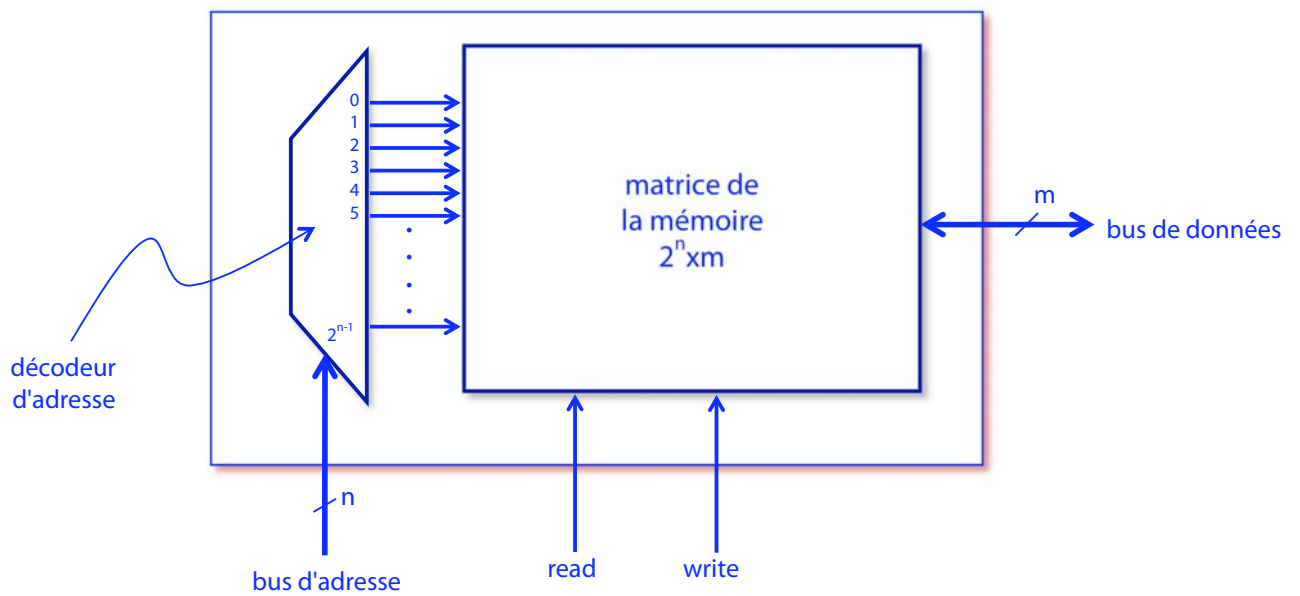


Les mémoires

Eduardo Sanchez
EPFL

Les mémoires

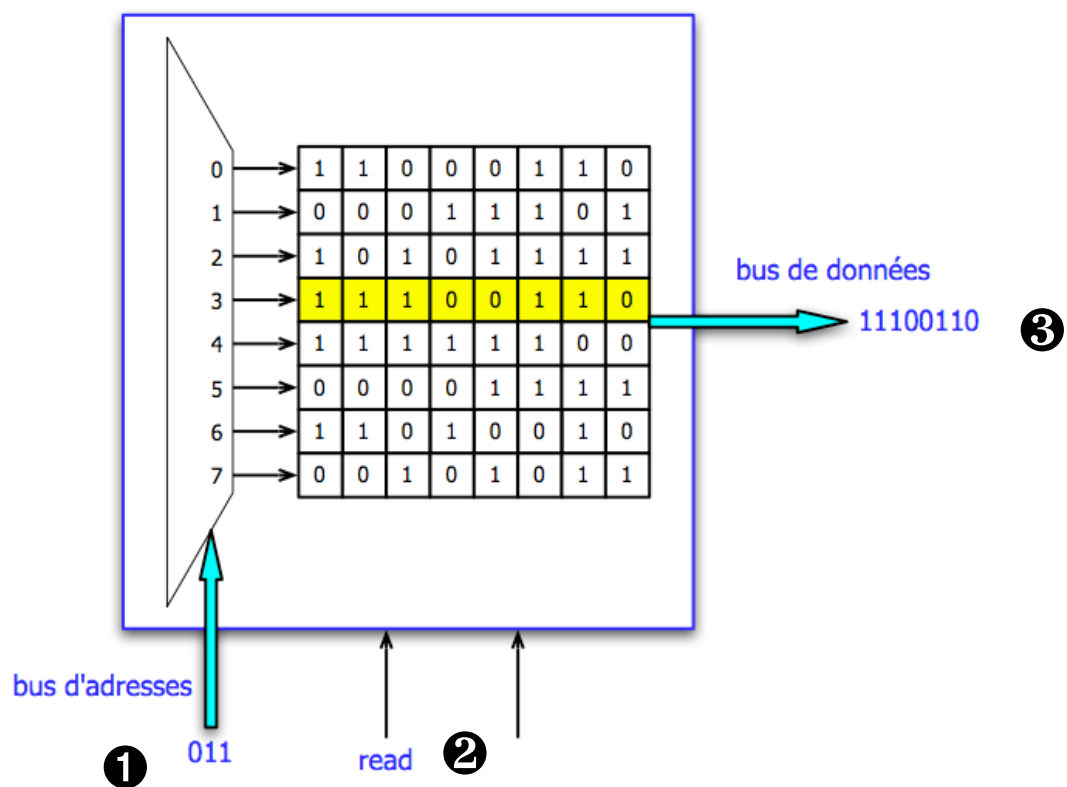
- Une mémoire est un élément de stockage d'information
- Les bits stockés sont organisés en forme de matrice: la dimension de la mémoire est donnée par le nombre de lignes fois la largeur de la ligne
- Chaque ligne de la mémoire est appelée un **mot**. Elle est identifiée par une **adresse** (numéro de la ligne)
- Le nombre de lignes est toujours une puissance de deux
- Deux opérations sont possibles, toujours sur un mot complet: la lecture (**read**) et l'écriture (**write**)



Eduardo Sanchez

3

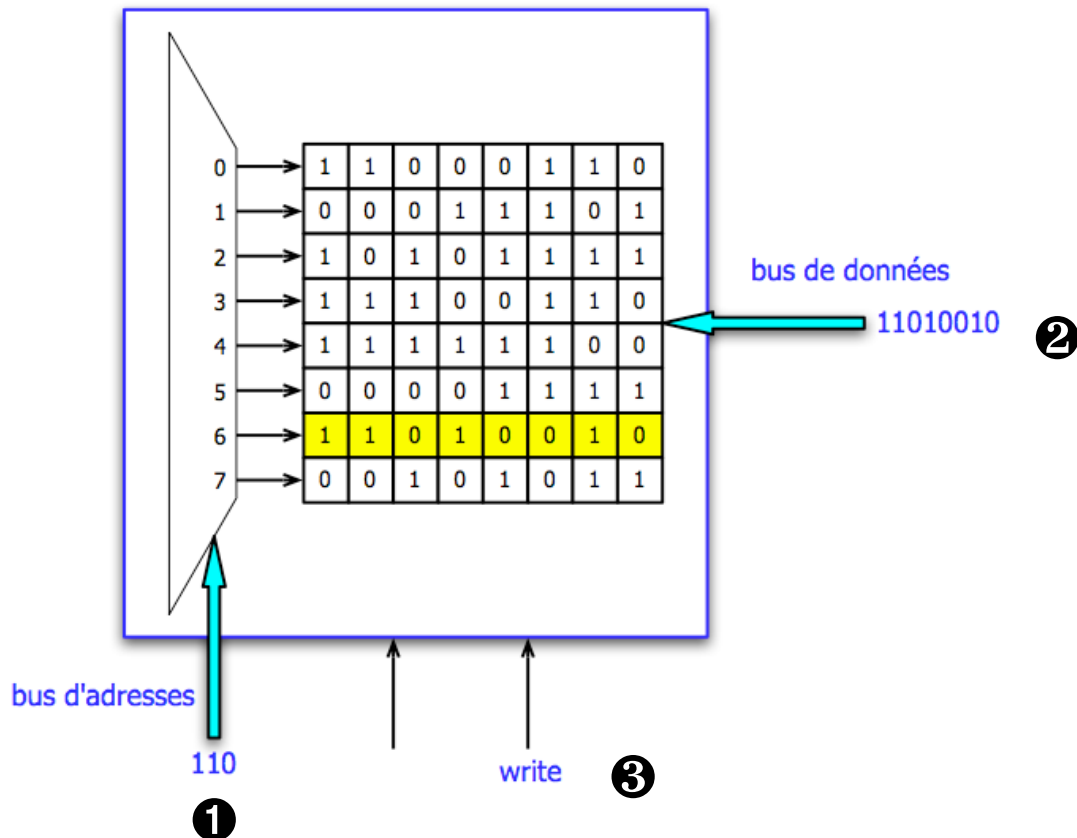
- Opération de lecture:



Eduardo Sanchez

4

- Opération d'écriture:



Eduardo Sanchez

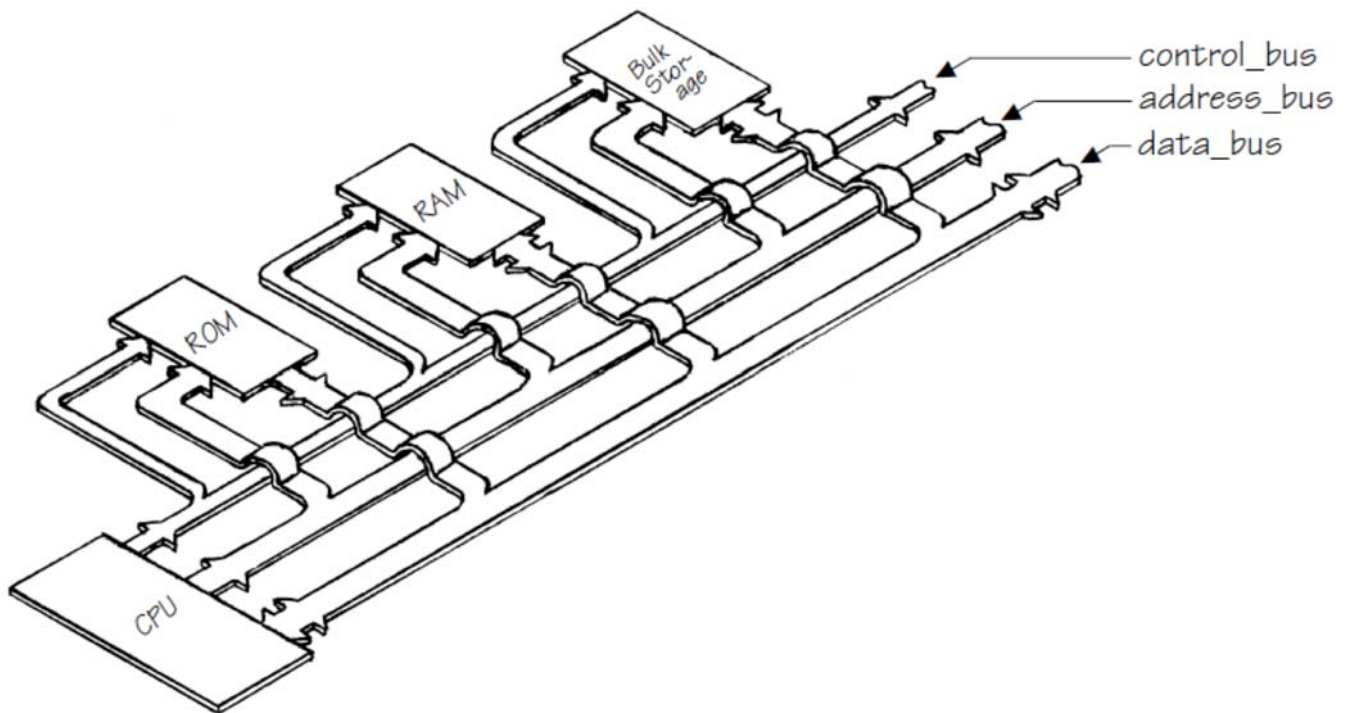
5

- Il existe deux grands types de mémoire:

- RAM** (*random-access memory*): mémoire volatile
 - statique (**SRAM**): l'information est conservée tant que la tension d'alimentation est présente
 - dynamique (**DRAM**): il faut recharger (rafraîchir) périodiquement les cellules de mémoire pour conserver les données
- ROM** (*read-only memory*): mémoire non volatile
 - mask**: le contenu est initialisé au moment de la fabrication et ne peut plus être modifié
 - PROM** (*programmable ROM*) ou **OTP ROM** (*one-time programmable*): le contenu peut être modifié une fois par l'utilisateur, à l'aide d'un équipement spécialisé
 - EPROM** (*erasable PROM*): le contenu peut être effacé et modifié plusieurs fois
 - UV EPROM**: le contenu est effacé par des rayons ultra-violets (plusieurs minutes)
 - EEPROM** (*electrically EPROM*) ou **E²PROM**: le contenu est effacé électriquement (quelques millisecondes)
 - Flash**: le contenu est effacé électriquement et plus rapidement que sur les EEPROM

Eduardo Sanchez

6

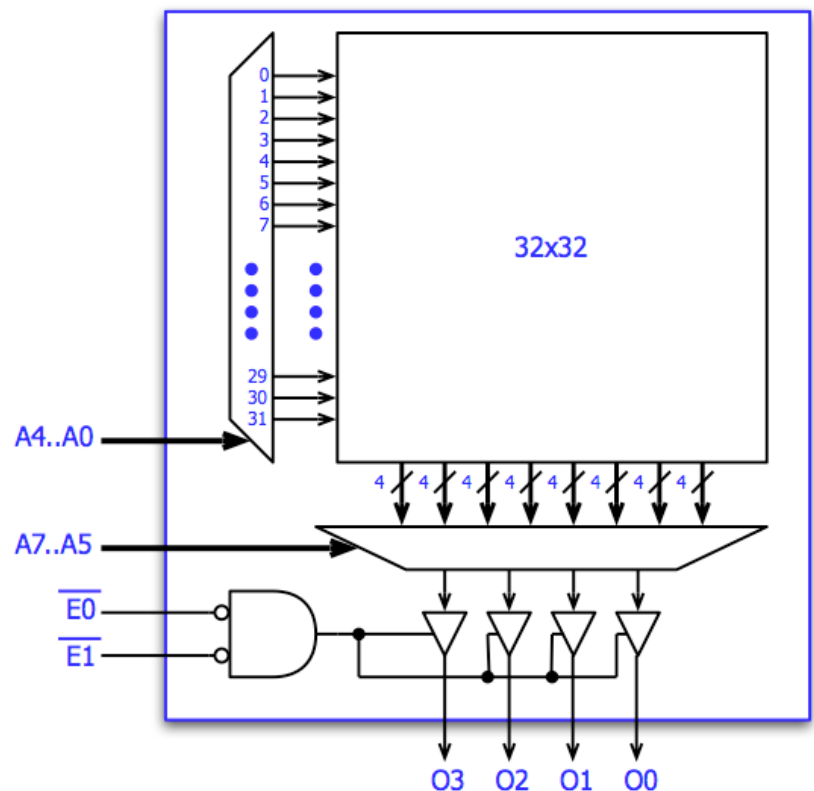


Eduardo Sanchez

7

Les mémoires ROM

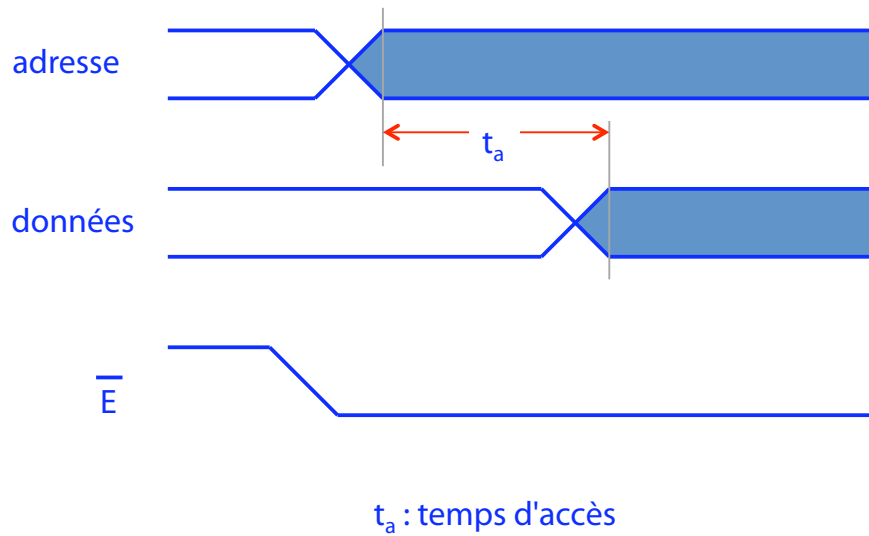
- Exemple: ROM 256x4



Eduardo Sanchez

8

- Cycle de lecture:



Eduardo Sanchez

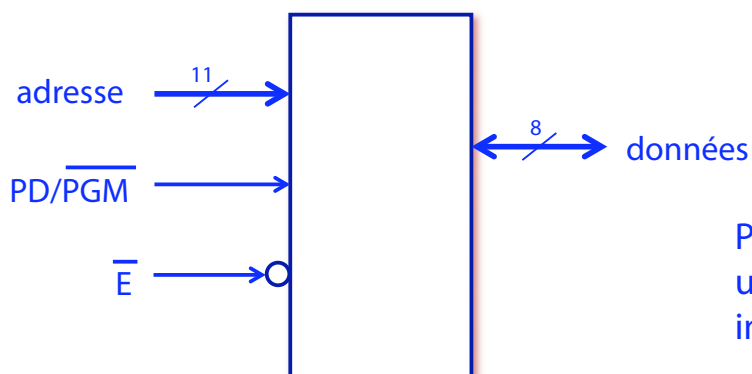
9

- PROM:

Une seule programmation est possible: elle est faite en brûlant des fusibles (un équipement particulier, le programmeur, est nécessaire). Par défaut, tous les bits sont initialement à 1 et l'on brûle les fusibles des bits qu'on doit programmer à 0

- EPROM:

Exemple: 2516 (2Kx8)

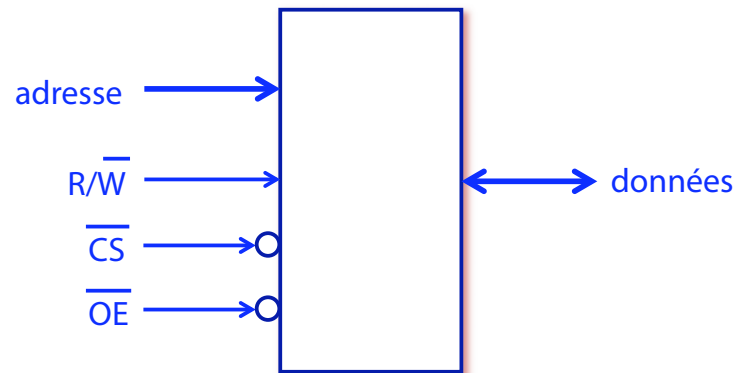


Pour programmer, il faut une tension $V_{pp}=25V$ et une impulsion sur l'entrée PD/PGM

Eduardo Sanchez

10

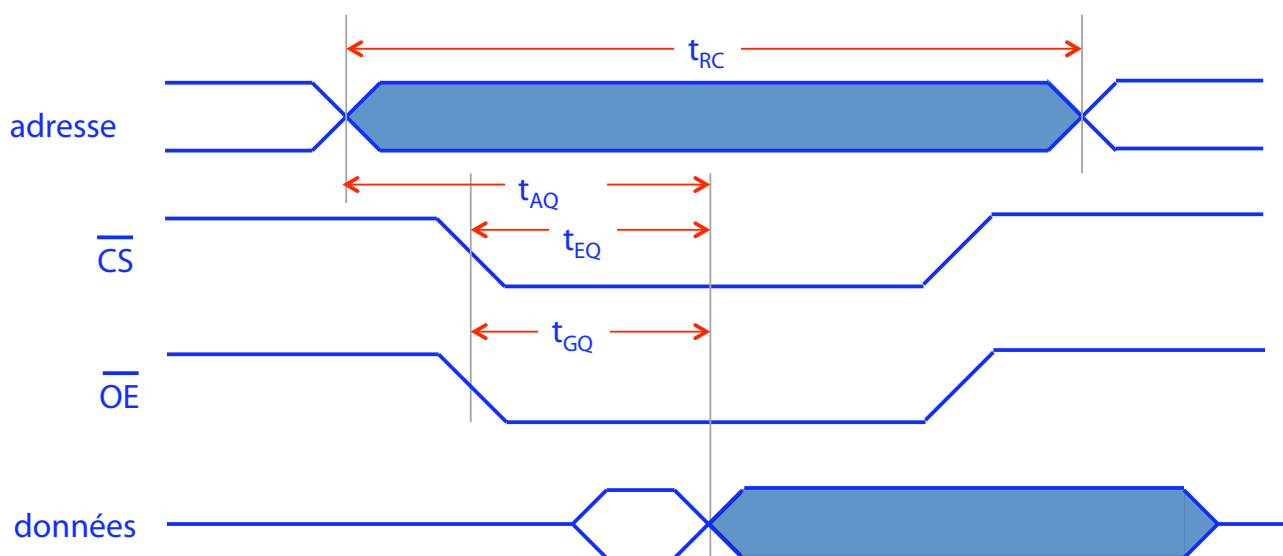
Les mémoires RAM statiques



Eduardo Sanchez

11

Cycle de lecture

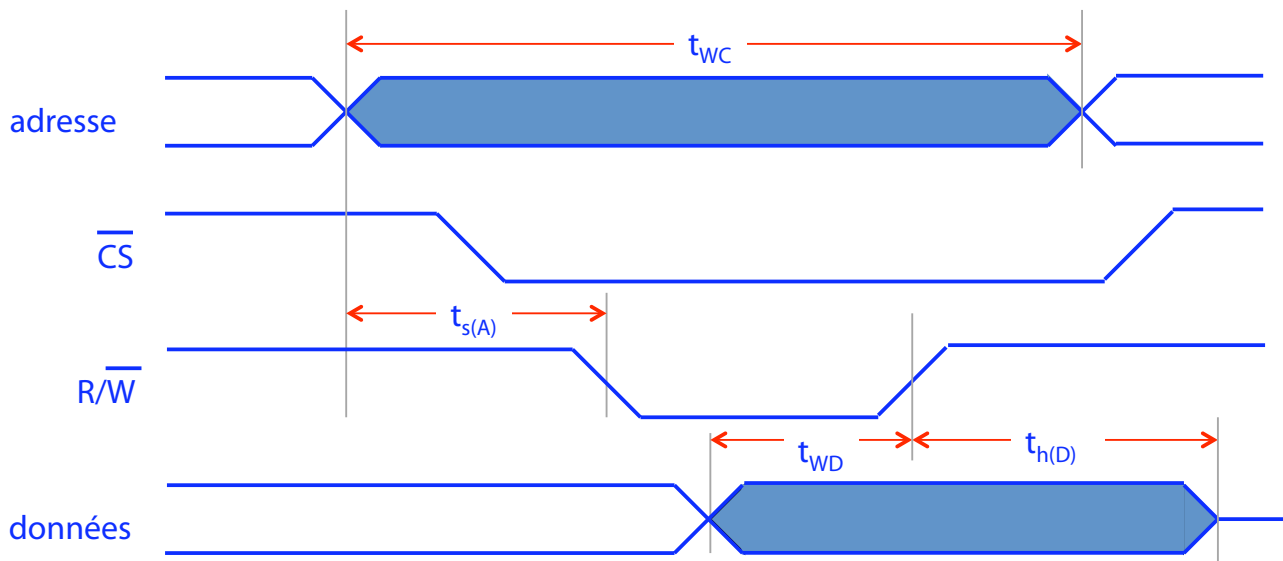


t_{RC} : temps du cycle de lecture

Eduardo Sanchez

12

Cycle d'écriture



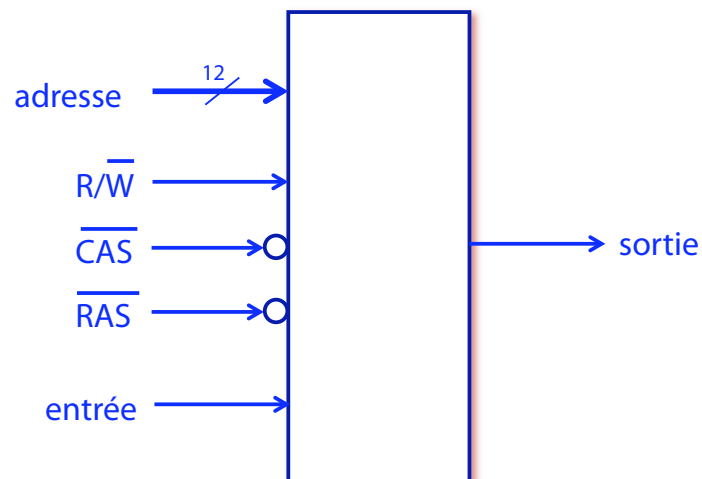
t_{WC} : temps du cycle d'écriture
 $t_{s(A)}$: address setup time
 $t_{h(D)}$: data hold time

Eduardo Sanchez

13

Les mémoires RAM dynamiques

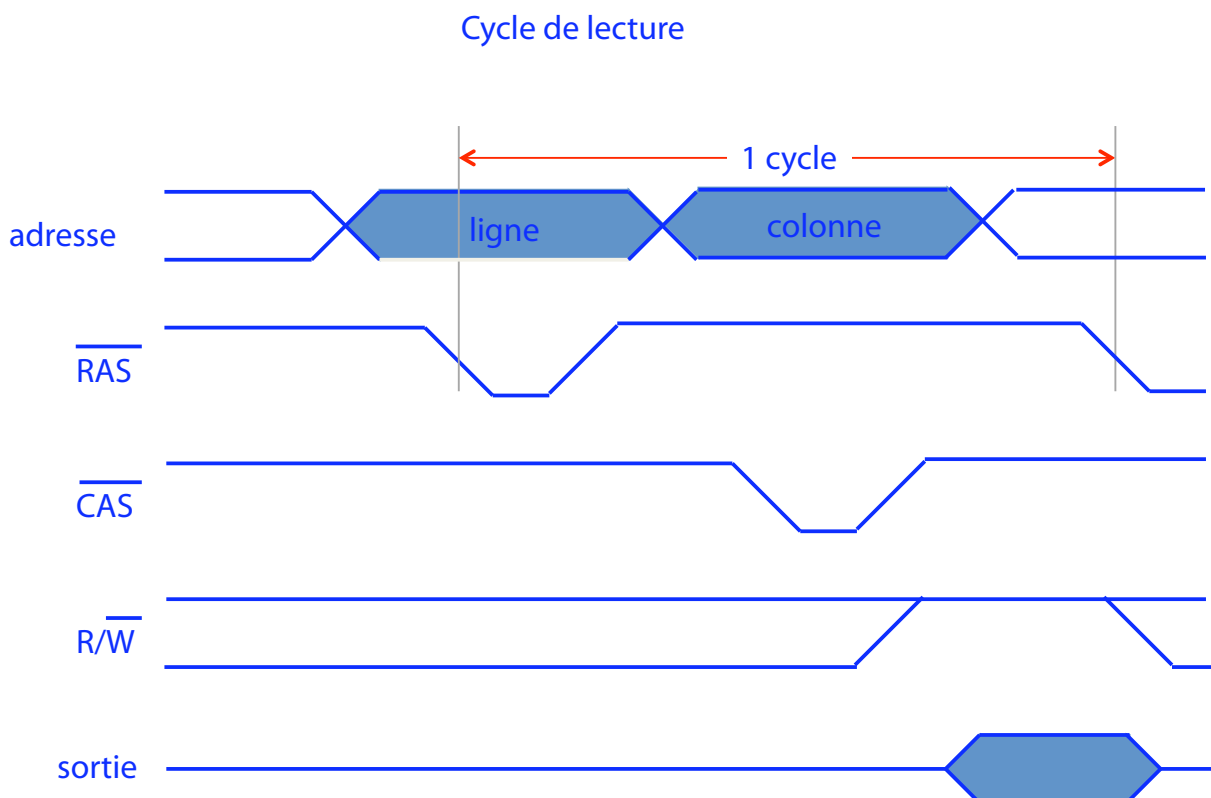
- Exemple: MCM516100 (16Mx1)



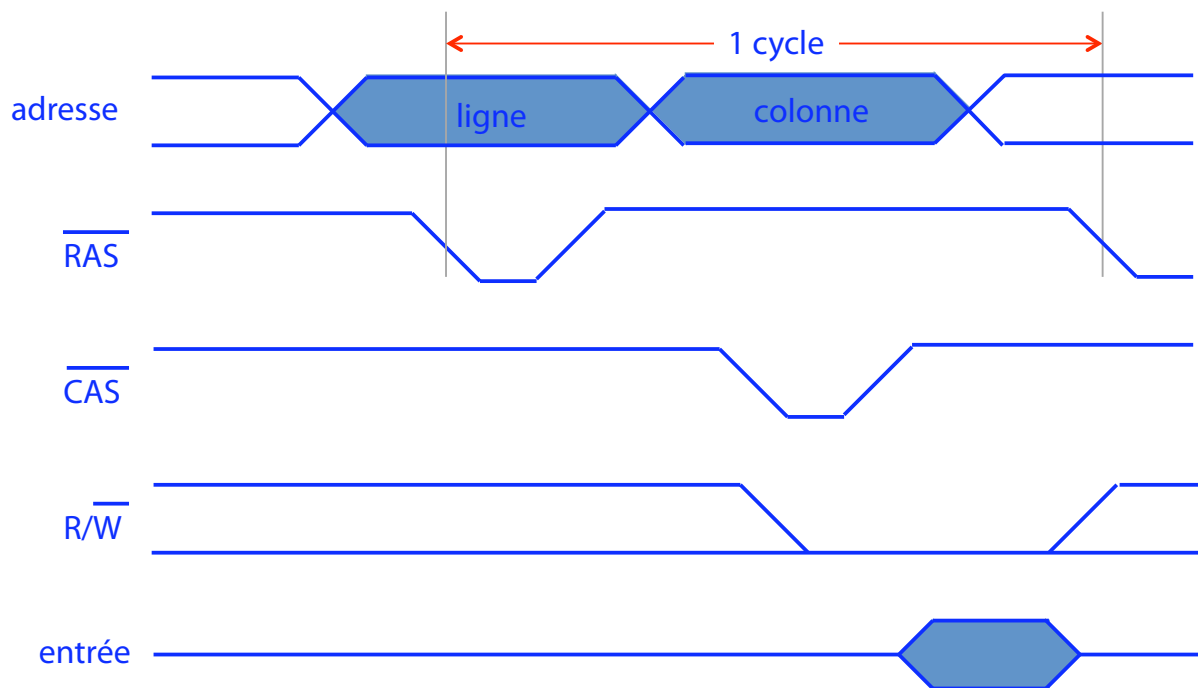
Eduardo Sanchez

14

- Un bit de donnée est stocké dans un condensateur (un latch est utilisé dans les SRAM). Une cellule de DRAM est donc très simple, ce qui permet la construction de très grandes mémoires à bas prix
- Par contre, le condensateur se décharge avec le temps et l'information est perdue si la charge n'est pas rafraîchie périodiquement
- Pour réduire le nombre de lignes d'adresse, les DRAMs utilisent un multiplexage de l'adresse: les bits d'adresse de la ligne et de la colonne sont envoyés en séquence.
 Tout d'abord, les bits d'adresse de la ligne sont envoyés, et stockés internement grâce au signal **RAS** (*row address strobe*). Ensuite, les bits d'adresse de la colonne sont envoyés, identifiés par le signal **CAS** (*column address strobe*)



Cycle d'écriture



Eduardo Sanchez

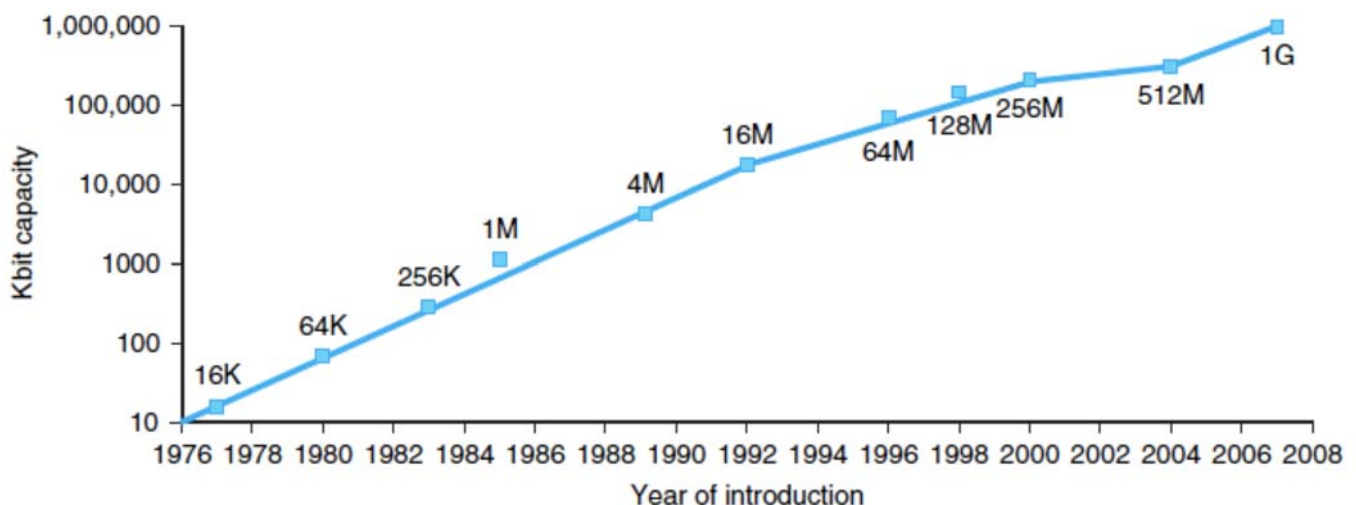
17

- Les opérations (lecture ou écriture) sur des données situées dans la même colonne (page) peuvent être accélérées avec des cycles en **mode page**: le signal **RAS** reste active pendant que le signal **CAS** change pour chaque colonne
- Généralement, une DRAM doit être rafraîchie chaque 8 à 16 ms. Une opération de lecture rafraîchit automatiquement toutes les adresses de la ligne sélectionnée. Mais, bien entendu, il est impossible de dépendre des cycles de lecture et il faut implémenter des cycles spéciaux de rafraîchissement

Eduardo Sanchez

18

- Il existe deux grands modes de rafraîchissement:
 - *burst refresh*: toutes les lignes de la mémoire sont rafraîchies consécutivement à chaque période de rafraîchissement. Toutes les opérations normales de lecture et d'écriture sont arrêtées pendant le cycle de rafraîchissement
 - *distributed refresh*: chaque est rafraîchie à intervalles réguliers, entre des opérations normales de lecture et d'écriture. Si, par exemple, une mémoire possède 1024 lignes et une période de rafraîchissement de 8 ms, chaque ligne doit être rafraîchie toutes les $8\text{ms}/1024 = 7.8\mu\text{s}$
- Il y a deux types d'opérations de rafraîchissement:
 - *RAS-only refresh*: le signal **RAS** est mis actif pour stocker l'adresse de la ligne à rafraîchir, mais le signal **CAS** reste inactif pendant tout le cycle. Un compteur externe est utilisé pour garder trace de la ligne rafraîchie
 - *CAS before RAS refresh*: ce cycle est commencé par une activation du signal **CAS** avant l'activation du signal **RAS**. L'adresse de la ligne à rafraîchir est générée par un compteur interne



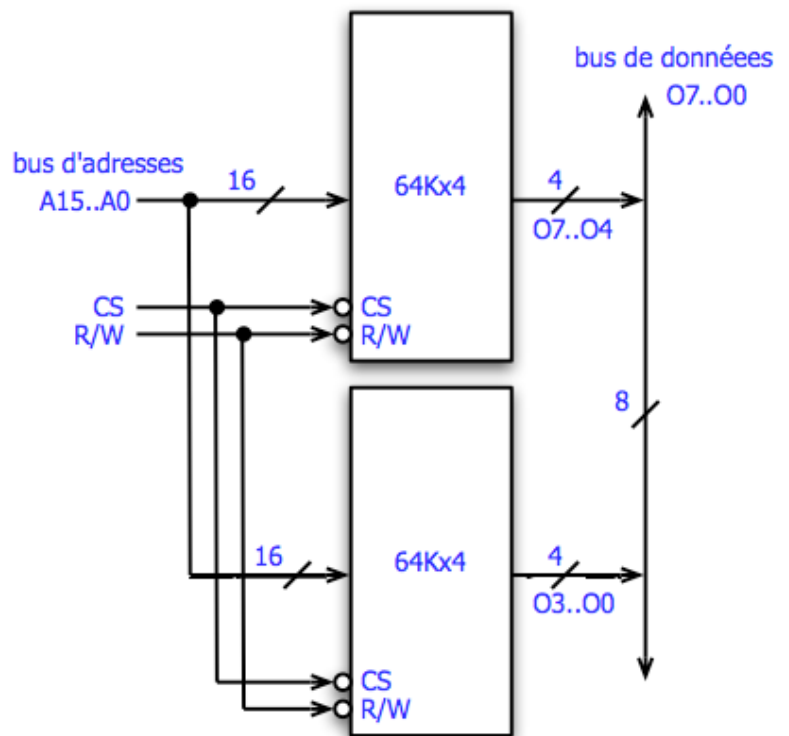
Les mémoires Flash

- Les EPROMs sont des mémoires non volatiles rapides, à grande densité et bas prix. Toutefois, elles possèdent un grand désavantage pour certain type d'applications: il faut les enlever du système pour les effacer et les reprogrammer
- Les EEPROMs ne présentent pas ce problème mais, par contre, possèdent une densité inférieure et un prix plus élevé
- Les mémoires Flash présentent un bon compromis: une densité beaucoup plus grande que les EEPROMs, à un coût nettement inférieur (bien que pas aussi bas que les EPROMs), tout en étant effaçables électriquement
- Le seul désavantage des mémoires Flash est que l'effacement n'est possible que dans sa totalité ou par secteurs, mais pas byte à byte

- Trois opérations sont possibles dans une mémoire Flash:
 - programmation: initialement, tous les bits sont à 1. Pour programmer un bit à 0, il faut lui appliquer une tension. Une fois programmée, la mémoire peut garder son contenu pendant 100 ans, sans besoin d'alimentation électrique
 - lecture: similaire à la lecture d'une RAM statique
 - effacement: la mémoire Flash doit être effacée avant d'être reprogrammée
- Il est aujourd'hui courant d'utiliser des mémoires Flash en tant que disques durs

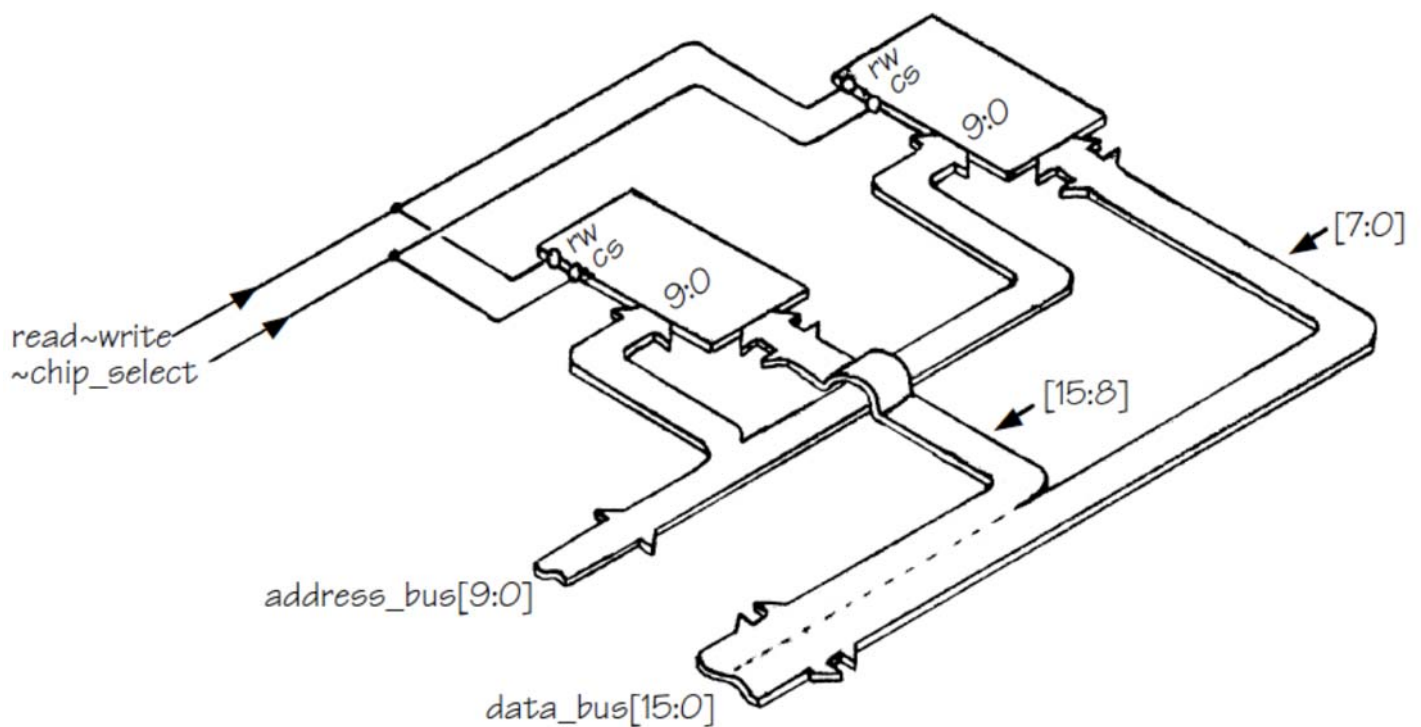
Extension de capacité des mémoires

- Exemple: utilisation de deux RAM 64Kx4 pour obtenir une mémoire 64Kx8



Eduardo Sanchez

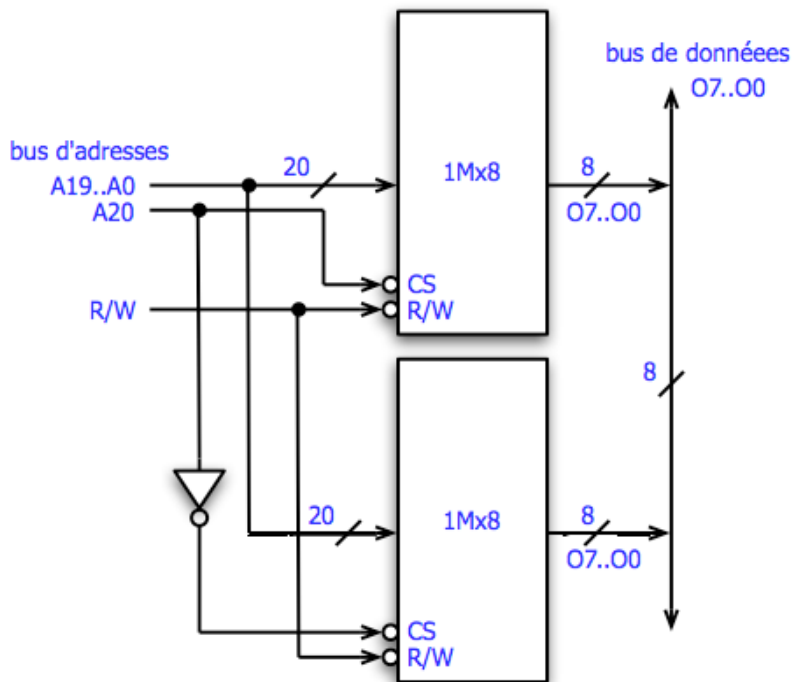
23



Eduardo Sanchez

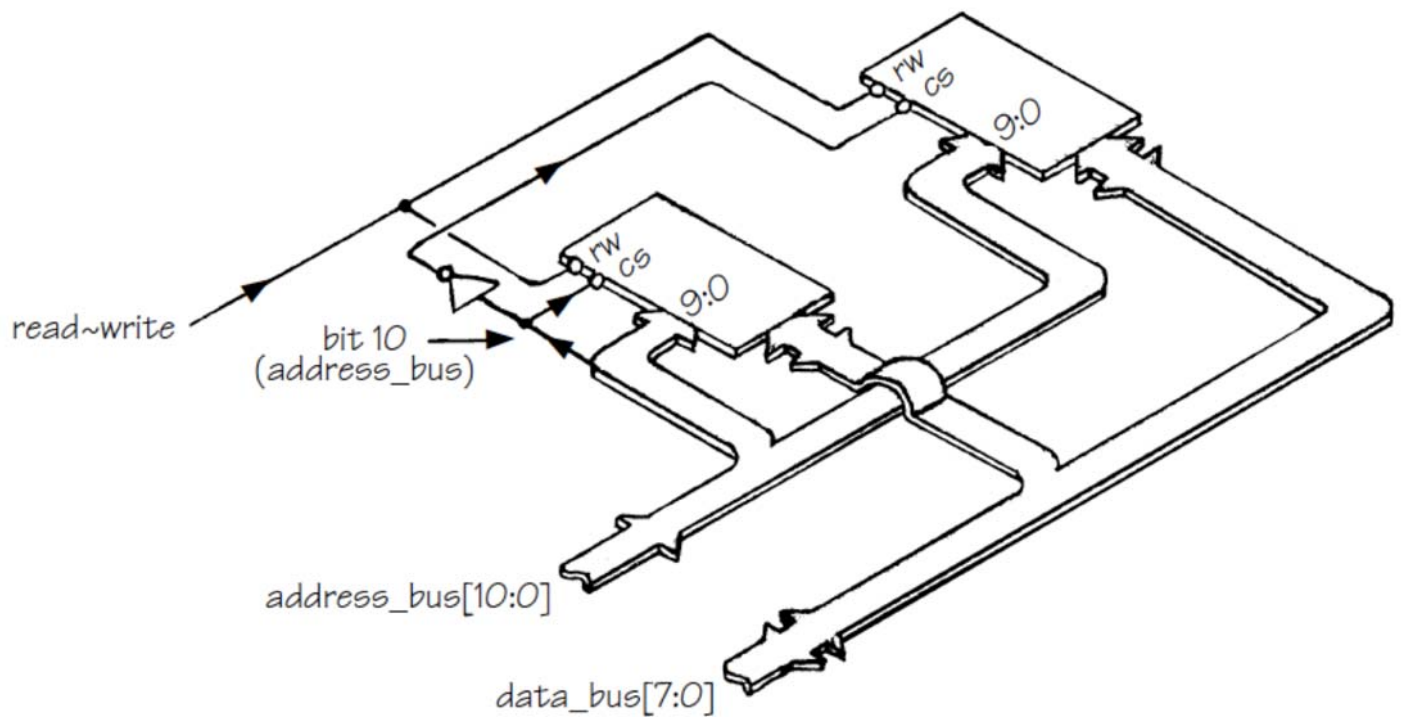
24

- Exemple: utilisation de deux RAM 1Mx8 pour obtenir une mémoire 2Mx8



Eduardo Sanchez

25

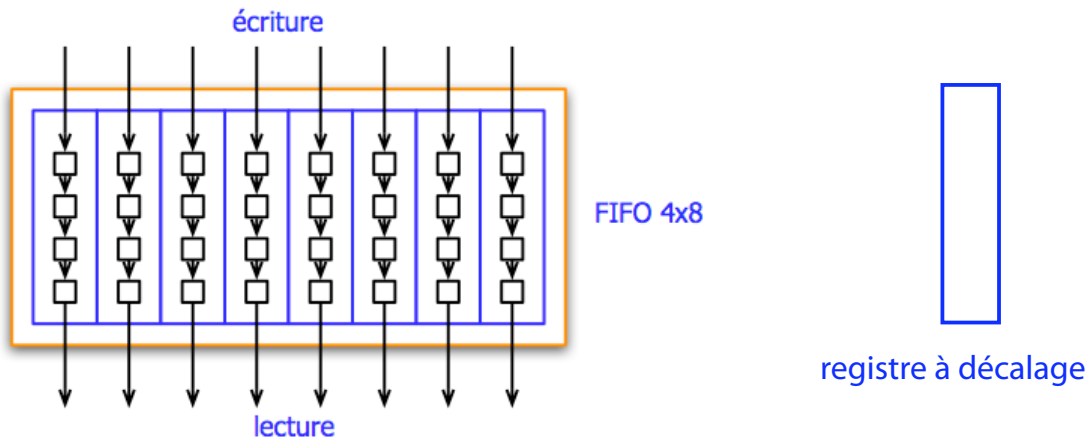


Eduardo Sanchez

26

Mémoires FIFO

- Les mémoires FIFO (*First In - First Out*) sont un type particulier de mémoires RAM, où les mots ne sont pas adressables: les mots sont lus dans le même ordre dans lequel ils sont écrits (le premier écrit est le premier lu)
- Une FIFO 4x8, par exemple, peut être vue comme un ensemble de 8 registres à décalage, de 4 bits chacun



Eduardo Sanchez

27

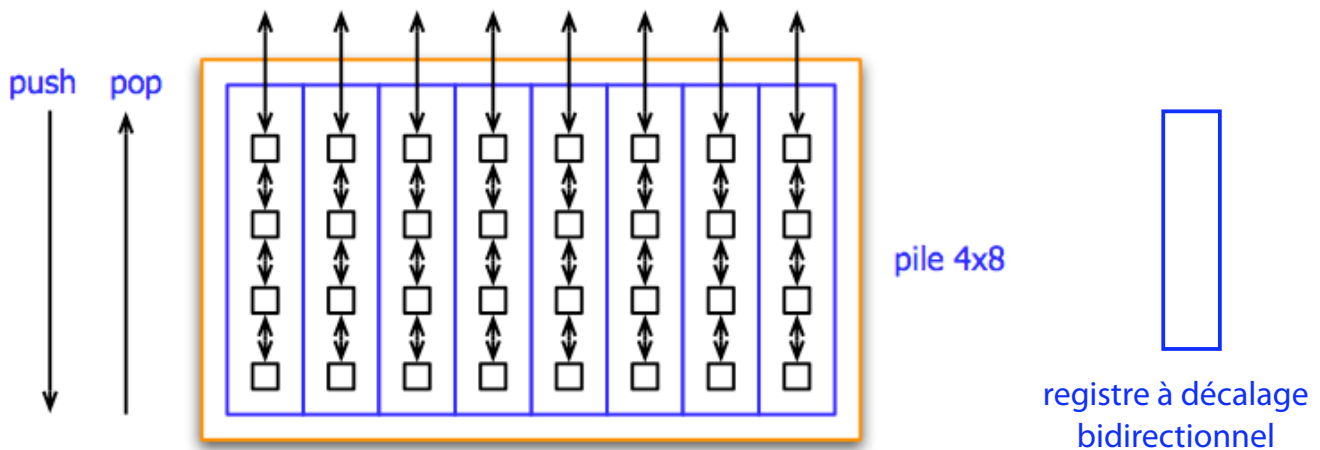
Mémoires LIFO

- Les mémoires LIFO (*Last In - First Out*), appelées également pile (*stack*), sont un autre type de mémoire non adressable, avec deux opérations possibles:
 - push: équivalent à l'écriture, met une information au sommet de la pile, en poussant vers le bas les informations déjà présentes dans la pile. L'information tout en bas de la pile sera perdue
 - pop: équivalent à la lecture, prend l'information qui se trouve au sommet de la pile, en poussant tout le contenu de la pile vers le sommet. L'information lue sera enlevée de la pile

Eduardo Sanchez

28

- Une pile 4x8, par exemple, peut être vue comme un ensemble de 8 registres à décalage, de 4 bits chacun. Mais contrairement aux FIFO, les registres à décalage sont bidirectionnels et on accède seulement au registre en haut (le sommet de la pile)



Eduardo Sanchez

29

- Les ordinateurs utilisent couramment des piles pour le stockage de l'information. Dans ces cas, la pile n'est pas implémentée à l'aide des registres, mais utilise une partie de la mémoire RAM de l'ordinateur
- Dans les piles à RAM, l'information ne bouge pas, n'est pas décalée vers le bas de la pile, mais ce qui bouge est le sommet de la pile, indiqué par un pointeur: le *stack pointer* (SP)
- Par convention arbitraire, lors d'un push le stack pointer est décrémenté (le sommet de la pile est déplacé vers les adresses basses de la mémoire), tandis que lors d'un pop le stack pointer est incrémenté (le sommet de la pile est déplacé vers les adresses hautes de la mémoire)

Eduardo Sanchez

30

- Exemple: supposons que la pile d'un ordinateur se trouve initialement dans l'état montré à droite, à l'intérieur de sa mémoire RAM 64Kx8
- Après une opération push de la donnée 0x74FA, la pile reste dans l'état montré à droite
- Nous avons supposé un ordinateur big-endian

