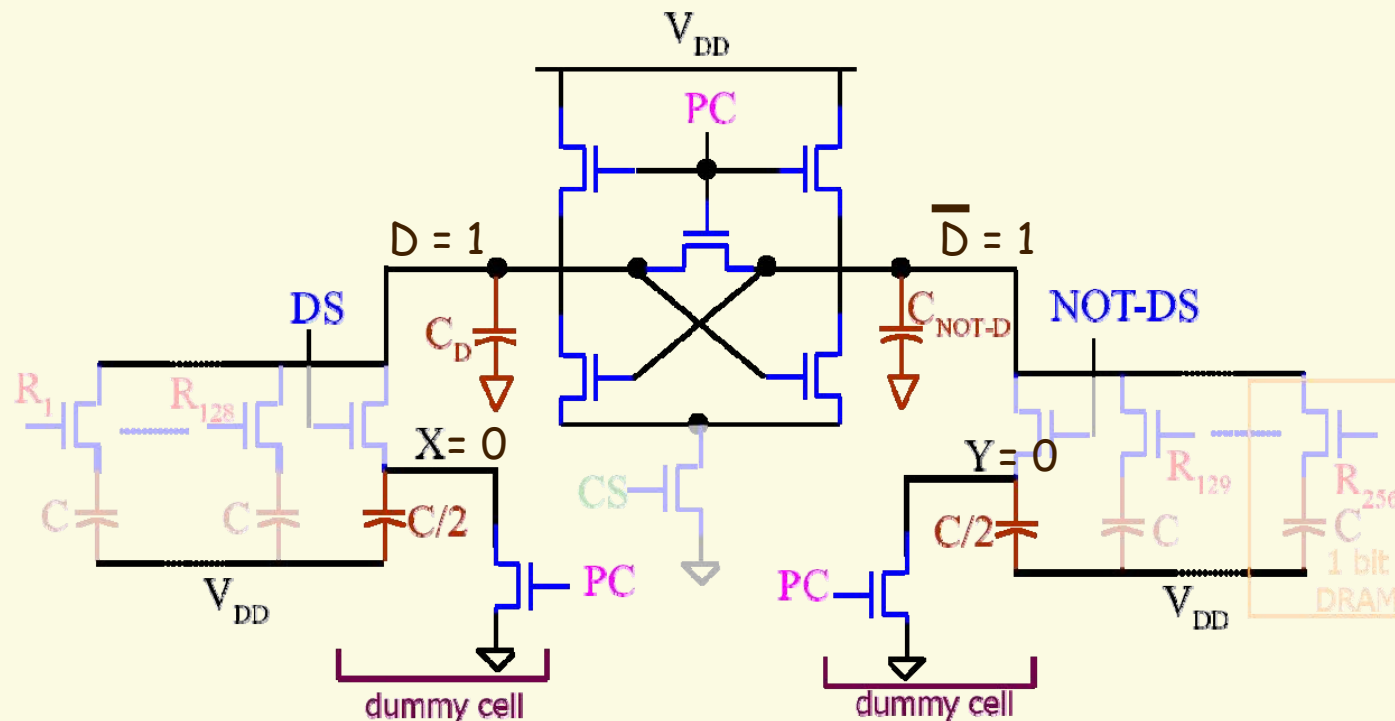


# Phase 1 du rafraîchissement

Précharge :  $PC = 1$  (tous les autres signaux sont égaux à 0)

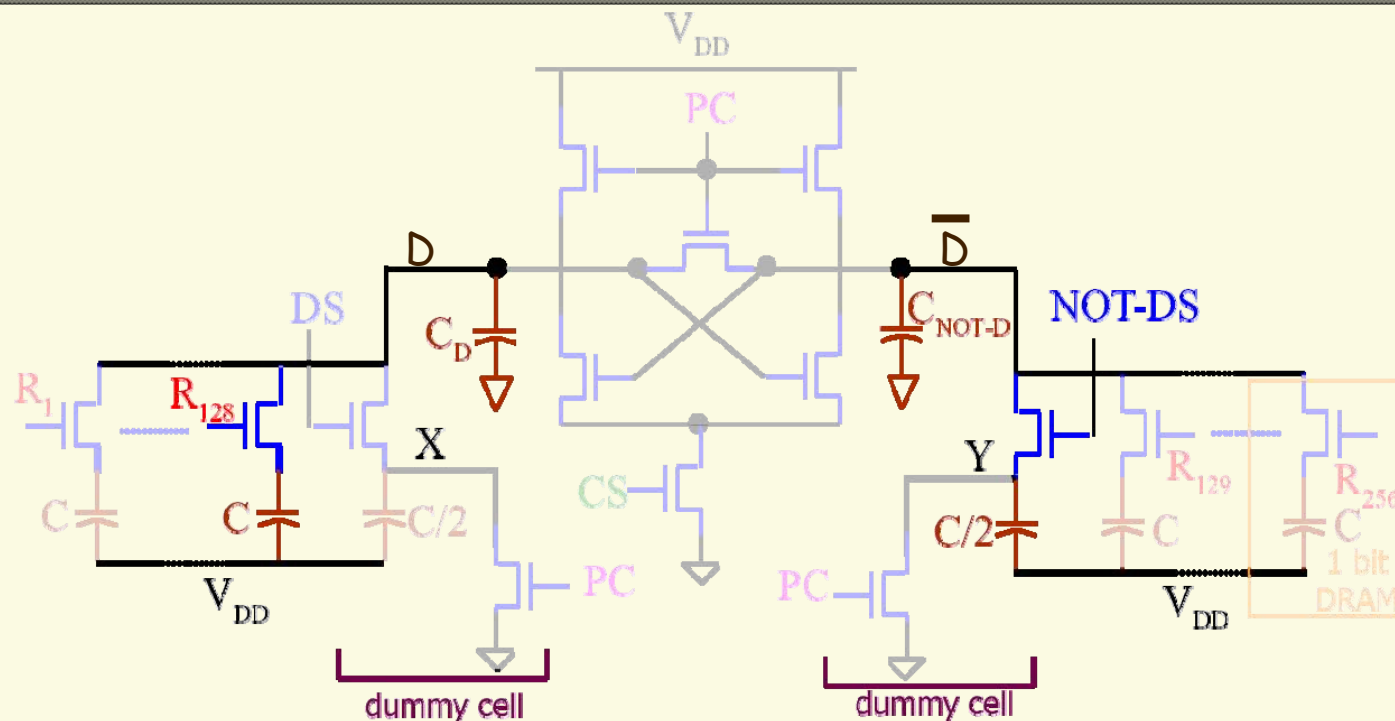
✓  $D = \bar{D} = 1$  (capacités de ligne  $C_D$  et  $C_{NOT-D}$  chargées)

✓  $X = Y = 0$  (capacités  $C/2$  associées déchargées)



## Phase 2 du rafraîchissement

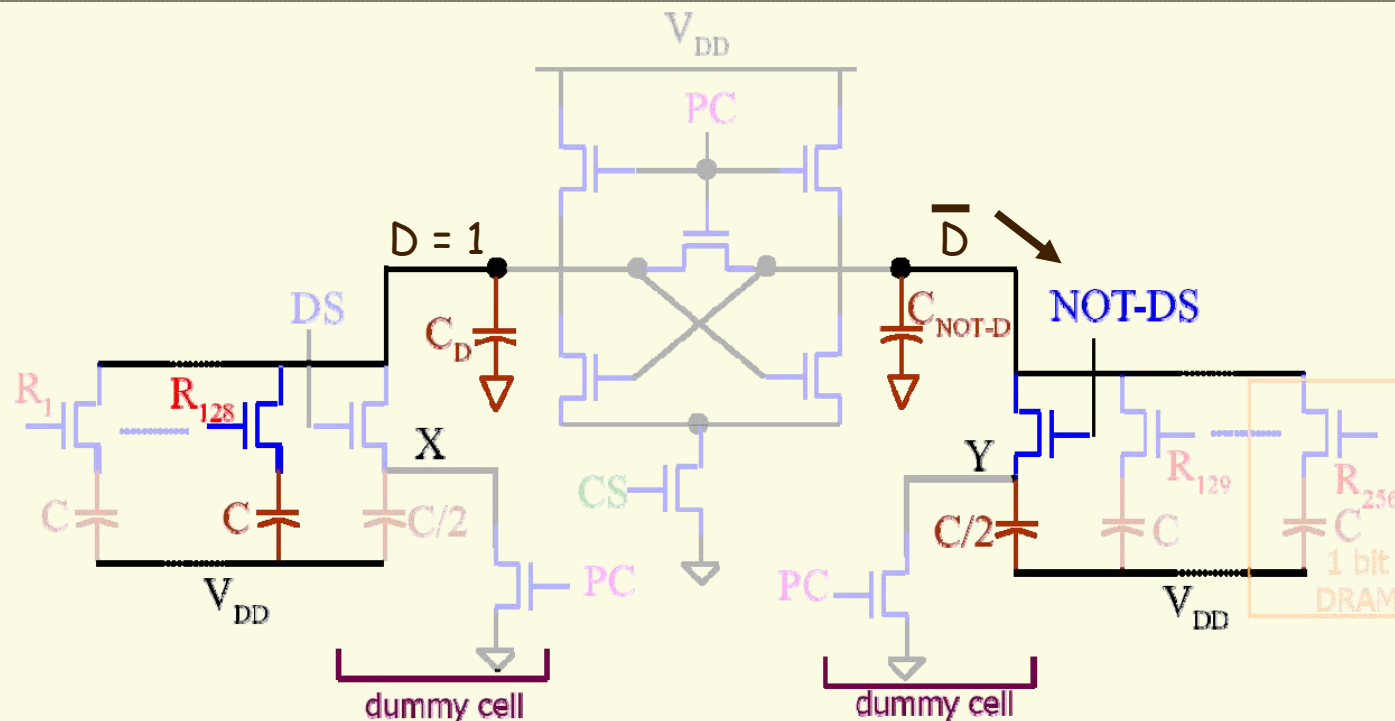
Sélection d'une ligne ( $R_i = 1$  avec  $0 < i < 256$ )  
Sélection de la « dummy cell » opposée :  
Ex :  $R_{128} = 1$  et NOT-DS = 1



## Phase 2 du rafraîchissement

## Cas 1 : Lecture d'un 1

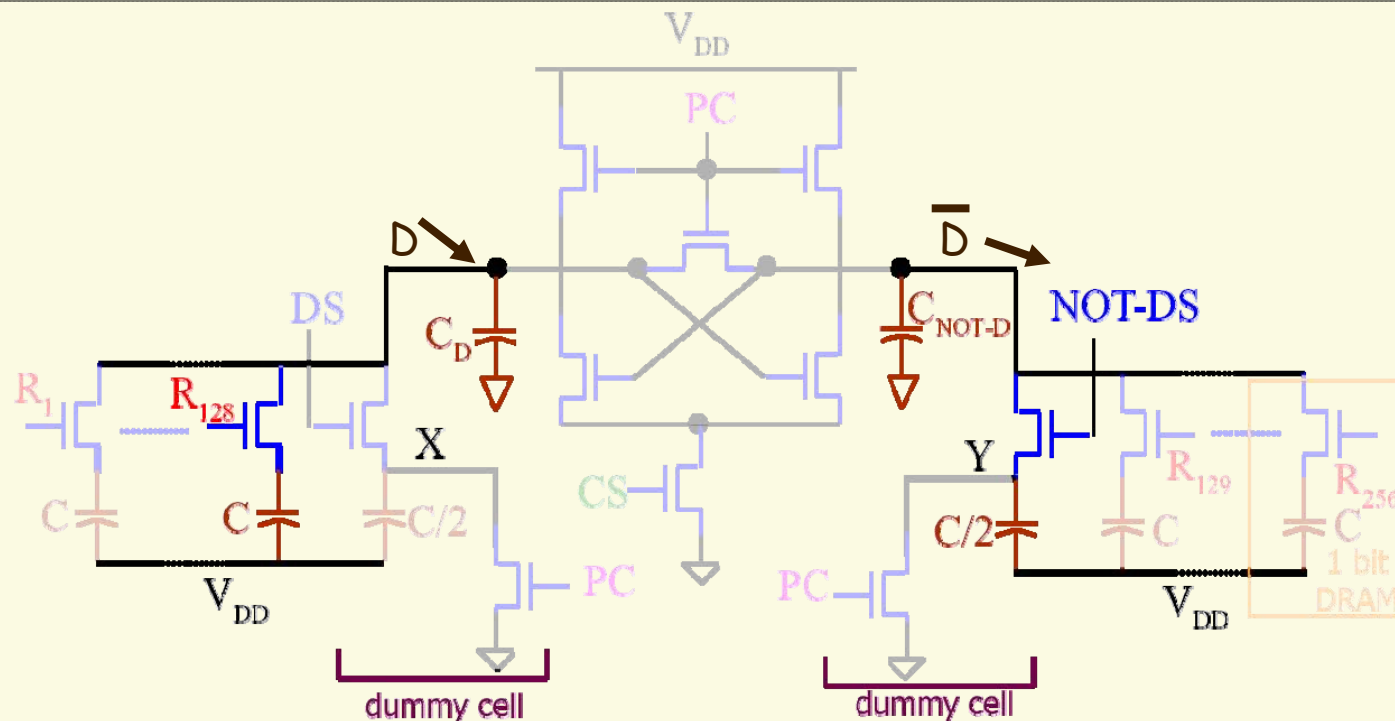
- ✓ Le potentiel en  $\underline{D}$  reste inchangé
- ✓ Le potentiel en  $\overline{D}$  chute (partage de charge entre  $C/2$  et  $C_{\text{NotD}}$ )



## Phase 2 du rafraîchissement

## Cas 1 : Lecture d'un 0

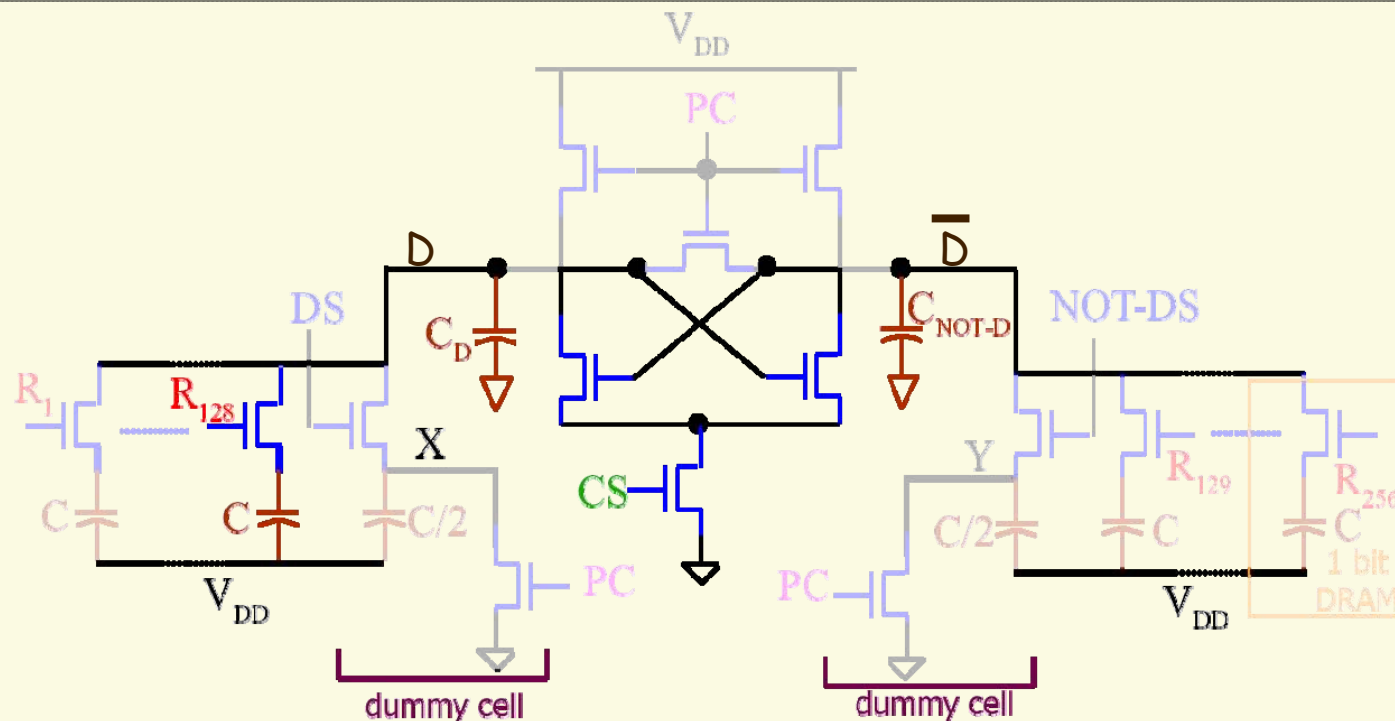
- ✓ Le potentiel en  $\underline{D}$  chute (partage de charge entre  $C$  et  $C_D$ )
- ✓ Le potentiel en  $\overline{D}$  chute aussi mais moins



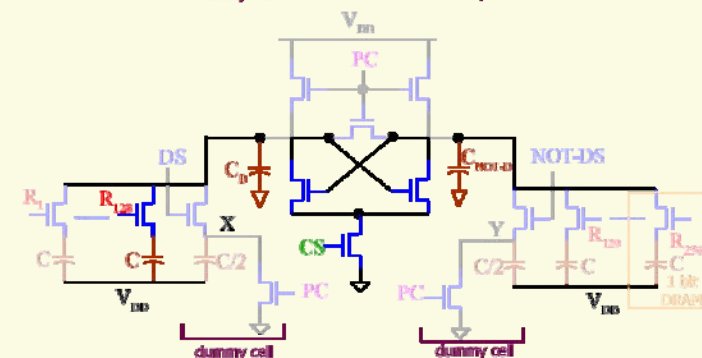
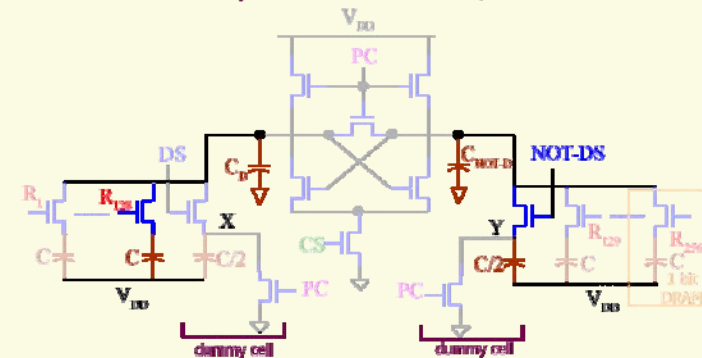
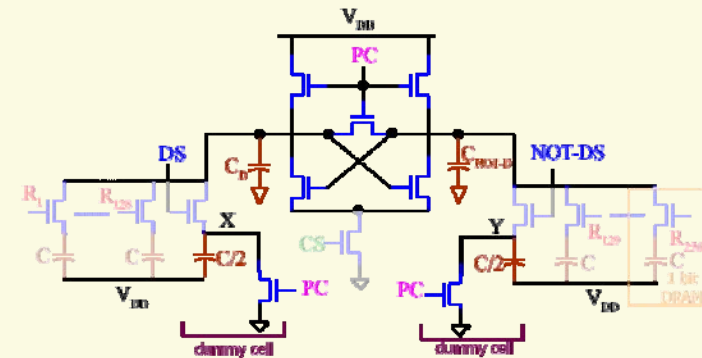
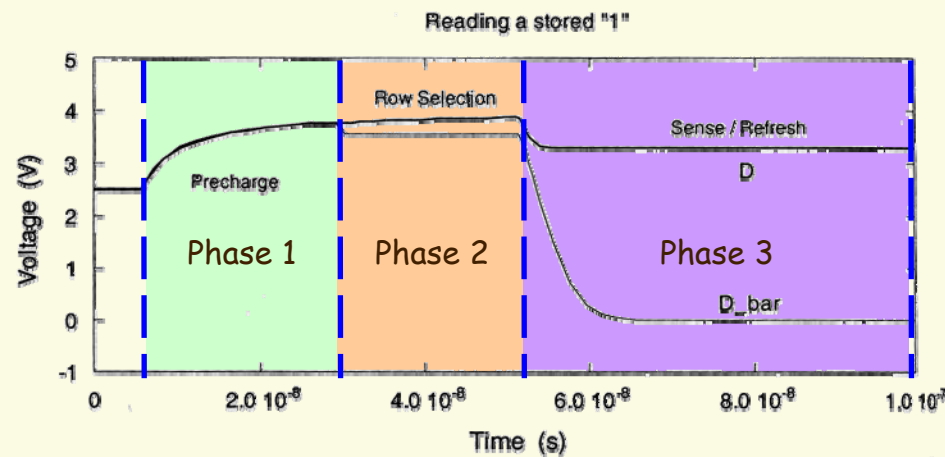
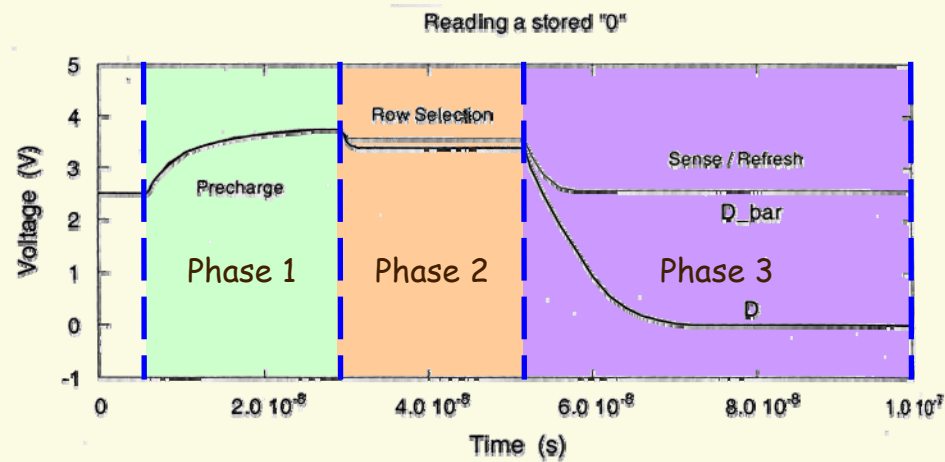
## Phase 3 du rafraîchissement

CS = 1 : Amplification différentielle de D et  $\bar{D}$

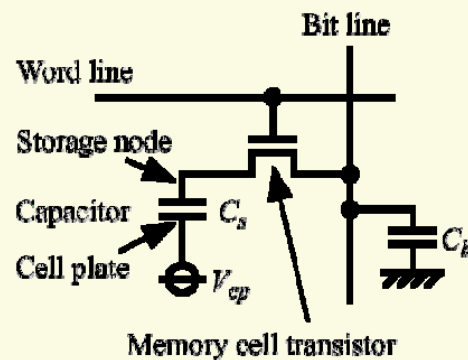
- ✓ D est forcé à 0 (1) et  $\overline{D}$  à 1 (0)
- ✓ La capacité associée à  $R_{128}$  se (dé)charge (car  $R_{128} = 1$ )



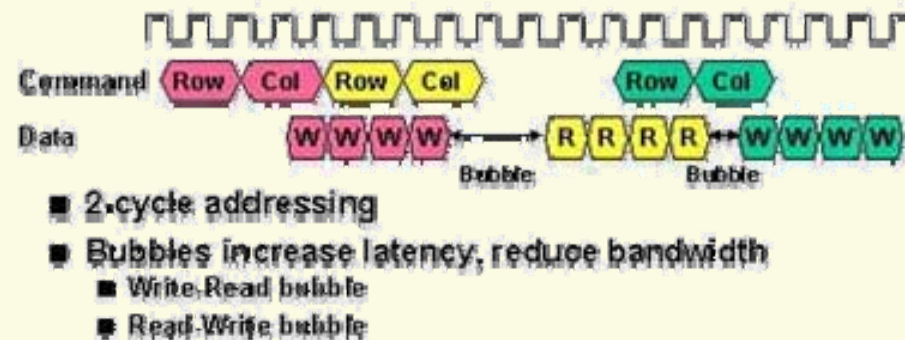
# Bilan de la lecture



# Bilan sur les mémoires DRAM



## PC100 SDRAM Protocol (32 Byte Xfer)



**Avantages** : Taille minimale, conception simple, technologie parfaitement maîtrisée

**Inconvénients** : lecture destructrice, courant de fuite, moins rapide que SRAM