

LICENCE
EEA

5 - Mémoires

SYSTEMES NUMERIQUES & PROCESSEURS EMBARQUES

Previously on LU3EE100...

iPad

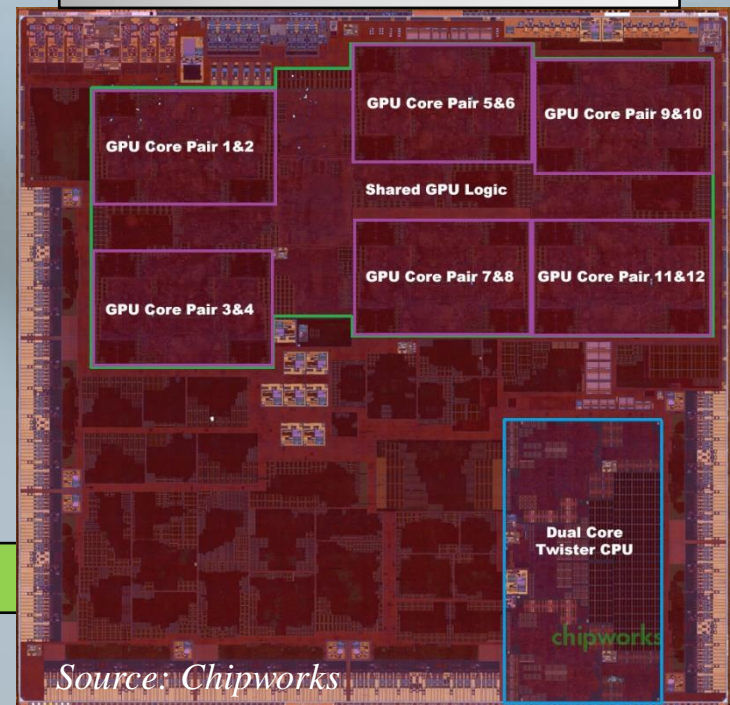


Source: Apple

C5

2

- 2 processeurs ARM
- Mémoire
- Périphériques (video)



Source: Chipworks

Processeur A9X

L3 EEA – LU3EE100

Plan

- Définitions – Généralités - Architecture
- Technologies mémoires RAM
 - SRAM
 - DRAM
- Technologies mémoires ROM
 - ROM
 - PROM
 - EPROM
- Technologies mémoires Flash
- Technologies mémoires MRAM

C5

3

Classification Mémoires

- Historiquement, les mémoires étaient classées selon les termes
 - RAM (Random Access Memory)
 - Mémoire Vive (Volatile)
 - ROM (Read Only Memory)
 - Mémoire Morte (Non Volatile)
- Random access → Accès aléatoire
 - Accès libre à n'importe quelle case de la mémoire
 - Par opposition à accès séquentiel
(Ex: Stockage sur une bande magnétique)

C5

4

Classification Mémoires

- Historiquement, les mémoires étaient classées selon les termes
 - RAM (Random Access Memory)
 - Mémoire Vive (Volatile)
 - ROM (Read Only Memory)
 - Mémoire Morte (Non Volatile)
- Volatile/Non volatile
 - Volatile:
La mémoire perd ses données si elle n'est pas alimentée
 - Non volatile:
La mémoire garde ses données, même sans alimentation

C5

5

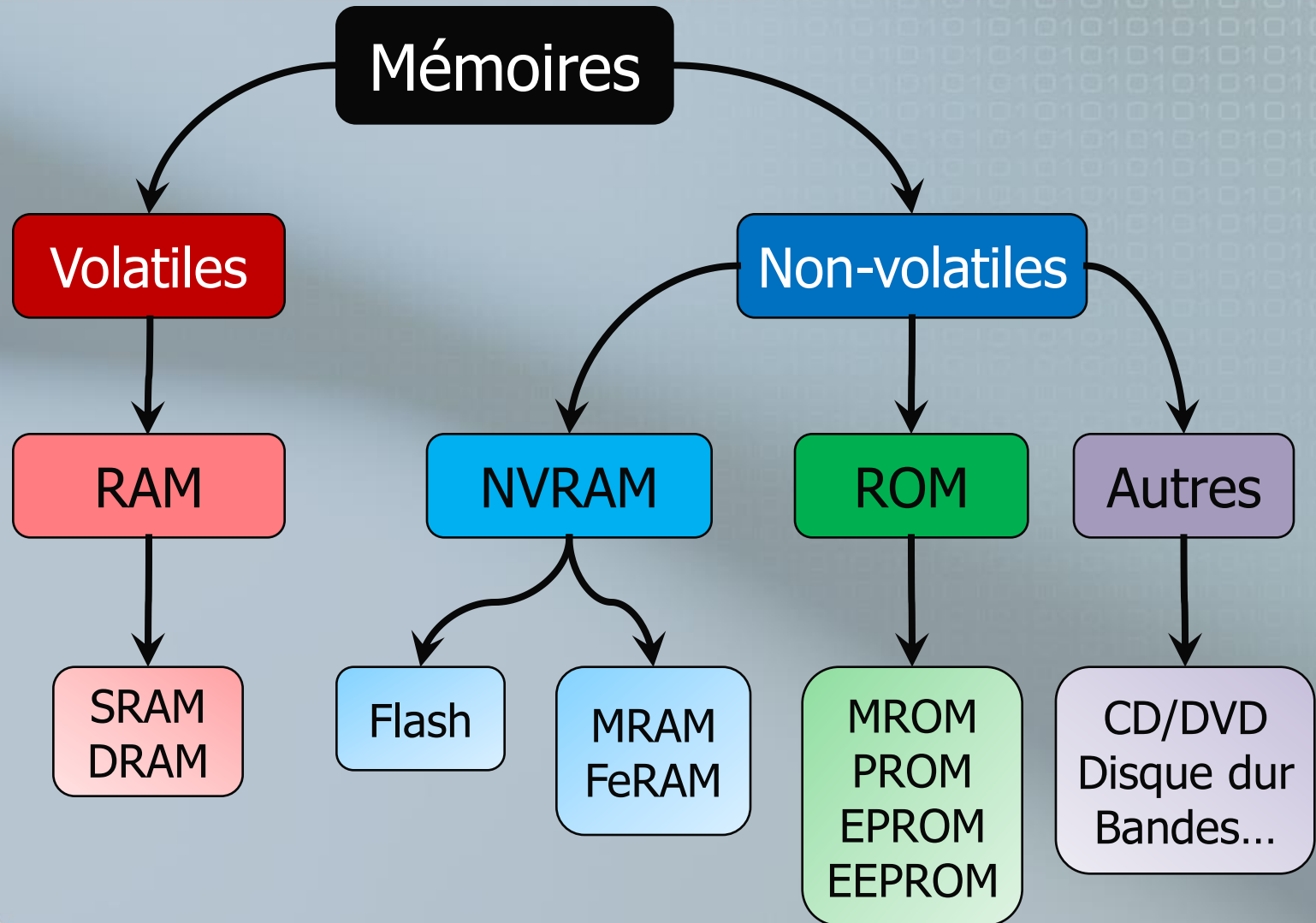
Classification Mémoires

- Historiquement, les mémoires étaient classées selon les termes
 - RAM (Random Access Memory)
 - Mémoire Vive (Volatile)
 - ROM (Read Only Memory)
 - Mémoire Morte (Non Volatile)
- Oui, mais...
 - Les ROM sont aussi des mémoires à accès aléatoire
 - On peut écrire des données dans des ROM
 - Cas des mémoires Flash (clés USB)

C5

6

Classification Mémoires



C5

7

Abréviations

- **RAM: Random Access Memory**

- SRAM: Static RAM
- DRAM: Dynamic RAM

- **NVRAM: Non Volatile RAM**

- MRAM: Magnetic RAM
- FeRAM: Ferroelectric RAM

- **ROM: Read Only Memory**

- MROM: Mask ROM
- PROM: Programmable ROM
- EPROM: Erasable PROM
- EEPROM: Electrically Erasable PROM

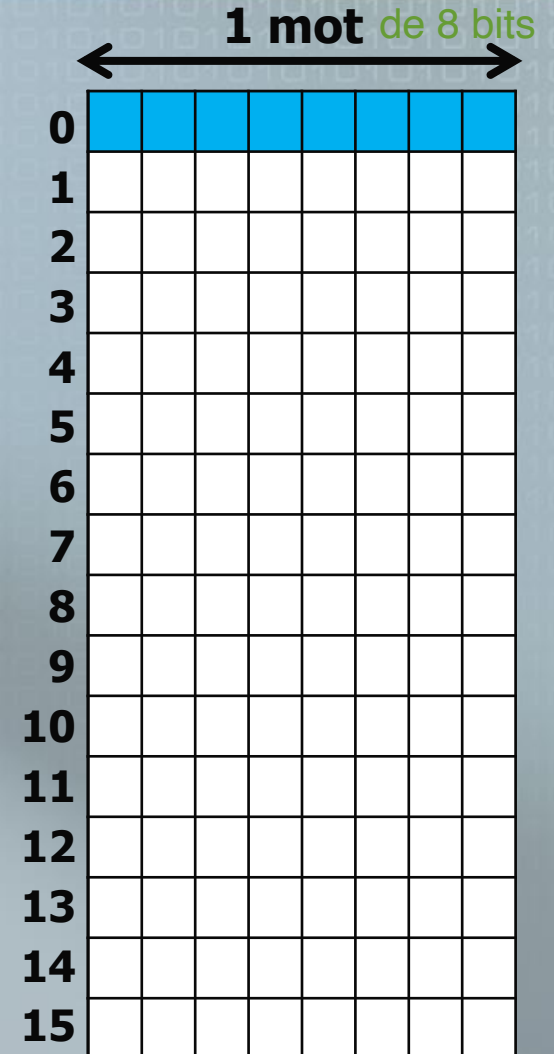
C5

8

Généralités

■ Données manipulées

- La mémoire est organisée en mots (word)
 - Selon le composant
1 mot = 8, 16, 32, 64, (...) bits
 - Fixe la taille du bus de données du composant
- 1 mot est rangé dans une case de la mémoire
 - Chaque case mémoire est identifiable par son adresse



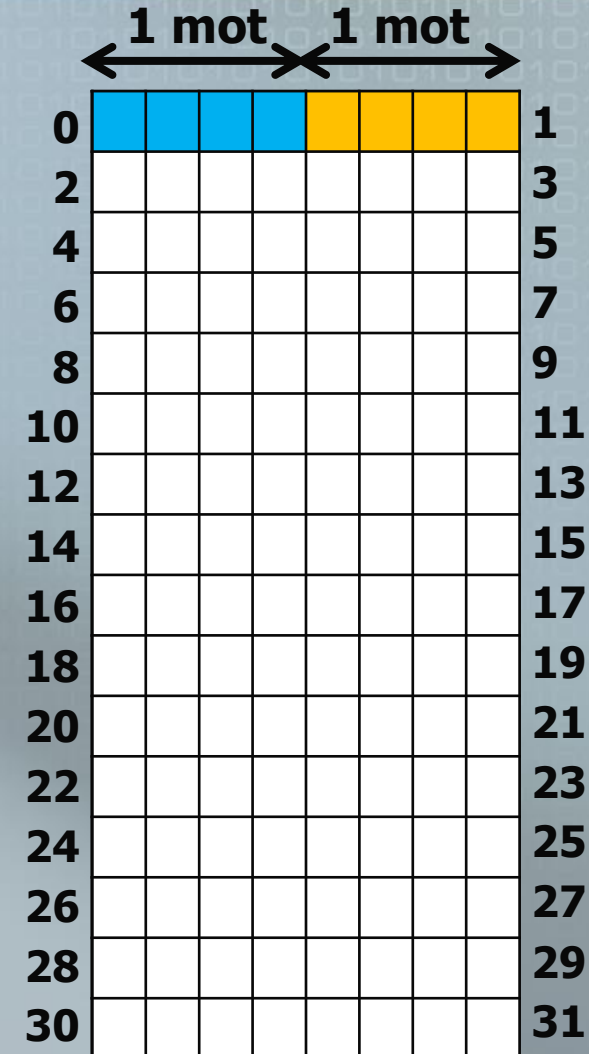
16 cases dont chaque case contient 1 mot de 8 bits

Généralités

plusieurs mots sur une même ligne

■ Organisation du Tableau Mémoire

- On peut très bien ranger plusieurs mots par ligne du tableau.
- La répartition des adresses permet d'identifier chaque sous-bloc d'une ligne



C5

10

Généralités

■ Capacité

- Exprimée en octets (Bytes, B) ou en bits (b)

Capacité	Valeur (en octets)	Valeur (en bits)
1 ko kilo octets	2^{10} 1024	2^{13} (8 kbits) 8192
64 ko	2^{16} 65536	2^{19} (512 kbits)
1 Mo	2^{20}	2^{23} (8 Mbits)
1 Go	2^{30}	2^{33} (8 Gbits)
4 Go	2^{32}	2^{35} (32 Gbits)

- Exemple: Mémoire de 64 Mégaoctets (Mo)
 - Si 1 mot = 32 bits (4 octets)
 - 16 Méga-mots (ou Méga-cases) dans la mémoire

Généralités

■ Performance

■ Critères Fonctionnels

- Capacité: Taille tableau mémoire, taille des mots...
- Latence : Durée opération mémoire (lecture/écriture)
- Débit : Nombre de bits transférables par seconde
- Configurabilité: Réglage latence/débit

■ Critères Non-Fonctionnels

- Surface du circuit
 - Va dépendre de la technologie utilisée
- Consommation
 - Coût énergétique d'une opération mémoire
- Alimentation
 - Tension minimale d'alimentation
 - Possibilité d'effectuer du Dynamic Voltage Scaling

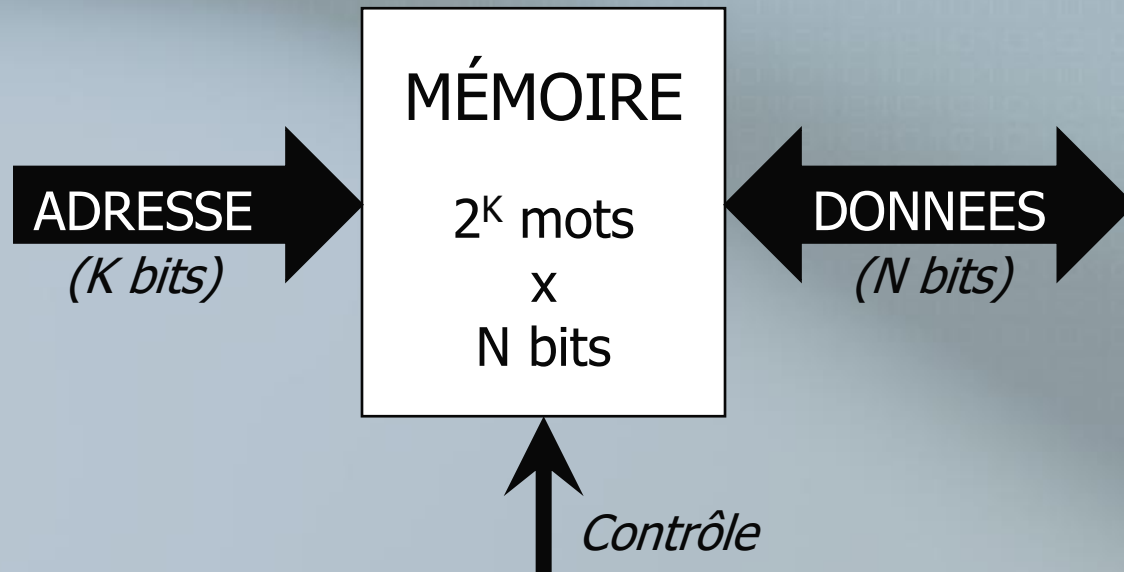
C5

12

Généralités

■ Accès (lecture/écriture)

- Bus d'adresse → Choix de la case mémoire (décodage)
- Bus de données → Pour envoyer ou récupérer la donnée
- Bits de contrôle → Read/write, chip select...



C5

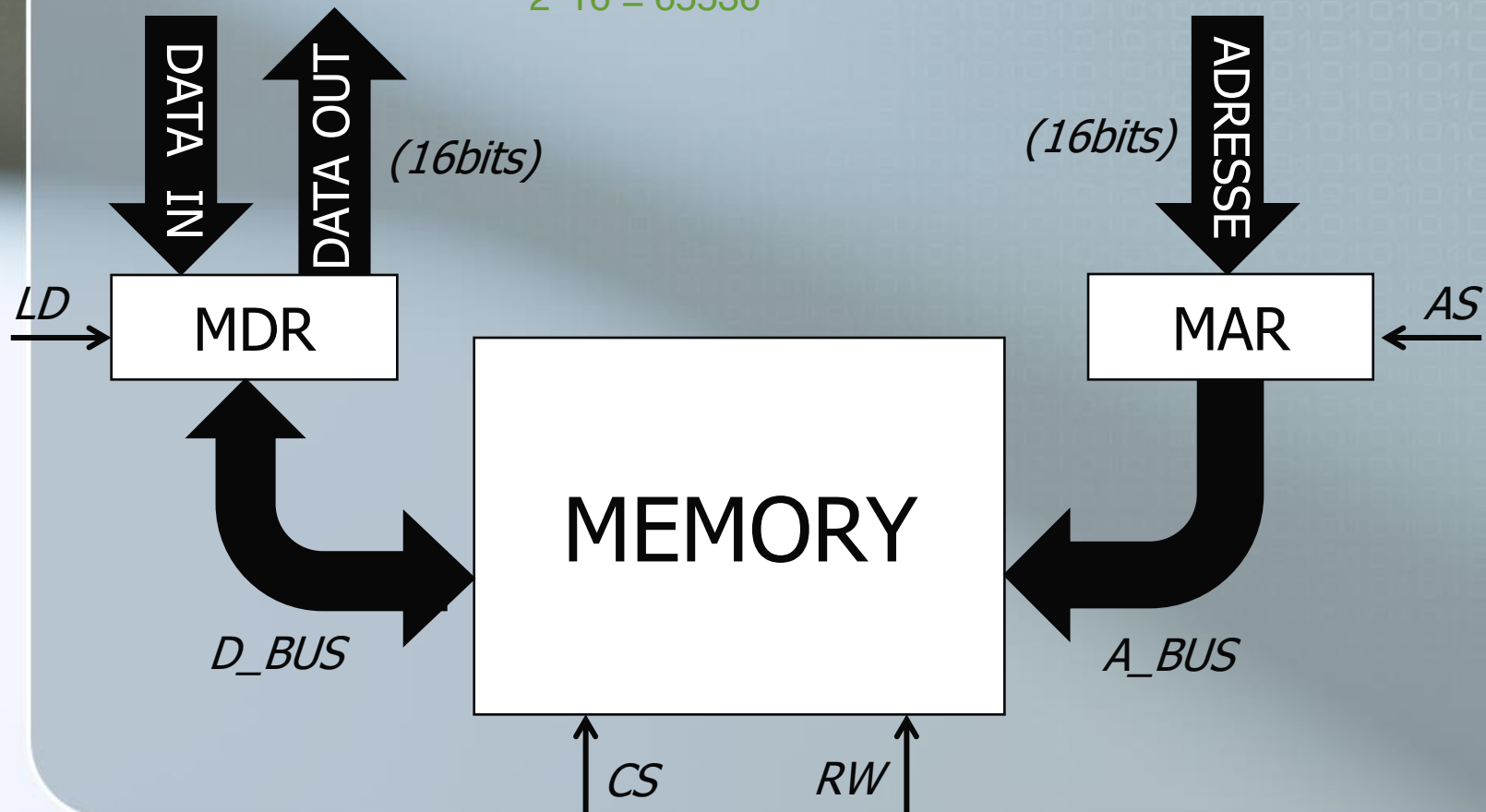
13

Exemple

■ Mémoire processeur LC-3 (*cf. prochain cours...*)

- Capacité: 64kmots de 16 bits → Adresses sur 16 bits

$$2^{16} = 65536$$



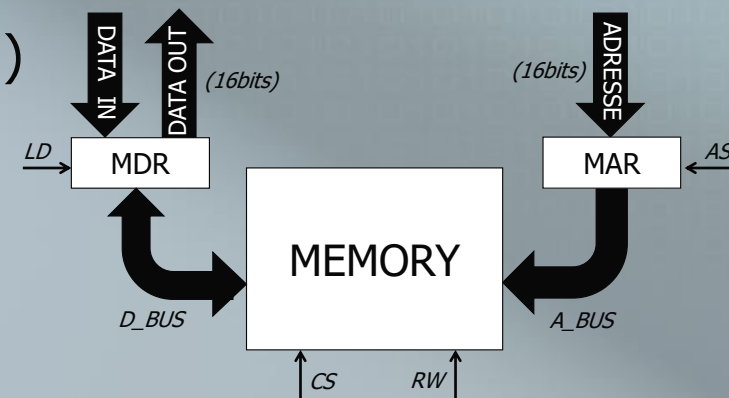
C5

14

Exemple

■ Mémoire processeur LC-3

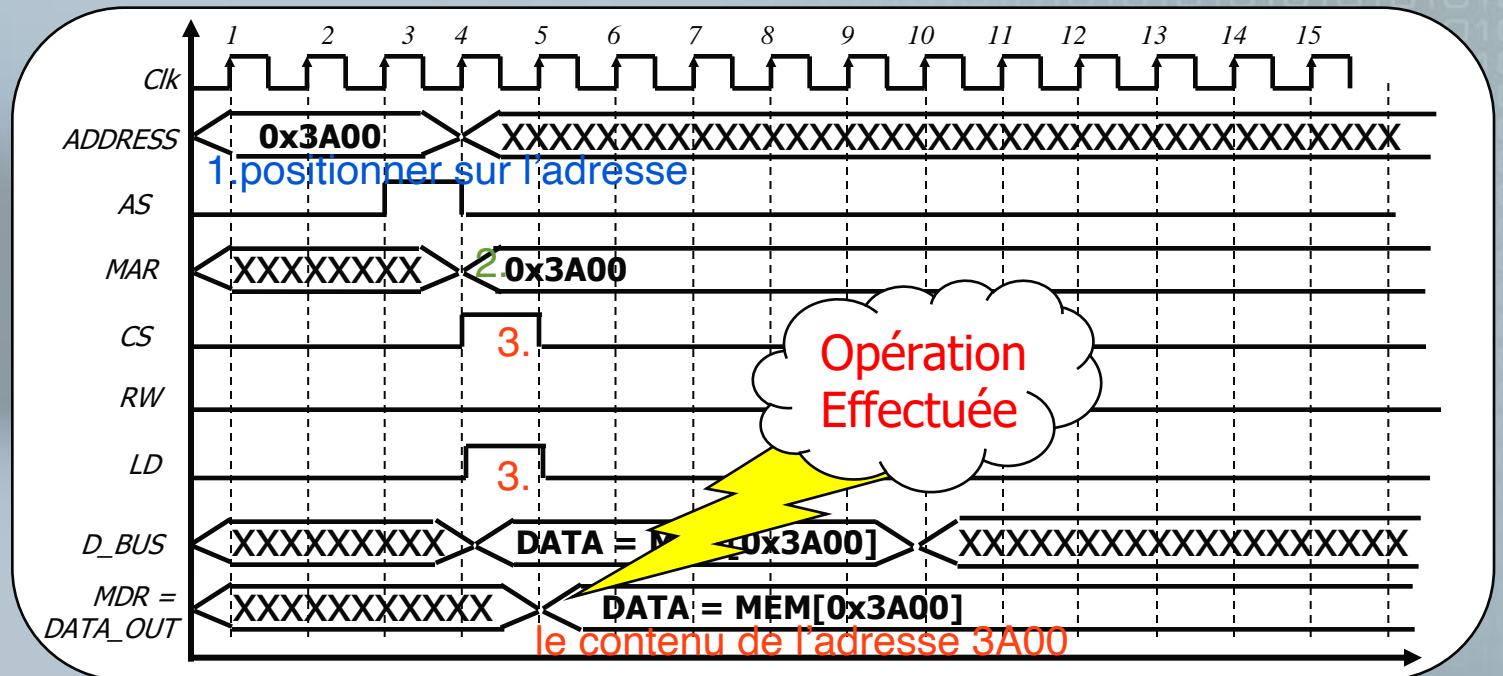
- MAR: Memory Address Register
- MDR: Memory Data Register
- AS: Address Strobe (*Commande MAR*)
- LD: Load (*Commande MDR*)
- A_BUS: Bus d'Adresses de la Mémoire
- D_BUS: Bus de Données de la Mémoire
- CS: Chip Select
- RW: Read (0) / Write (1)



C5

15

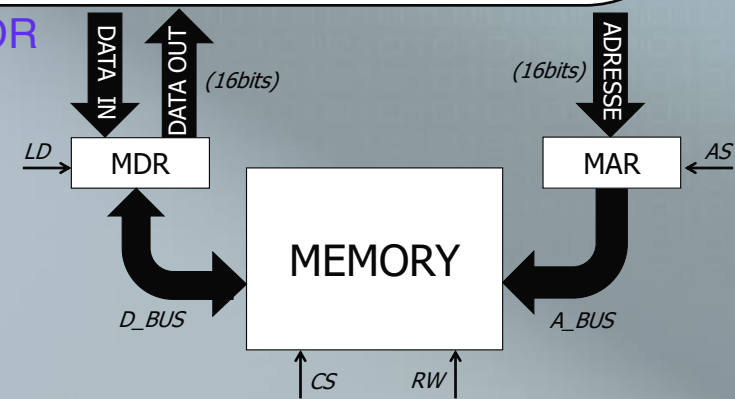
Exemple



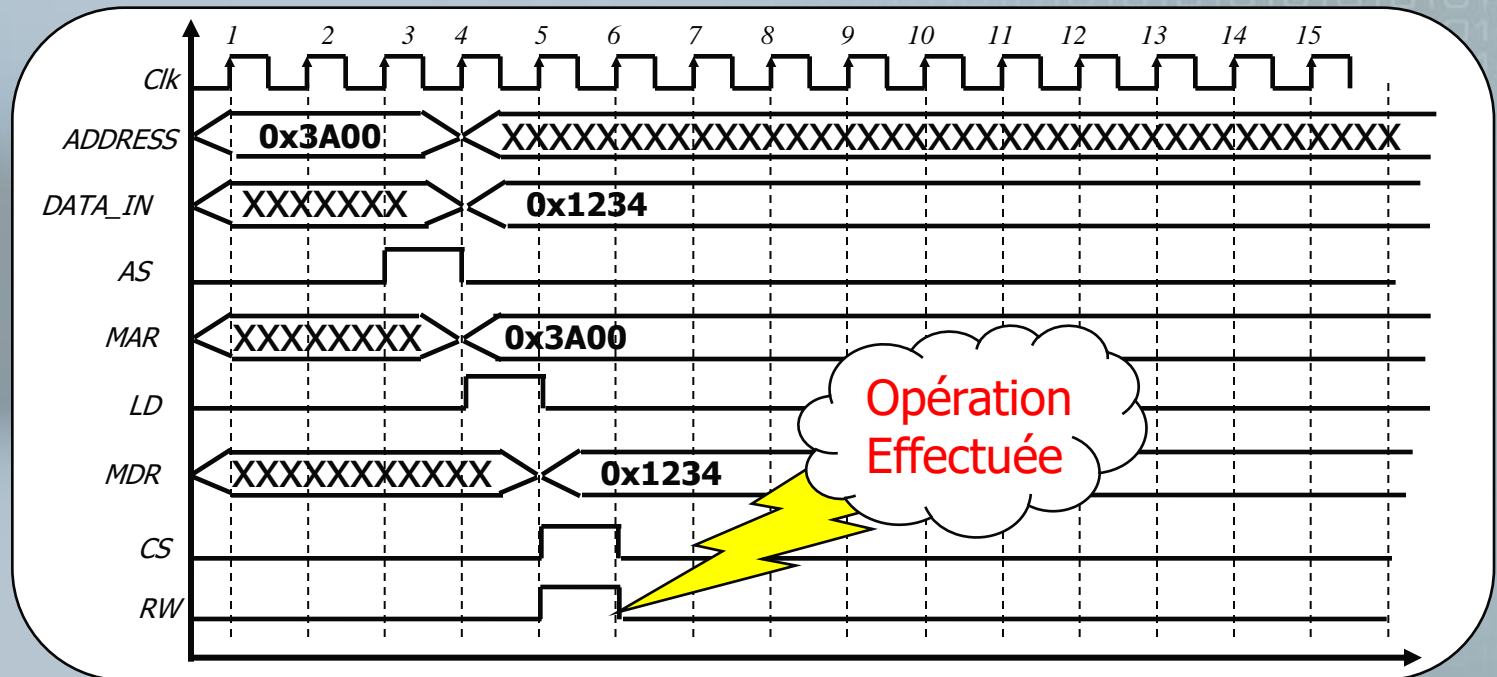
décalage 1 cycle: mémorisée dans MDR

■ Lecture mémoire LC-3

➤ Latence d'un cycle

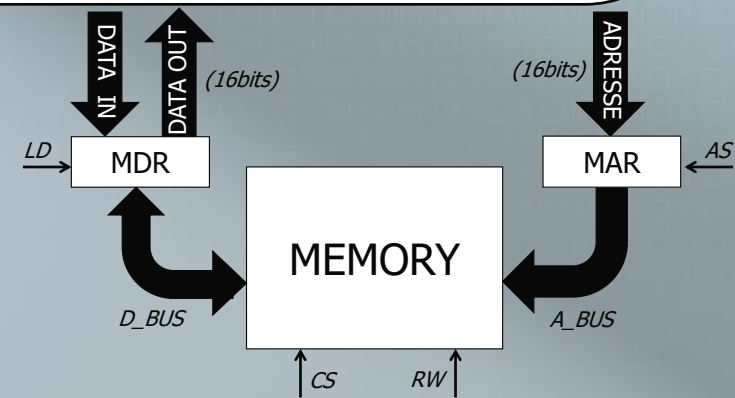


Exemple



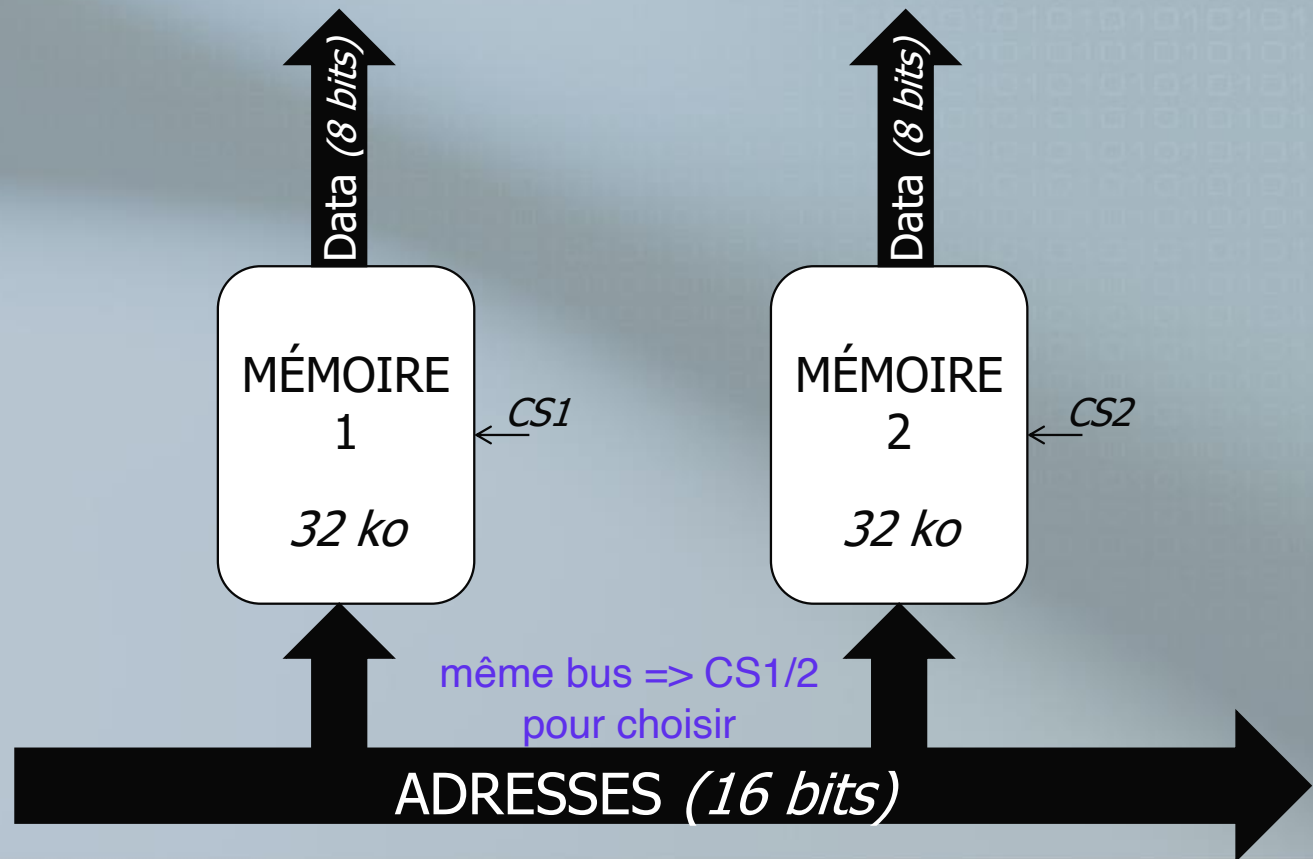
■ Ecriture mémoire LC-3

➤ Latence de deux cycles



Mémoire et Décodage Adresse

- 2 mémoires avec un bus d'adresses commun
 - Comment accéder à chaque mémoire individuellement?



C5

18

Mémoire et Décodage Adresse

- Mémoire de 32 koctets
 - Organisée en 32kcases de 8 bits
 - 32kcases = 2^{15} cases → 15 bits d'adresses
 - Adresses: de 0x0000 à 0x7FFF

0 -> 000 0000 0000 0000 0x0000
Max -> 111 1111 1111 1111 0x7FFF

0x0000

0x7FFF

MÉMOIRE

C5

19

Mémoire et Décodage Adresse

- 2 mémoires 32 koctets (32 kcases de 8 bits)

- 64 kcases au total (2^{16} cases)

- Bus d'adresses partagé

- 2^{16} cases \rightarrow 16 bits d'adresse

- Adresses: 0x0000 à 0xFFFF

Bloc 1

Bloc 2

```
0 000 0000 0000 0000
0 111 1111 1111 1111
1 000 0000 0000 0000
1 111 1111 1111 1111
```

0x0000

MÉMOIRE
1

0x7FFF

0x8000

MÉMOIRE
2

0xFFFF

C5

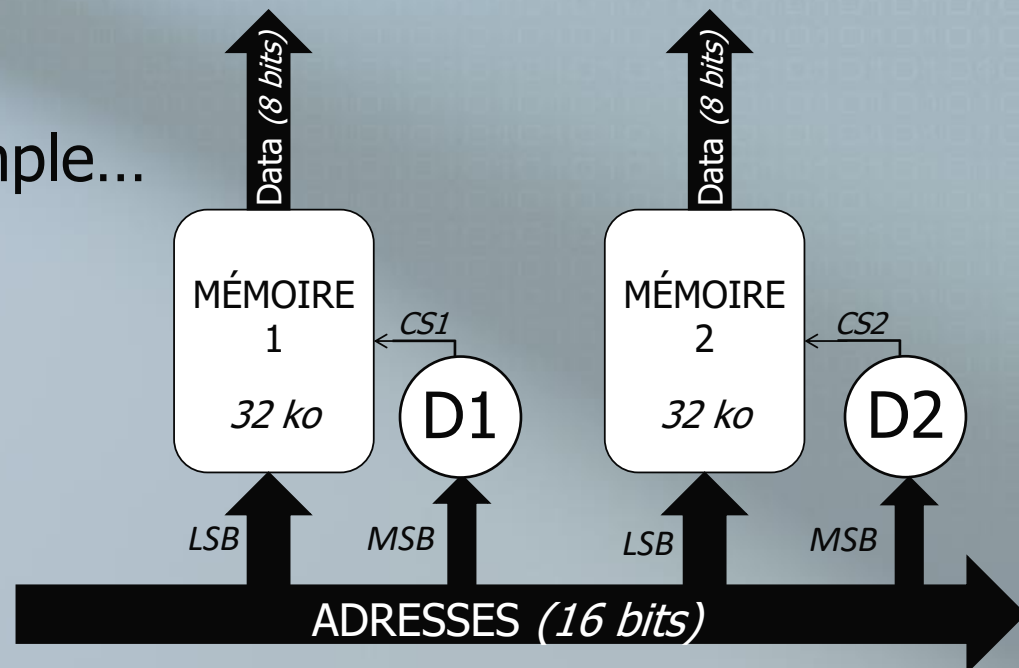
20

Mémoire et Décodage Adresse

- Activation du Chip Select d'un bloc mémoire
- Décodage de l'adresse ($A_{15}...A_0$) désirée
 - Adresse $[0x0000;0x7FFF] \rightarrow A_{15}=0 \rightarrow$ Validation CS1
 - Adresse $[0x8000;0xFFFF] \rightarrow A_{15}=1 \rightarrow$ Validation CS2

- Pour notre exemple...

- $D1 = \overline{A_{15}}$
- $D2 = A_{15}$

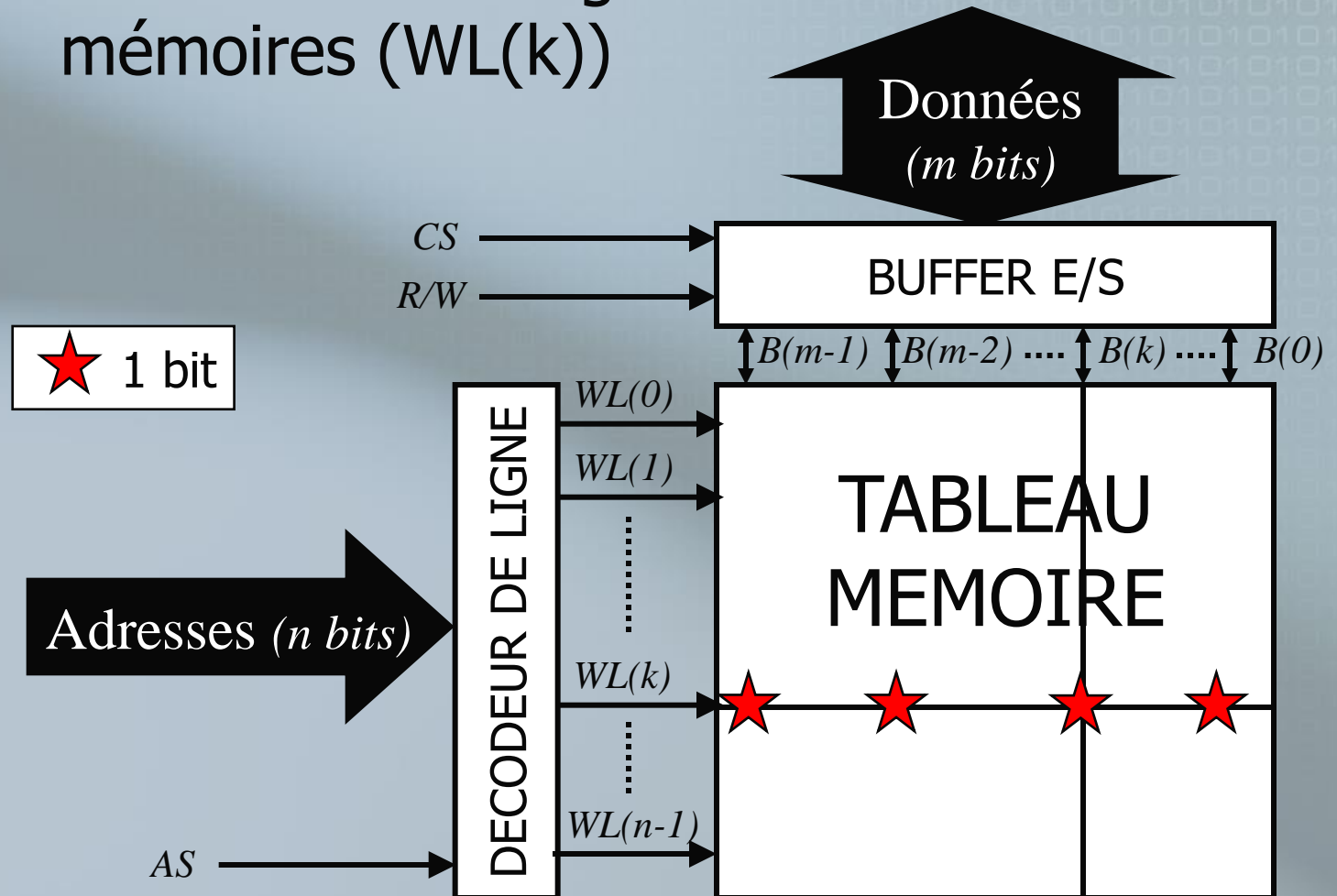


C5

21

Organisation Interne Mémoire

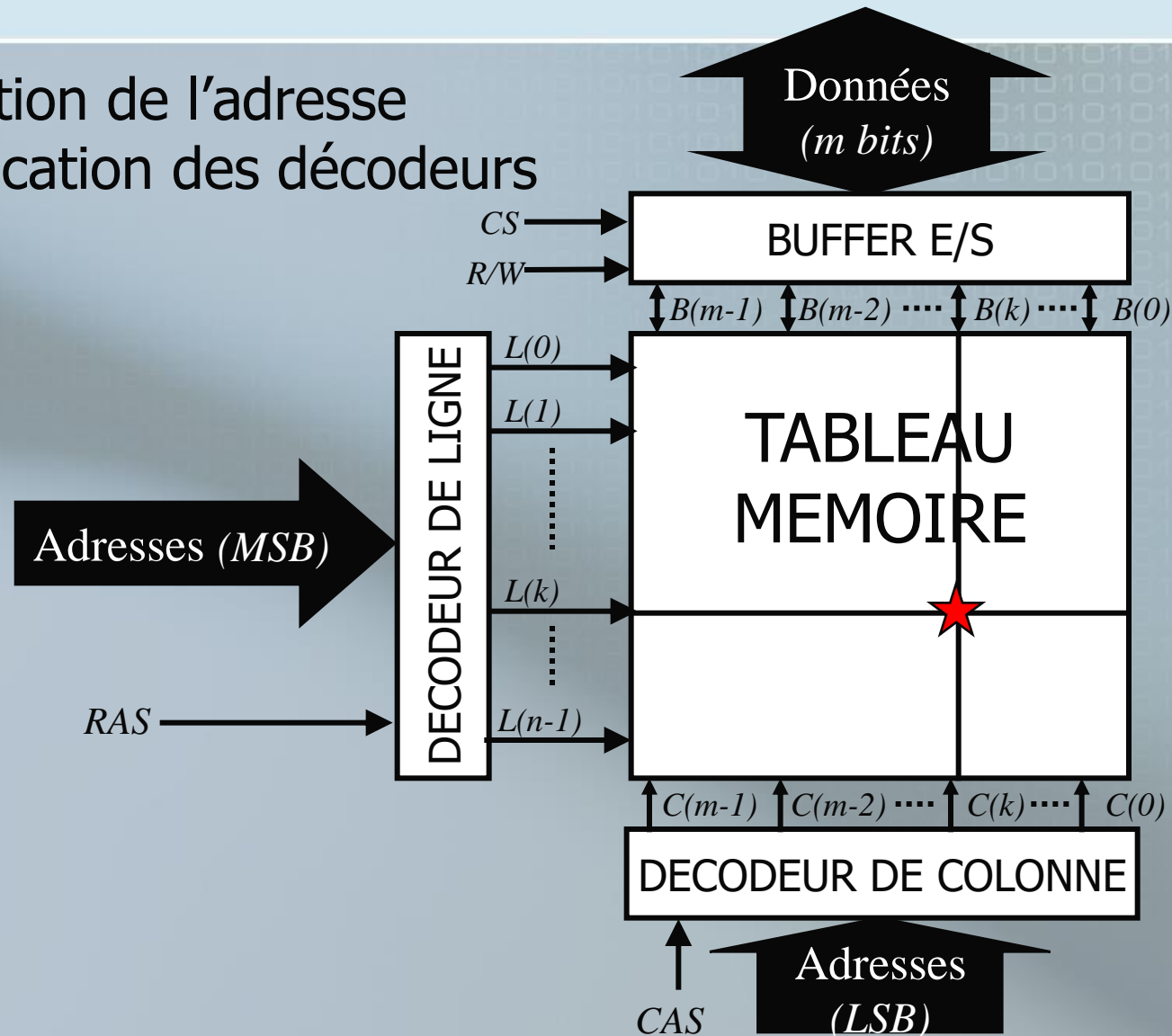
- Le décodeur de ligne valide un des mots mémoires ($WL(k)$)



Organisation Interne Mémoire

- Séparation de l'adresse
- Simplification des décodeurs

★ 1 bit



C5

23

Organisation Interne Mémoire

- Si plusieurs mots par lignes
- 1 colonne = 1 mot

★ 1 mot de N bits

Adresses (MSB)

RAS

DECODEUR DE LIGNE

CS
R/W

Données
(m bits)

BUFFER E/S

$B(m-1)$ $B(m-2)$... $B(k)$... $B(0)$

TABLEAU
MEMOIRE

$L(0)$
 $L(1)$
...
 $L(k)$
...
 $L(n-1)$

$C(m-1)$ $C(m-2)$... $C(k)$... $C(0)$

DECODEUR DE COLONNE

CAS

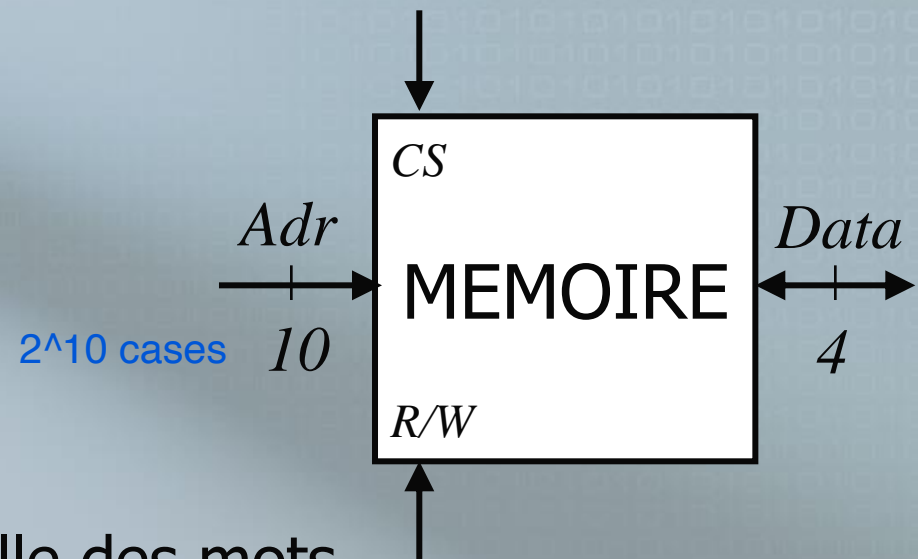
Adresses
(LSB)

C5

24

Organisation Externe Mémoire

- Association de plusieurs blocs



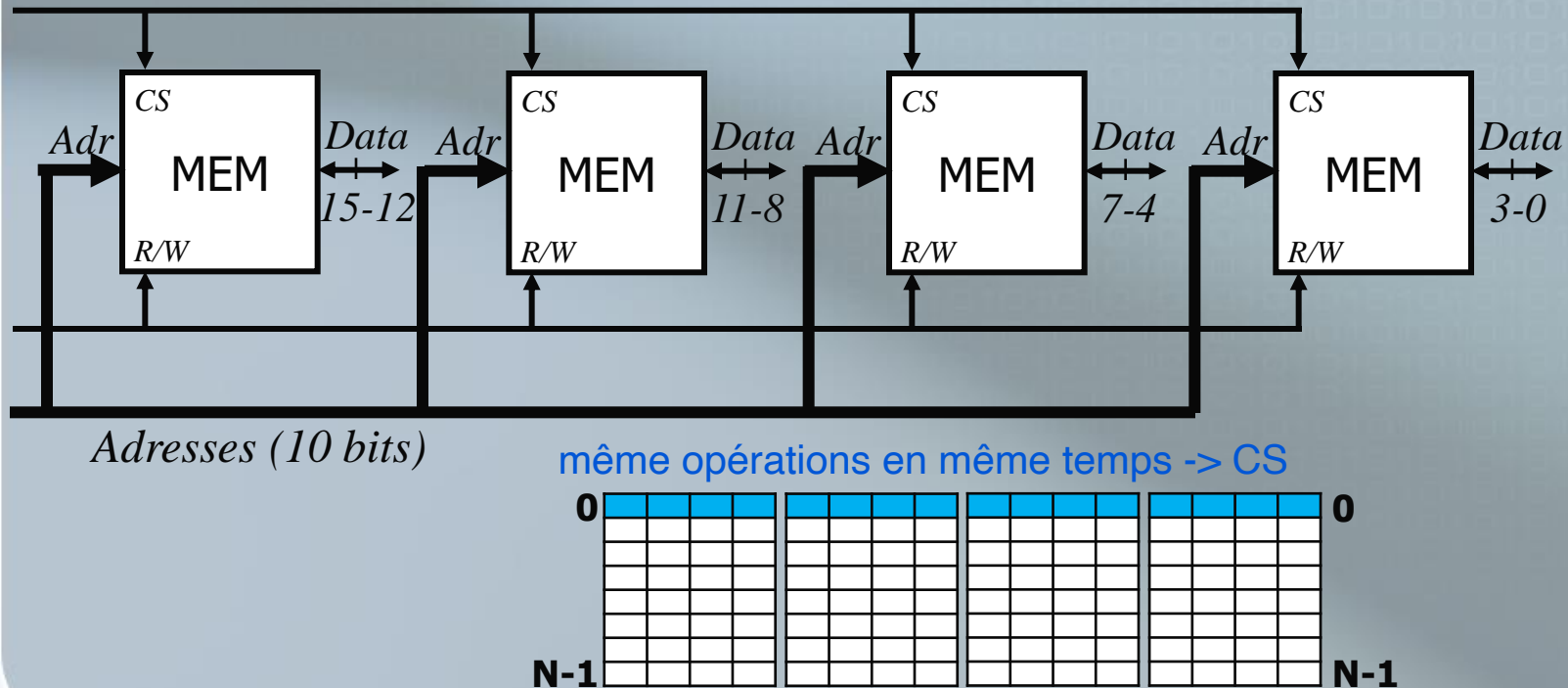
- Permet de:
 - Augmenter la taille des mots
 - Augmenter l'espace de stockage (le nombre de mots)

C5

25

Organisation Externe Mémoire

- Augmentation de la taille des mots
 - Exemple: passage de mots de 4 à 16 bits

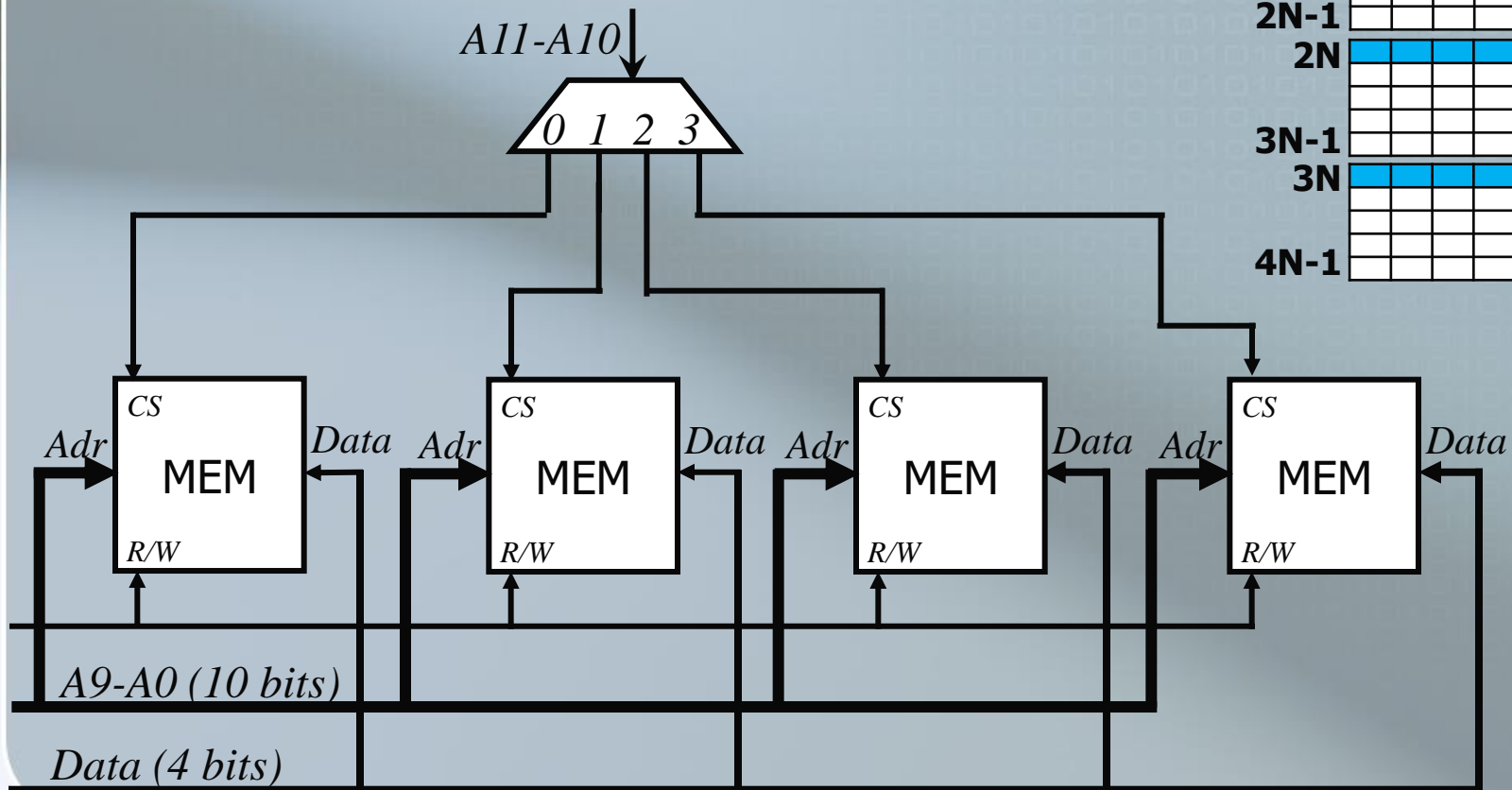


C5

26

Organisation Externe RAM

- Augmentation espace de stockage
 - Exemple: Passage d'une adresse de 10 à 12 bits

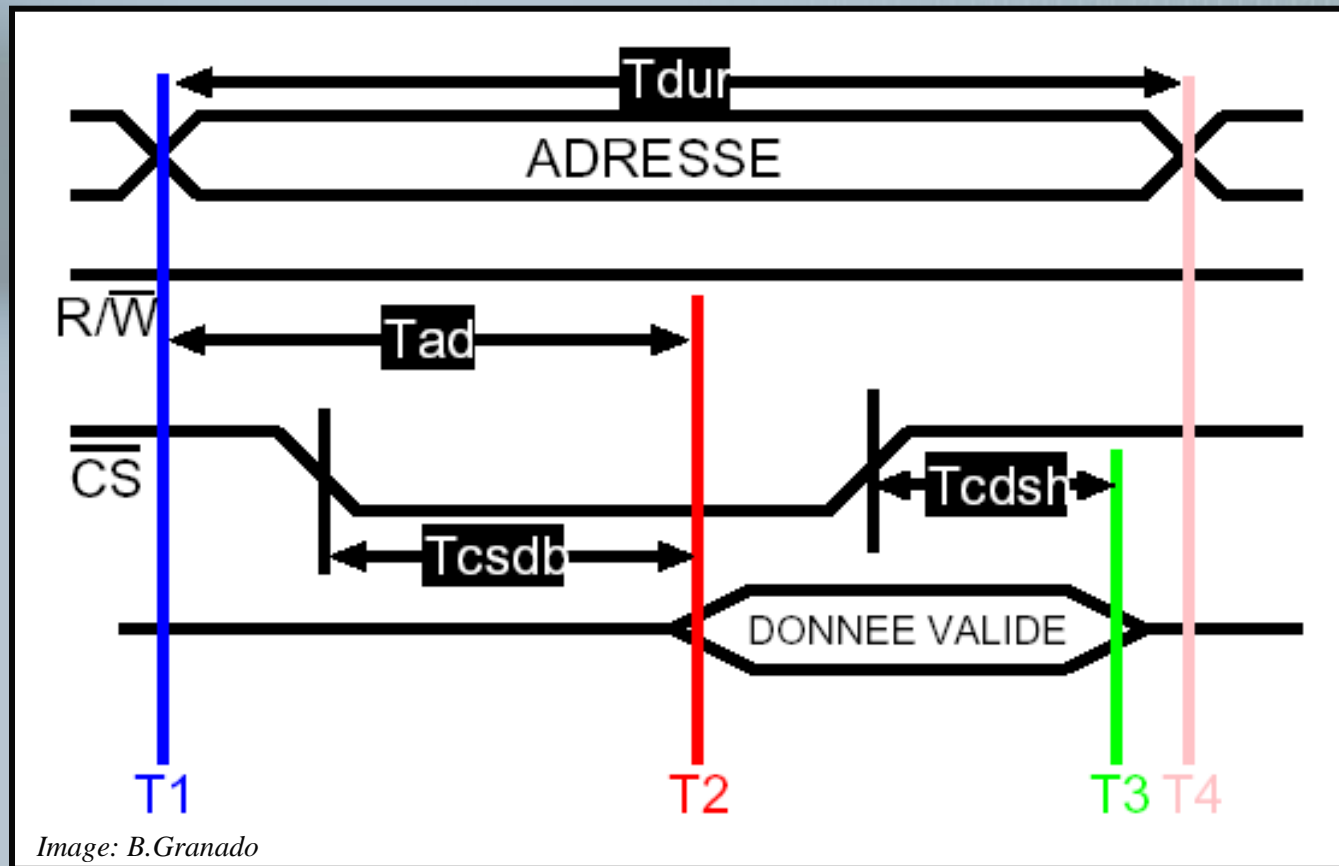


C5

27

Cycles de Fonctionnement

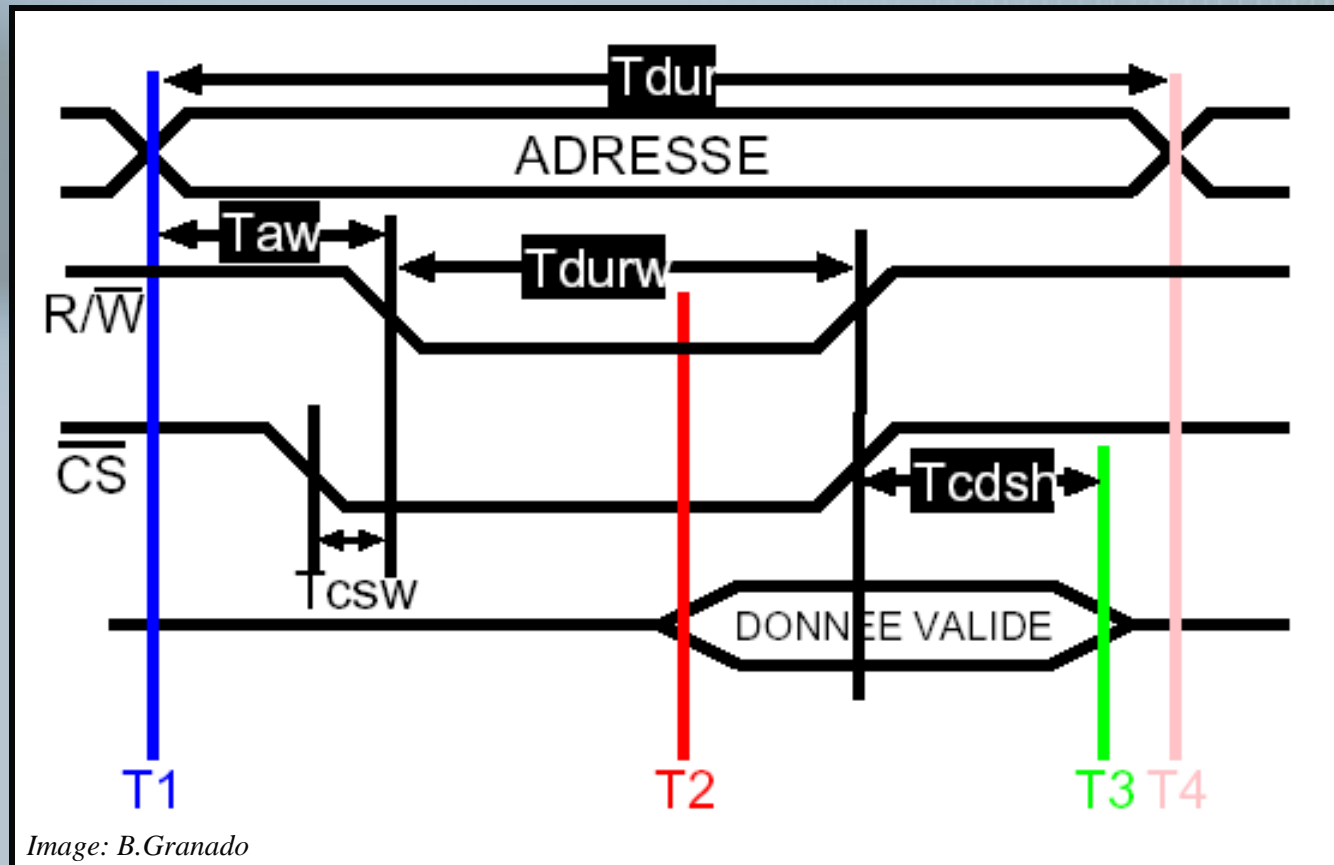
■ Cycle de lecture (exemple)



- Les timings à respecter sont fonction du composant

Cycles de Fonctionnement

■ Cycle d'écriture (exemple)



- Les timings à respecter sont fonction du composant

Description VHDL

- Description d'un composant de 128 mots de 8 bits

```
entity memoire is
port ( donnee : in std_logic_vector (7 downto 0);
      adresse: in std_logic_vector (6 downto 0);
      rw : in std_logic;                7 => 2^7 cases = 128 cases
      cs : in std_logic;
      sortie: out std_logic_vector (7 downto 0));
end memoire;
```

C5

30

Description VHDL

```
architecture comport of memoire is
    type memory is array (0 to 127) of std_logic_vector(7 downto 0);
    signal mem : memory; // signal de type memory
begin

    process(cs, adresse, donnee, rw) is
    begin
        if cs='0' then
            if rw='1' then read
                sortie <= mem(conv_integer(adresse));
            elsif rw='0' then write
                mem(conv_integer(adresse)) <= donnee;
            end if;
        else
            sortie <= "ZZZZZZZZ"; // état haute impédance
        end if;
    end process;
end comport;
```

*Déclaration d'un type
tableau de std_logic_vector*

Description VHDL

```
architecture comport of memoire is
type memory is array (0 to 127) of std_logic_vector(7 downto 0);
signal mem : memory;
begin

    process(cs, adresse, donnee, rw) is
    begin
        if cs='0' then
            if rw='1' then
                sortie <= mem(conv_integer(adresse));
            elsif rw='0' then
                mem(conv_integer(adresse)) <= donnee;
            end if;
        else
            sortie <= "ZZZZZZZZ";
        end if;
    end process;
end comport;
```

*Fonction de conversion
std_logic → integer*

C5

32

Description VHDL

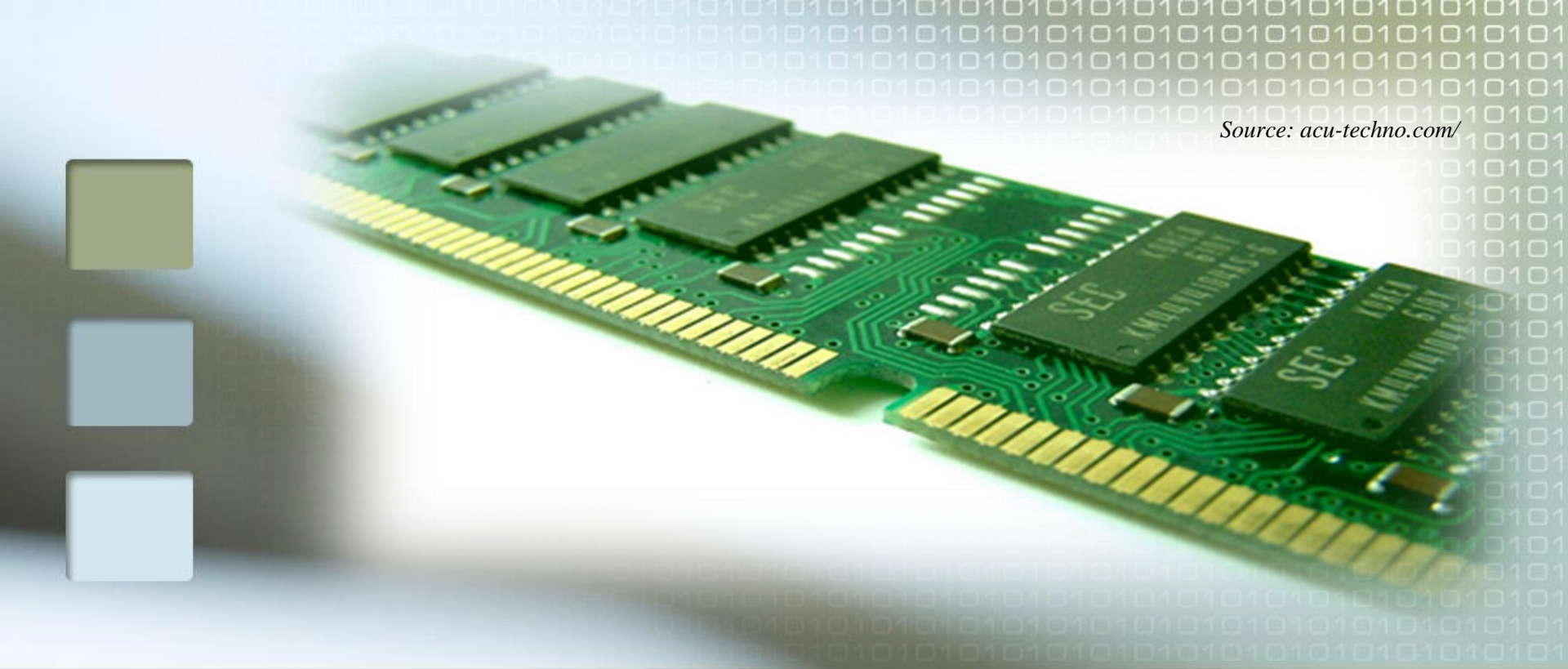
```
architecture comport of memoire is
type memory is array (0 to 127) of std_logic_vector(7 downto 0);
signal mem : memory;
begin

    process(cs, adresse, donnee, rw) is
    begin
        if cs='0' then
            if rw='1' then
                sortie <= mem(conv_integer(adresse));
            elsif rw='0' then
                mem(conv_integer(adresse)) <= donnee;
            end if;
        else
            sortie <= "ZZZZZZZZ";
        end if;
    end process;
end comport;
```

*Sortie en haute impédance
si boîtier désélectionné*

C5

33



Mémoires RAM

SRAM, DRAM...

Mémoires RAM

- Mémoire volatile
- Il existe deux grandes familles de RAM
 - Mémoires RAM statiques (SRAM)
 - Technologie basée sur l'utilisation de portes logiques
 - Très rapides
 - Plus volumineuses et consommatrices d'énergie
 - Mémoires RAM dynamiques (DRAM)
 - Technologie basée sur l'utilisation de condensateurs
 - Plus compactes
 - Moins rapides et nécessitent un rafraîchissement de l'information

C5

35

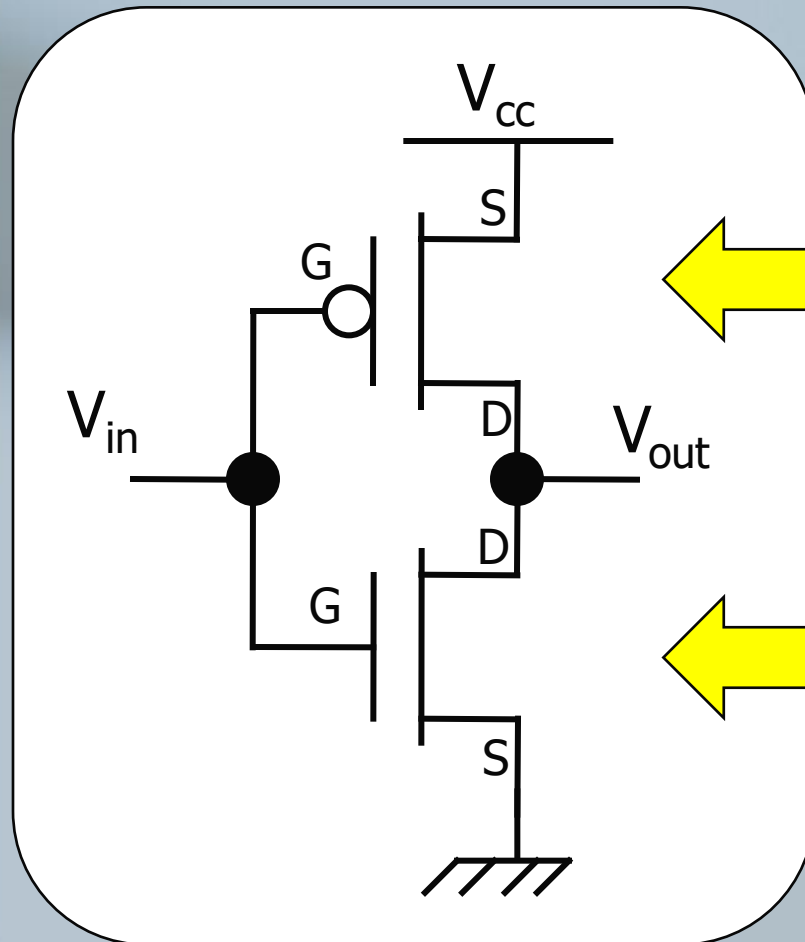
Static RAM (SRAM)

- Mémoire volatile
- Cellule rapide
- Cellule volumineuse: *(Nécessite 2 inverseurs)*
- Information stockée sans besoin de rafraîchissement
 - Tant que le circuit est alimenté
- Lecture non destructrice
- Coût important

C5

36

Inverseur CMOS



- 2 transistors

- PMOS

- Passant si $V_G = 0V$

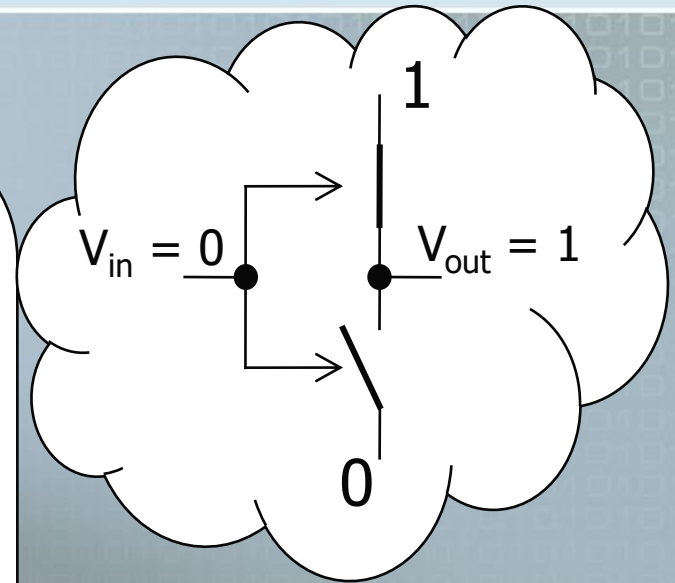
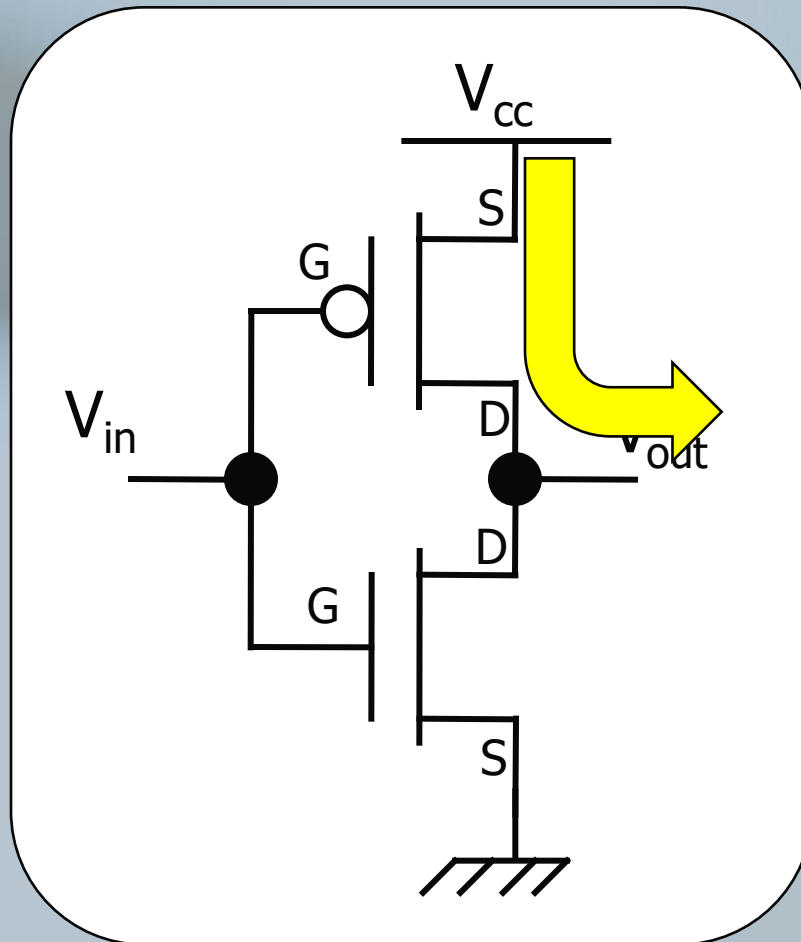
- Bloqué si $V_G = V_{CC}$

- NMOS

- Bloqué si $V_G = 0V$

- Passant si $V_G = V_{CC}$

Inverseur CMOS

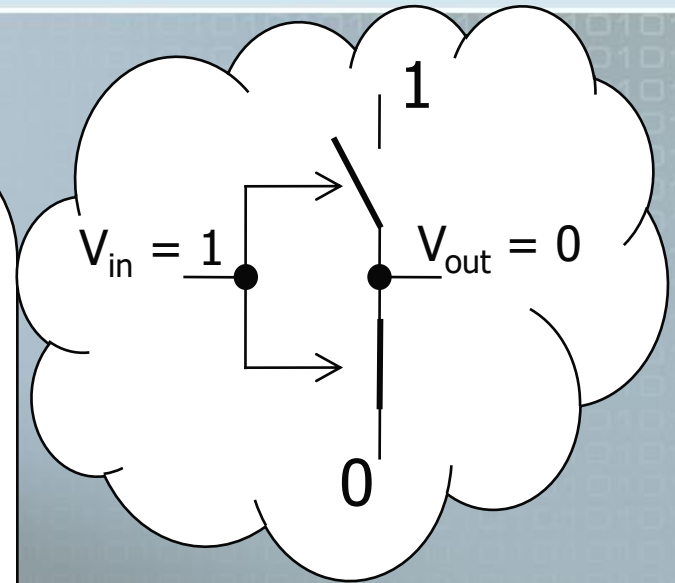
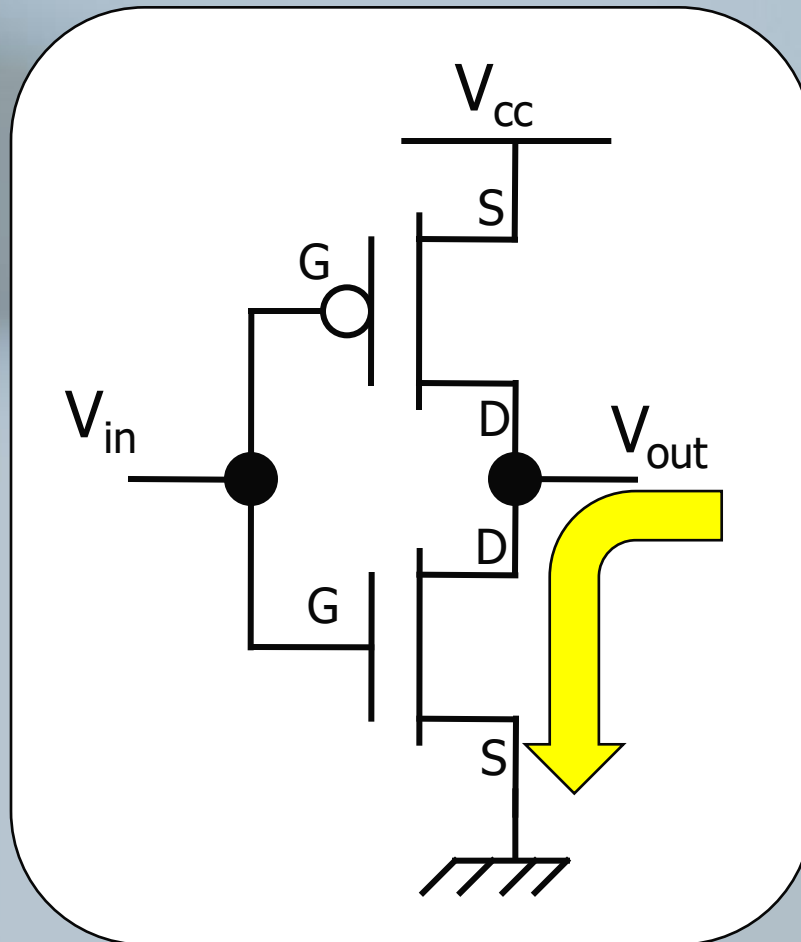


- $V_{in} = 0$
 - PMOS passant
 - NMOS bloqué
- $V_{out} = 1$

C5

38

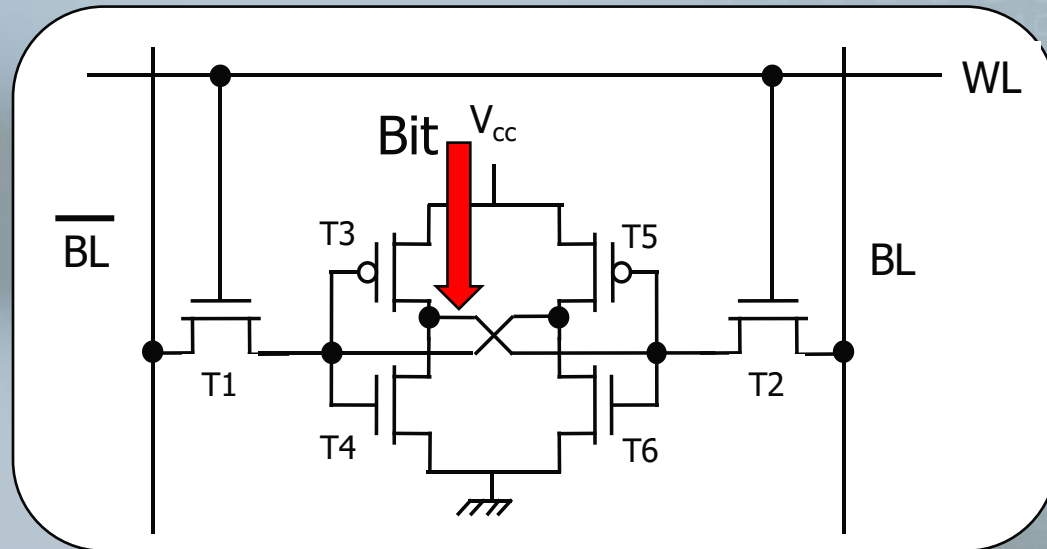
Inverseur CMOS



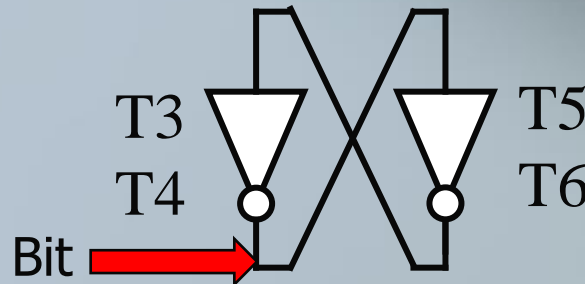
- $V_{in} = 1$
 - PMOS bloqué
 - NMOS passant
- $V_{out} = 0$

Static RAM (SRAM)

■ Point mémoire



*WL: Word Line
BL: Bit Line*



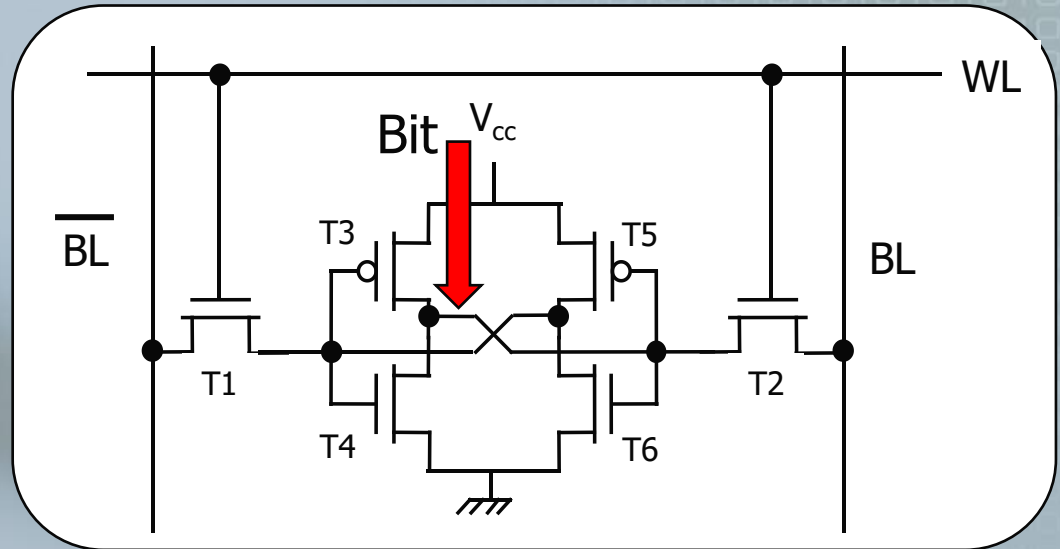
- T1 et T2 contrôlent l'accès au point mémoire

C5

40

Static RAM (SRAM)

■ Lecture



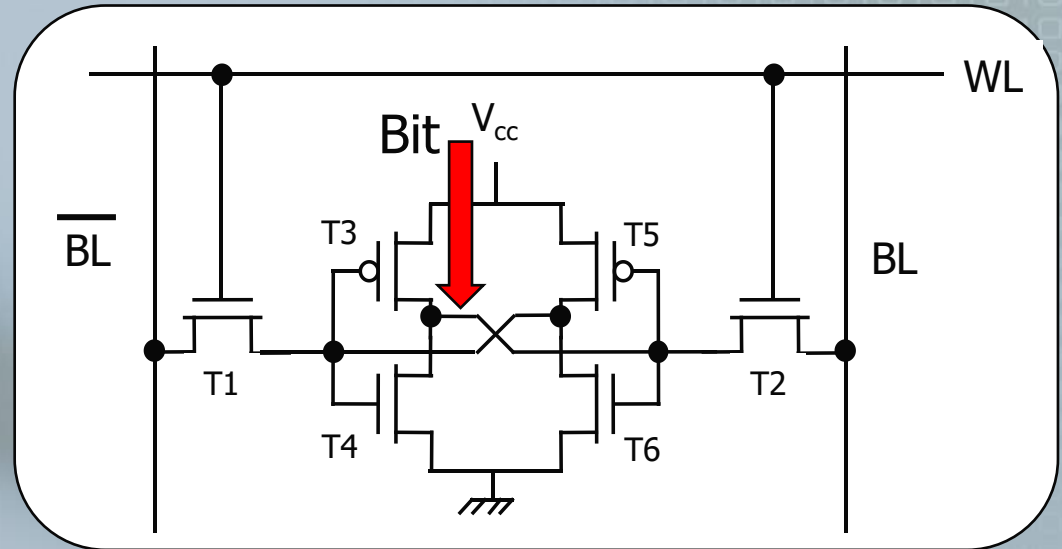
- Précharge de BL et /BL à une tension entre 0 et V_{CC}
- Mise à 1 de WL \rightarrow T1 et T2 passants
 - Si Bit = 0 $\rightarrow V_{BL} \searrow$ et $V_{/BL} \nearrow$
 - Si Bit = 1 $\rightarrow V_{BL} \nearrow$ et $V_{/BL} \searrow$
- Donnée en sortie est fixée selon la différence entre V_{BL} et $V_{/BL}$

C5

41

Static RAM (SRAM)

■ Ecriture



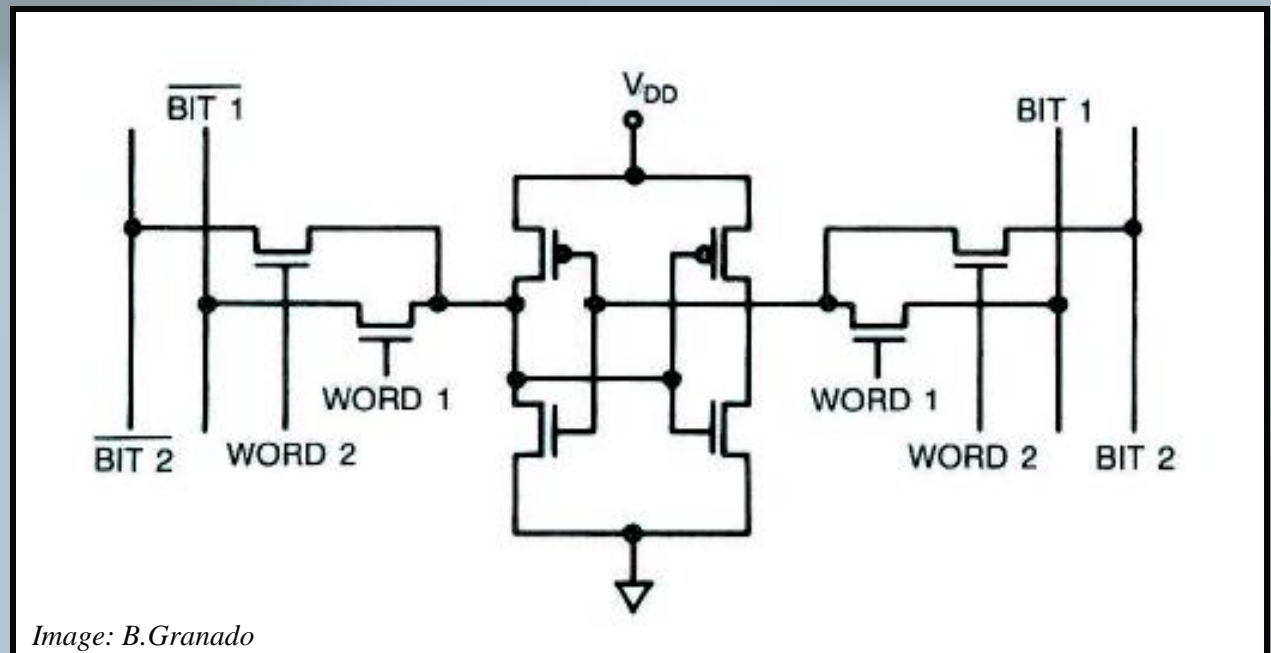
- La donnée à écrire est positionnée sur BL (son inverse sur /BL)
- Mise à 1 de WL → Enregistrement de la donnée
 - Possible car $T1, T2 \gg T3, T4, T5, T6$

Static RAM (SRAM)

- Autres types de cellule

- Cellule Biport

- Permet d'accéder deux fois simultanément à un point mémoire (en lecture ou en écriture)



Dynamic RAM (DRAM)

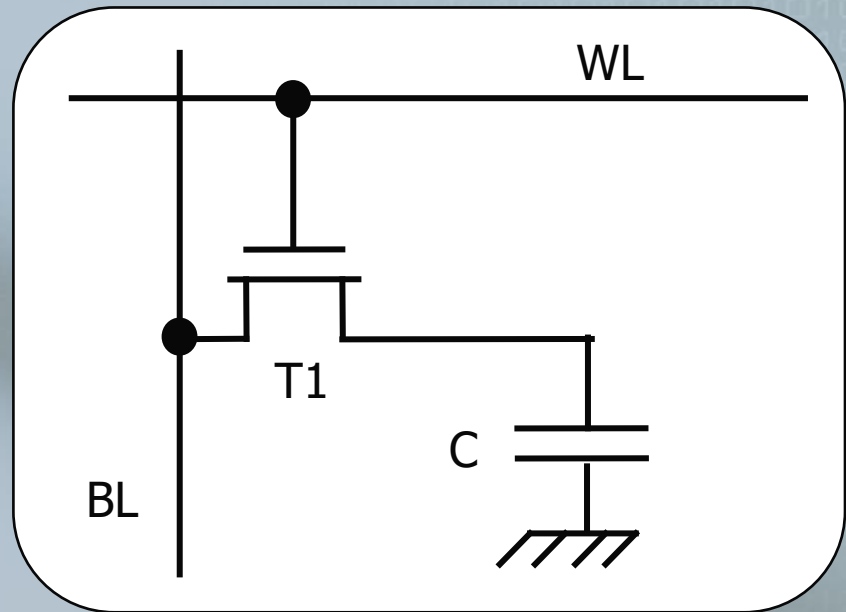
- Mémoire volatile
- Cellule moins rapide
- Cellule compacte (*1 capacité*)
- Nécessité de rafraîchir les informations stockées (décharge du condensateur)
- Lecture destructive
- Bon marché

C5

44

Dynamic RAM (DRAM)

■ Point Mémoire

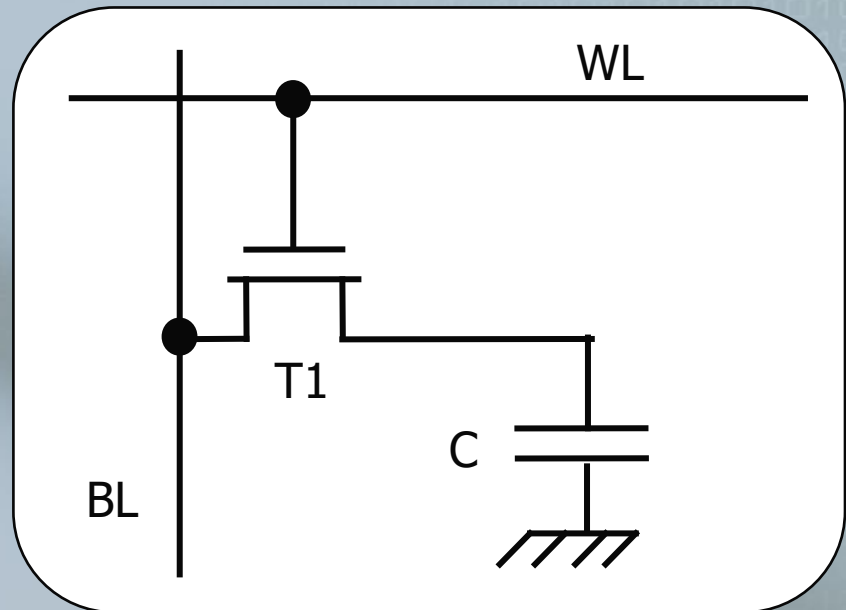


C5

45

Dynamic RAM (DRAM)

■ Lecture



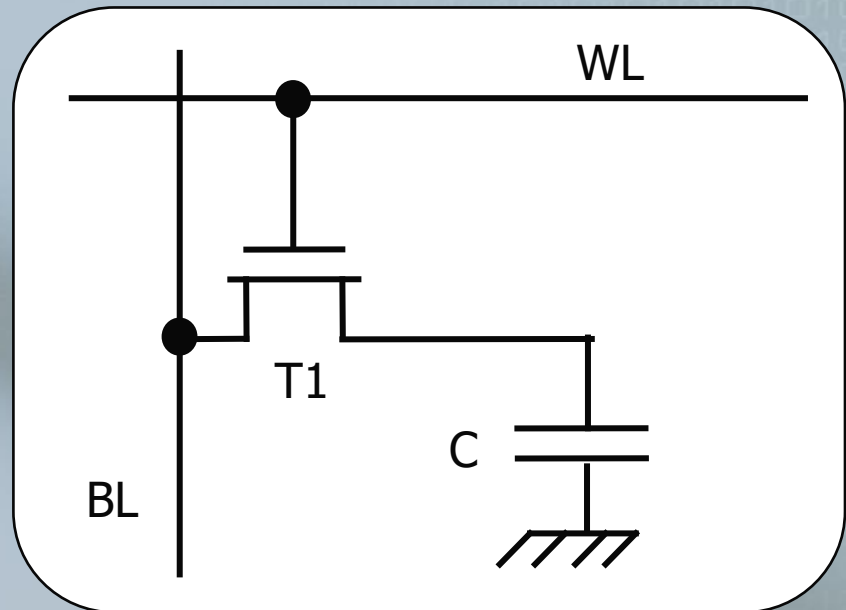
- Mise à 1 de WL → T1 passant
- Niveau logique contenu dans C transféré dans BL

C5

46

Dynamic RAM (DRAM)

■ Ecriture



- Positionnement de la valeur à écrire sur BL
- Mise à 1 de WL → T1 passant
 - Enregistrement de la donnée dans C

Technologie SDRAM

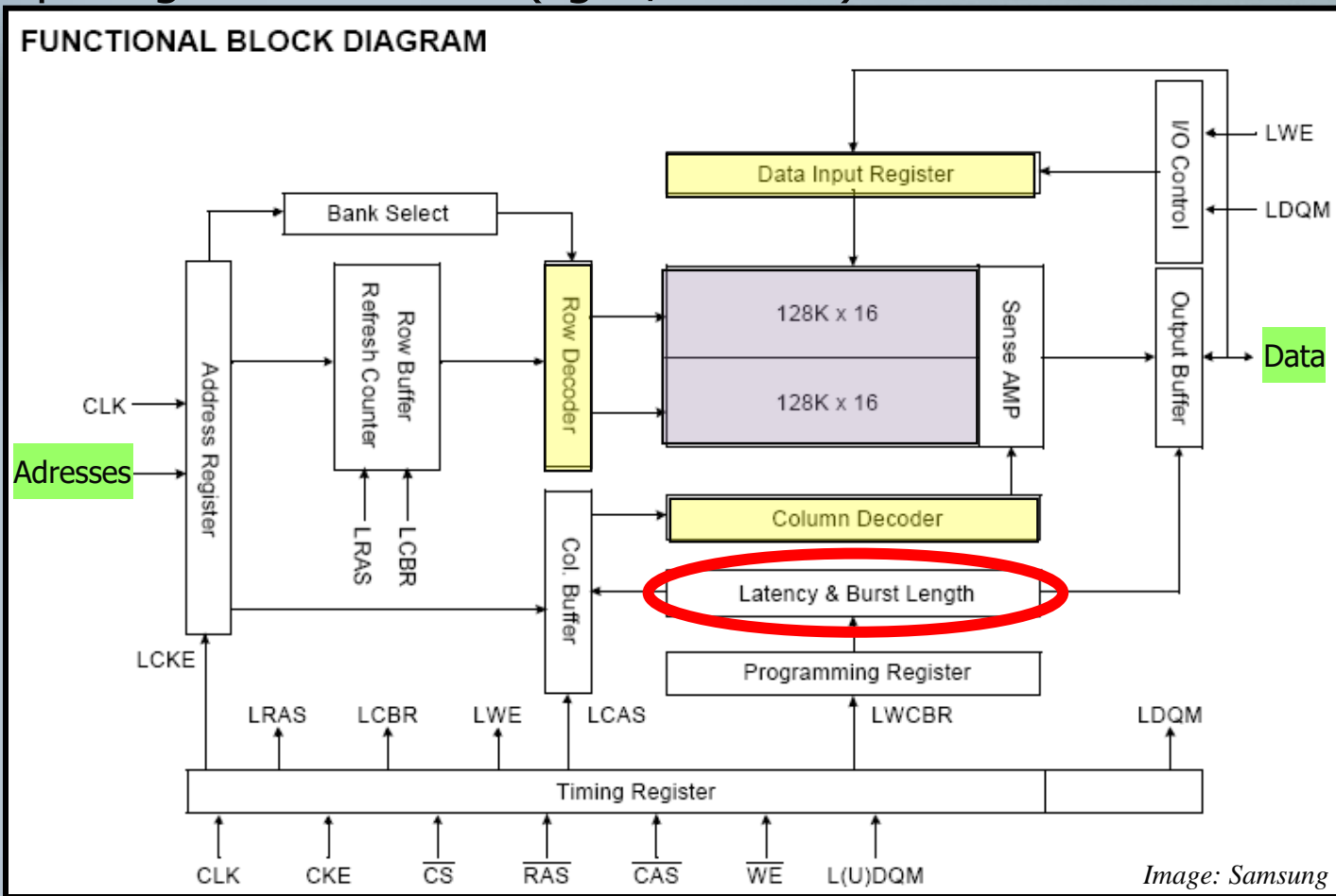
- Jusqu'au milieu des années 90: RAM asynchrone
- Synchronisation difficile avec des cartes mère d'ordinateurs
- Technologie Synchronous DRAM (SDRAM)
 - Réduction des cycles de latence et d'accès mémoire
 - Pipelinage d'instructions
 - Introduction du mode rafale (Burst)

C5

48

Exemple: SDRAM Samsung

- 4 Mb – 2 bancs mémoire
- Multiplexage des adresses (ligne/colonne)



C5

49

Exemple: SDRAM Samsung

■ Signaux de contrôle

PIN FUNCTION DESCRIPTION

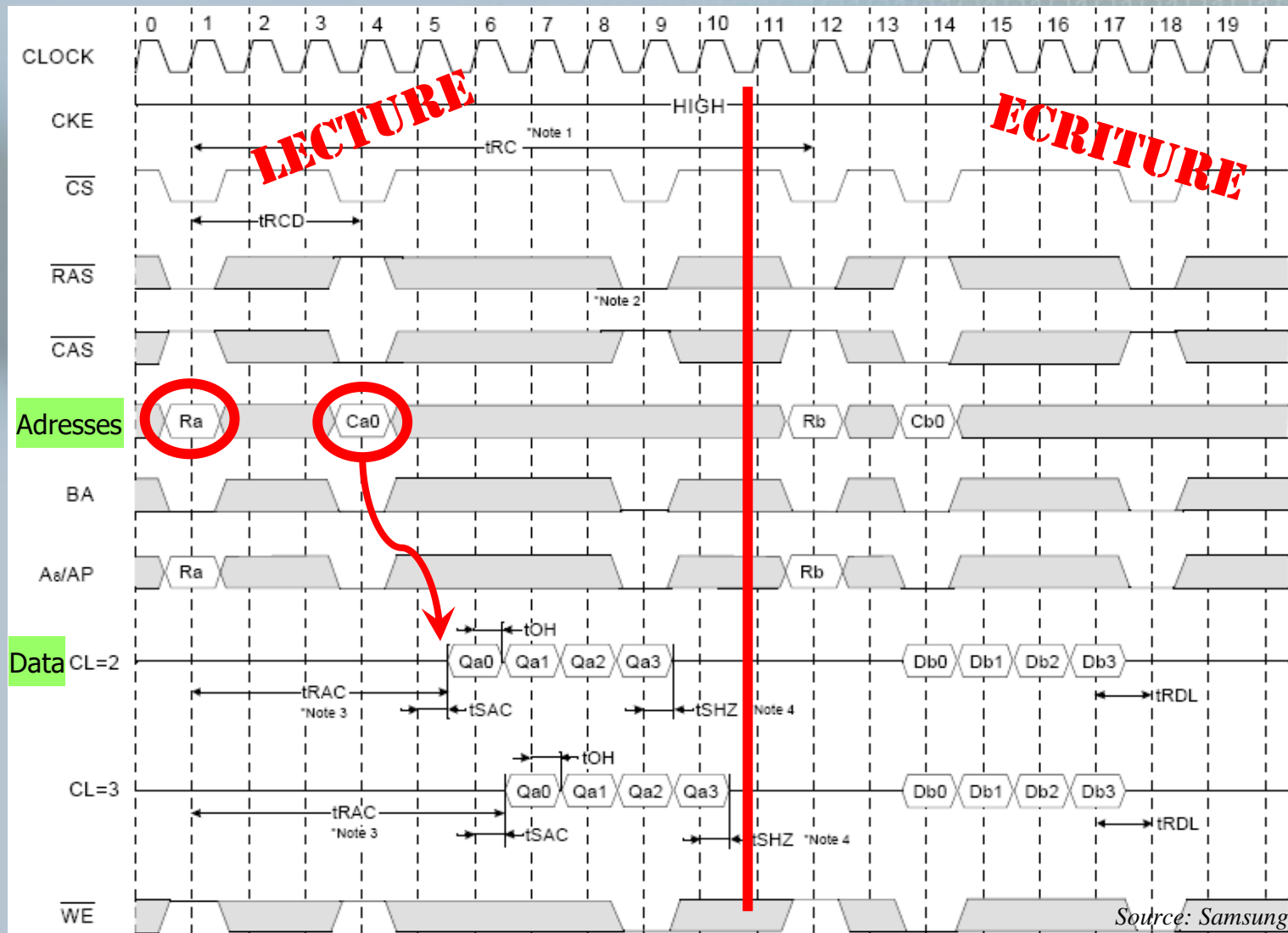
Source: Samsung

Pin	Name	Input Function
CLK	<i>System Clock</i>	Active on the positive going edge to sample all inputs.
$\overline{\text{CS}}$	<i>Chip Select</i>	Disables or enables device operation by masking or enabling all inputs except CLK, CKE and L(U)DQM
CKE	<i>Clock Enable</i>	Masks system clock to freeze operation from the next clock cycle. CKE should be enabled at least one cycle prior to new command. Disable input buffers for power down in standby.
Adresses	<i>Address</i>	Row / column addresses are multiplexed on the same pins. Row address : RA0 ~ RA8, column address : CA0 ~ CA7
BA	<i>Bank Select Address</i>	Selects bank to be activated during row address latch time. Selects bank for read/write during column address latch time.
$\overline{\text{RAS}}$	<i>Row Address Strobe</i>	Latches row addresses on the positive going edge of the CLK with $\overline{\text{RAS}}$ low. Enables row access & precharge.
$\overline{\text{CAS}}$	<i>Column Address Strobe</i>	Latches column addresses on the positive going edge of the CLK with $\overline{\text{CAS}}$ low. Enables column access.
$\overline{\text{WE}}$	<i>Write Enable</i>	Enables write operation and row precharge. Latches data in starting from $\overline{\text{CAS}}$, WE active.
Data	<i>Data Input/Output Mask</i>	Makes data output Hi-Z, tshz after the clock and masks the output. Blocks data input when L(U)DQM active.

C5

50

Exemple: SDRAM Samsung



C5

51

Technologie DDR-SDRAM

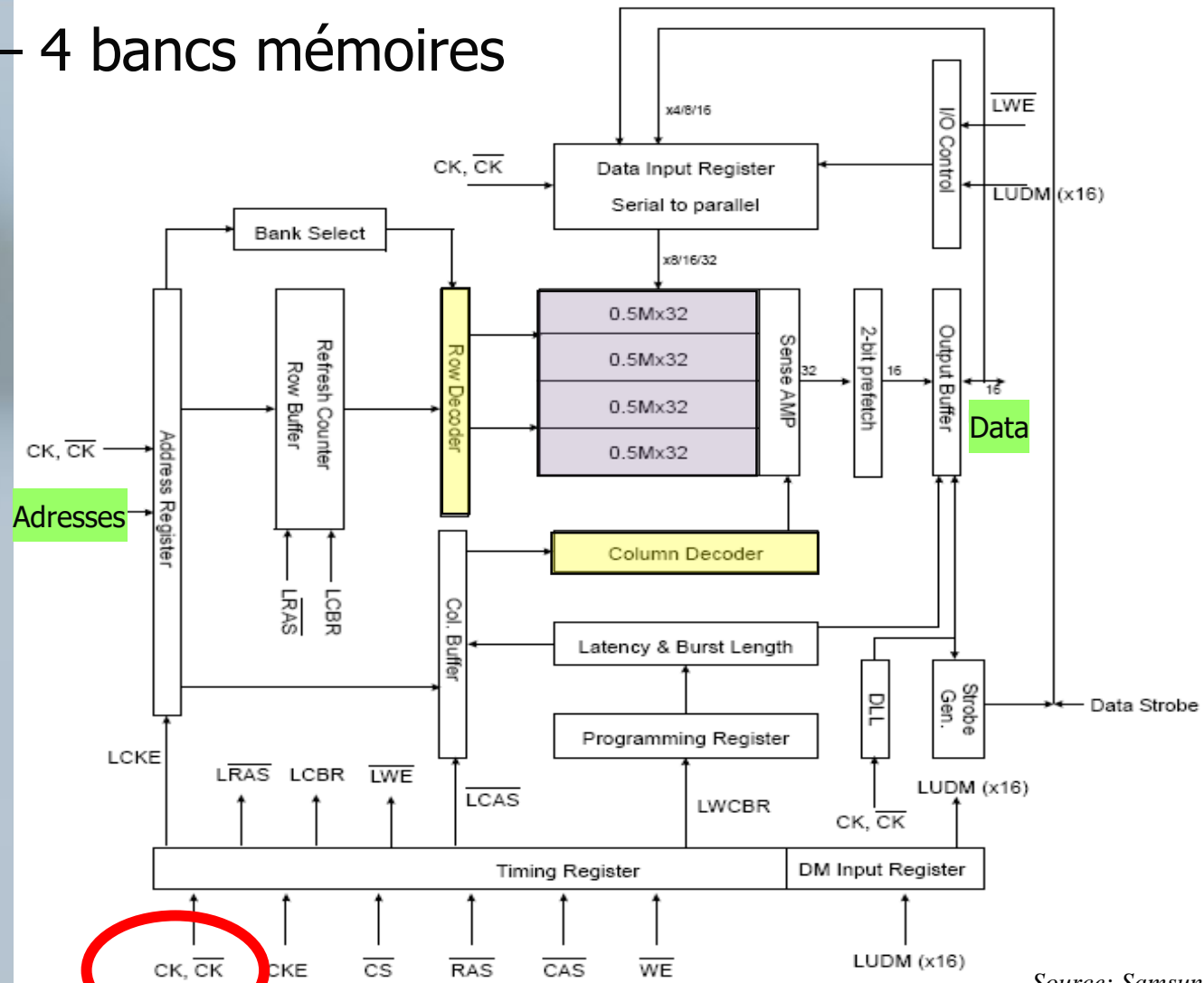
- Double Data Rate SDRAM
- Concept: Réduire l'écart entre les fréquences des processeurs (\sim GHz) et celles des bus mémoires (\sim 100 MHz)
- Solution: Lecture et écriture sur les deux fronts d'horloge

C5

52

Exemple: Samsung

- 4Mb – 4 bancs mémoires



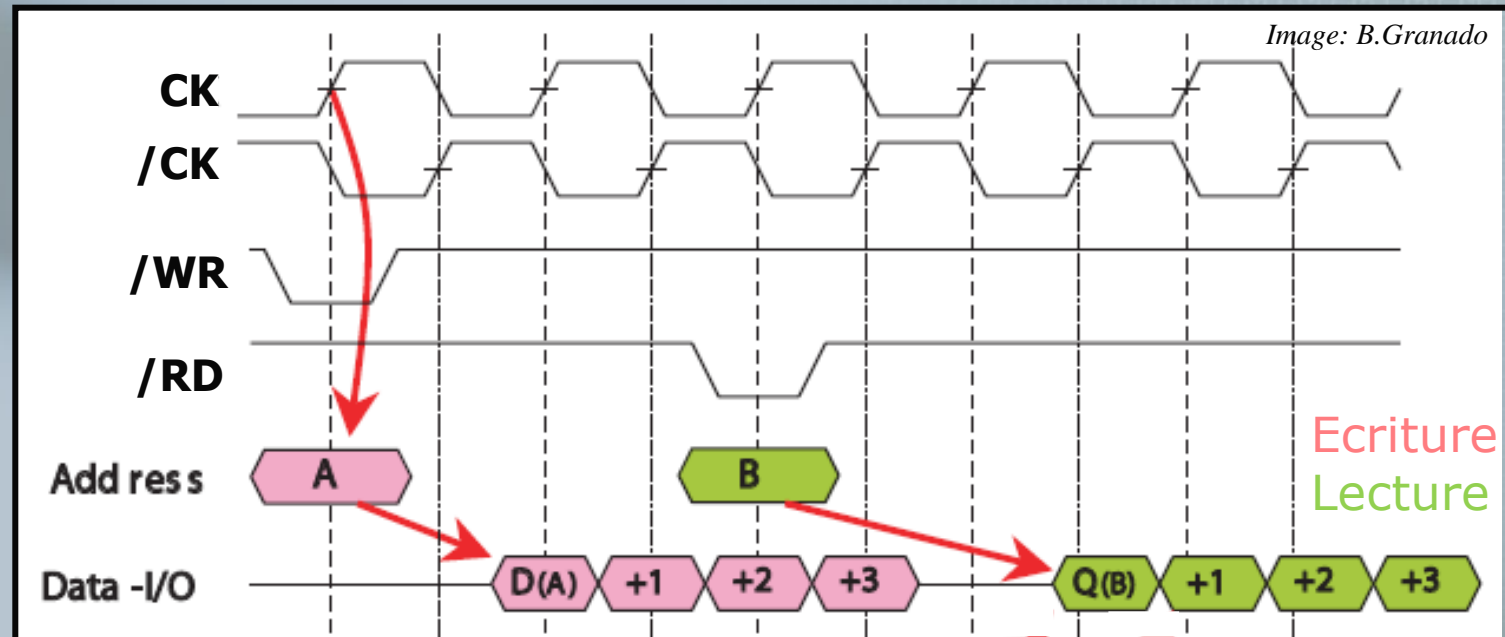
Source: Samsung

C5

53

Exemple: DDR-SDRAM

■ Chronogramme de fonctionnement



C5

54

Technologie DDR-SDRAM

■ Générations

- 1993: SDRAM
- 2000: DDR-SDRAM
- 2003: DDR2-SDRAM
- 2007: DDR3-SDRAM
- 2014: DDR4-SDRAM
- 2021: DDR5-SDRAM

■ A chaque nouvelle génération

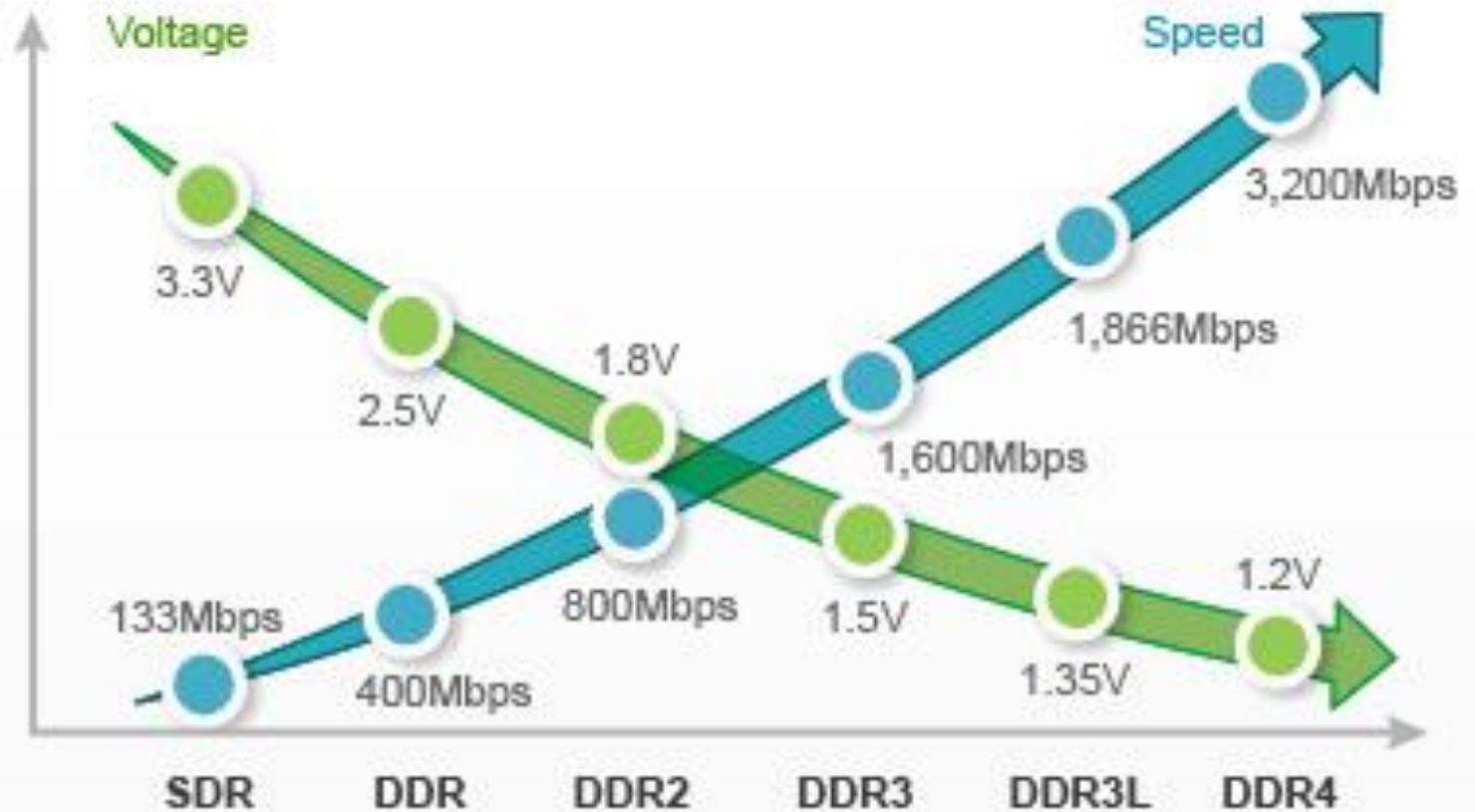
- Doublement des débits
- Abaissement des tensions d'alimentation

C5

55

Bilan DDR-SDRAM

Source: Samsung



C5

56

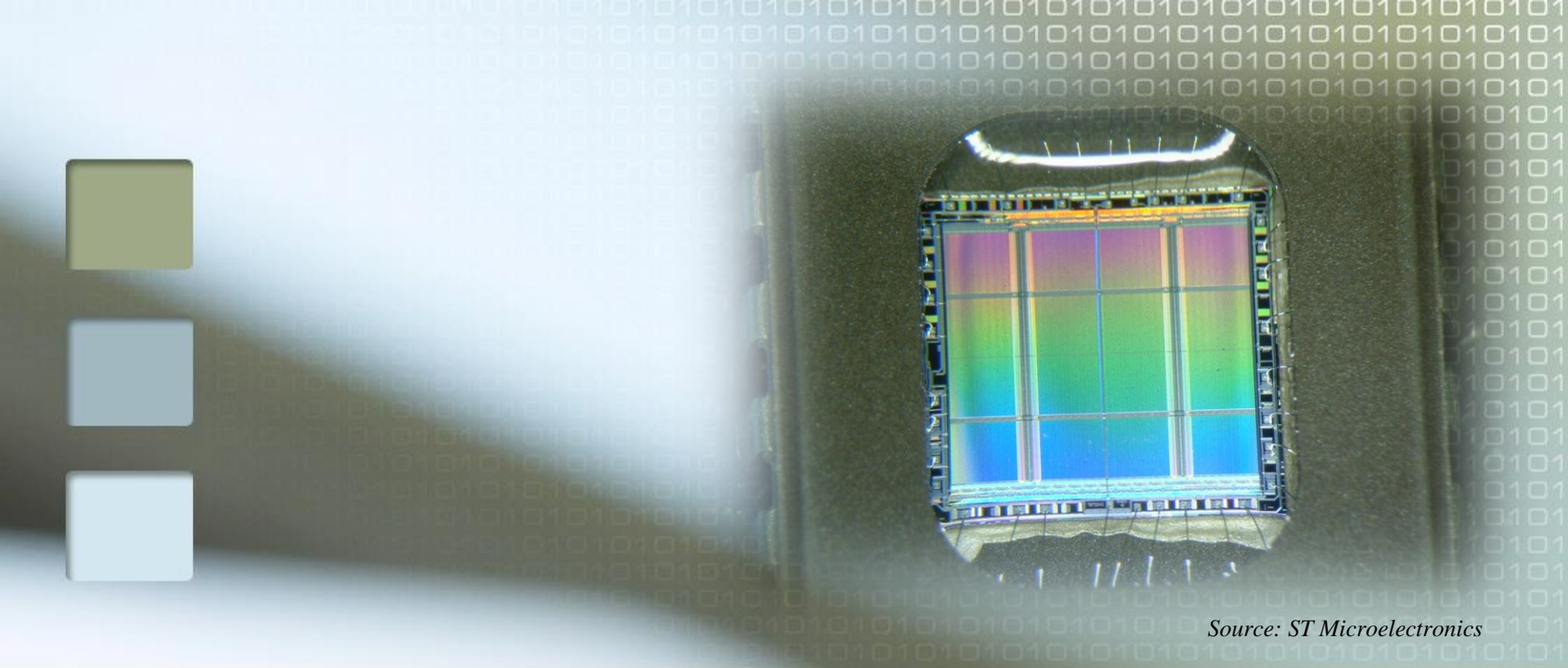
Bilan DDR-SDRAM

Source: Samsung



C5

57



Source: ST Microelectronics

Mémoires ROM

MROM, PROM, EPROM, EEPROM...

Mémoires ROM

- Mémoire non volatile
 - Données conservées même sans alimentation
- Applications
 - BIOS ordinateurs
 - Reconfiguration de FPGA
 - Stockage de programmes dans un système embarqué

Technologies Mémoires ROM

- Mask – ROM
 - Réalisé lors de la fabrication du circuit
- PROM
 - Fusible
 - Programmable une fois
- (E)EPROM
 - Transistors à Grille Flottante
 - Reprogrammable

FLEXIBILITE

C5

60

Mask - ROM

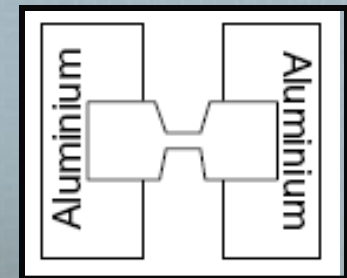
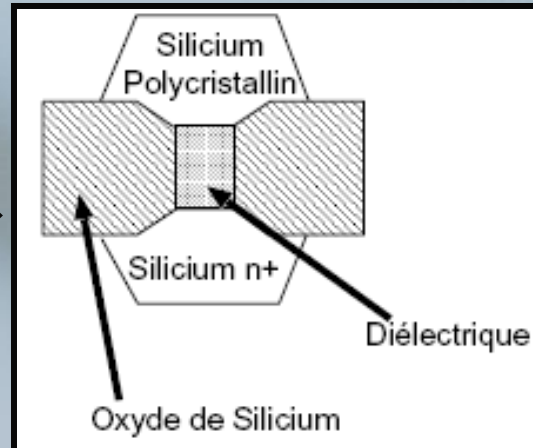
- Aucune souplesse
- Faible coût

C5

61

Programmable ROM

- Composant programmable une fois
- L'état de chaque bit est verrouillé par un fusible ou un antifusible



- Programmation en appliquant de fortes tensions pour claquer les fusibles
- Faible utilisation

C5

62

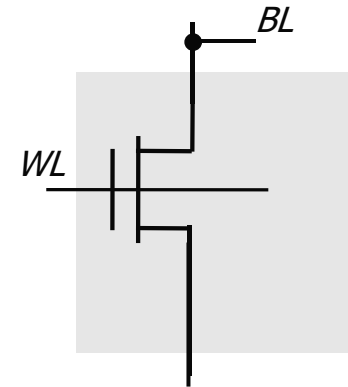
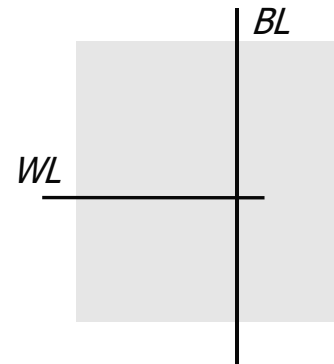
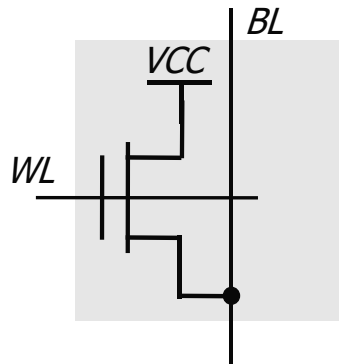
Images: B.Granado

Cellules ROM

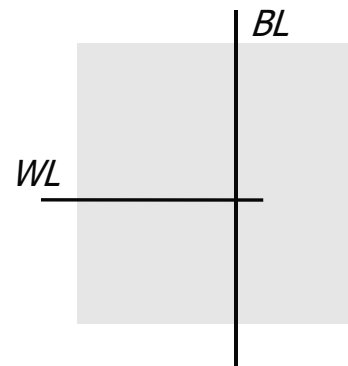
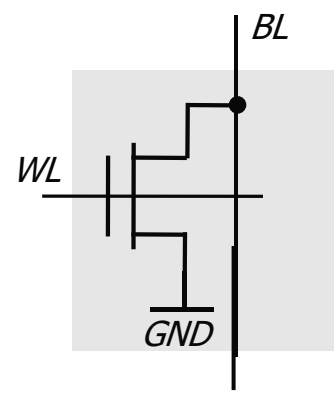
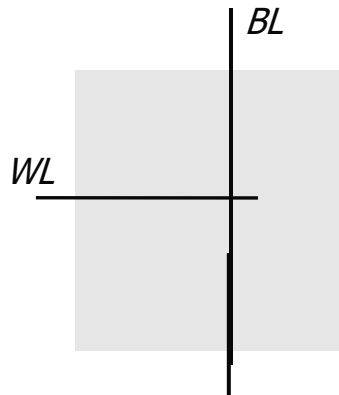
*Valeur du
point mémoire*

Technologie

1



0



OR ROM

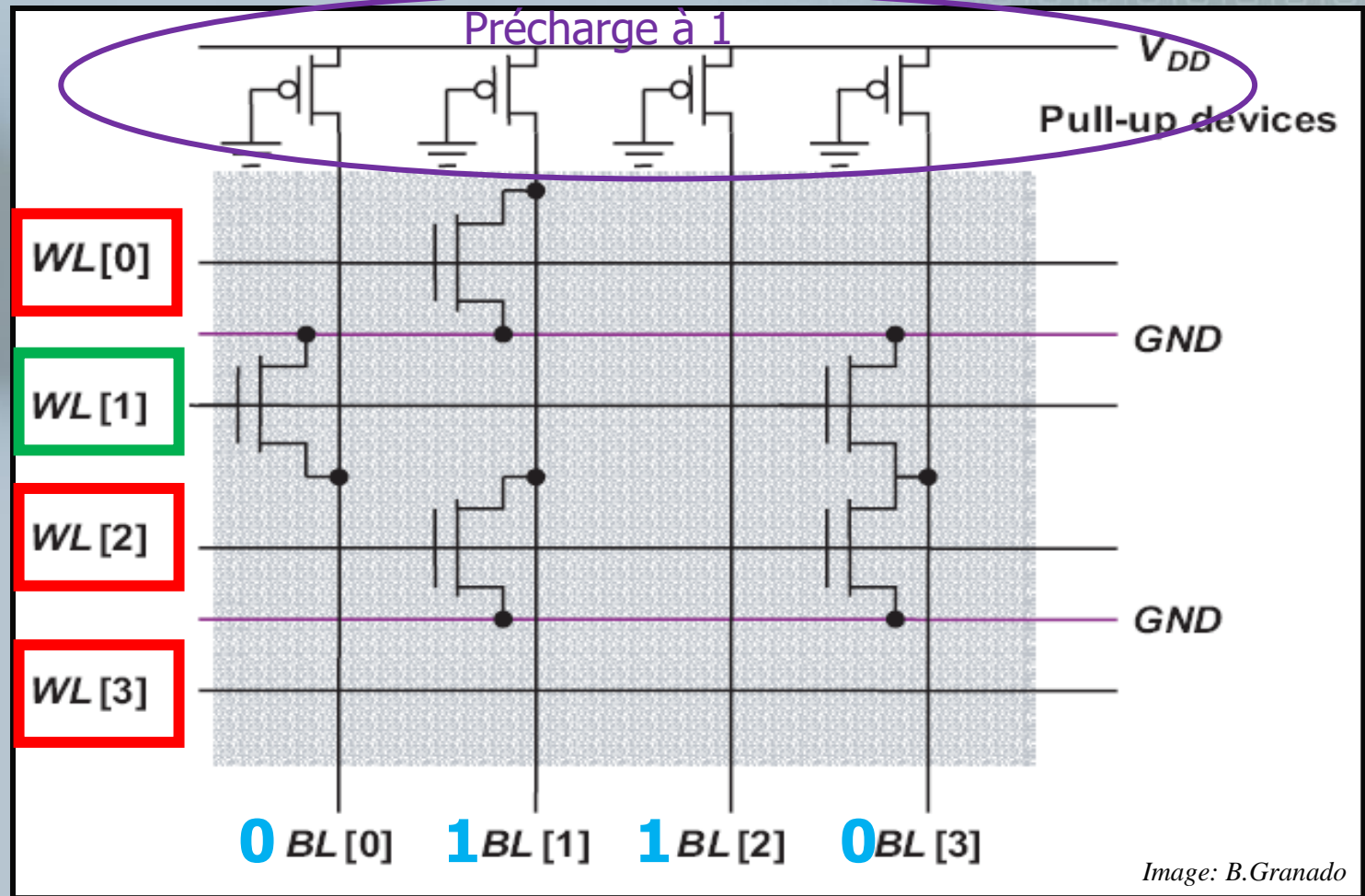
NOR ROM

NAND ROM

C5

63

Tableau Mémoire NOR ROM

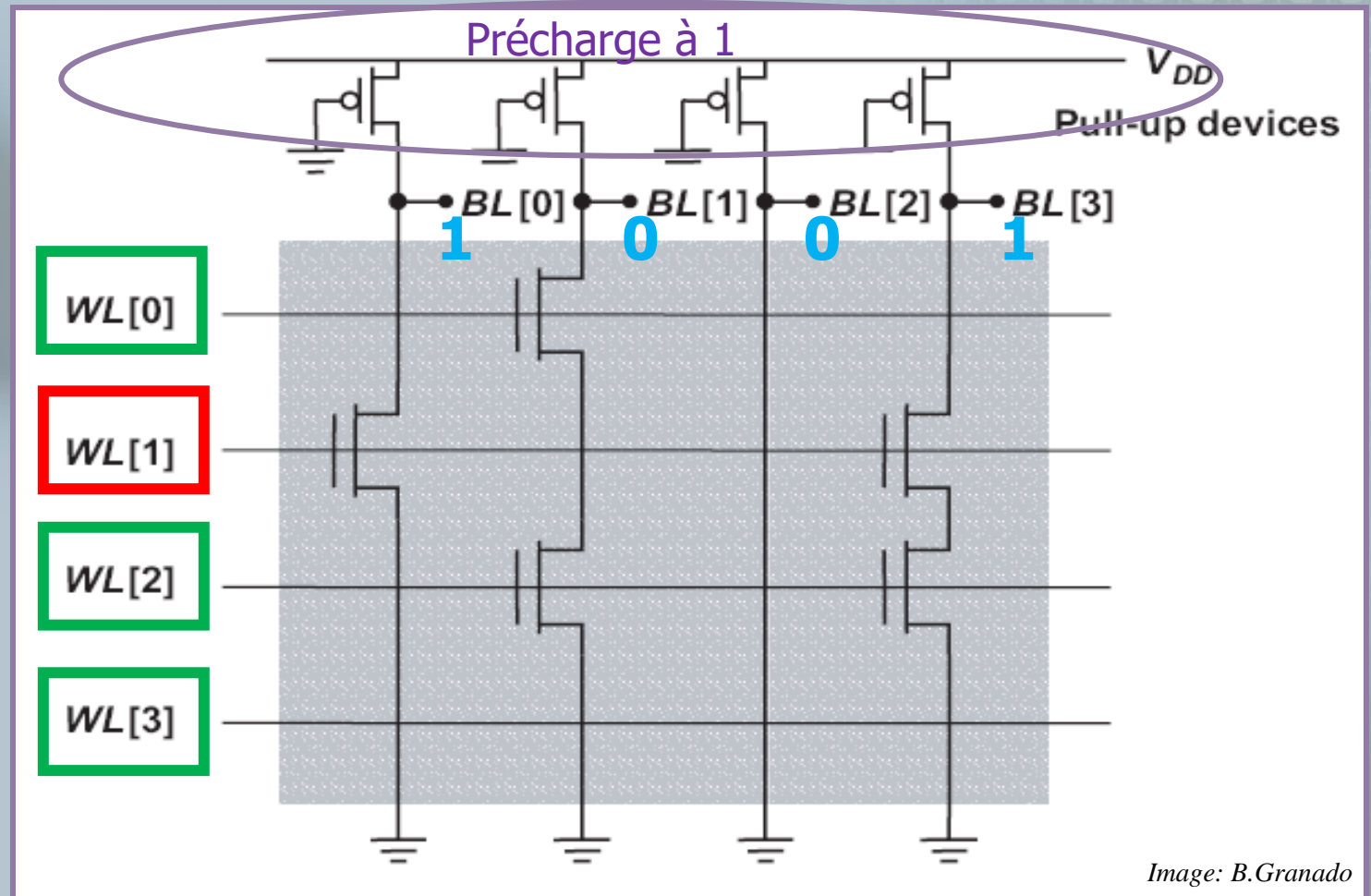


- Mise à 1 du WL de la cellule à lire

C5

64

Tableau Mémoire NAND ROM



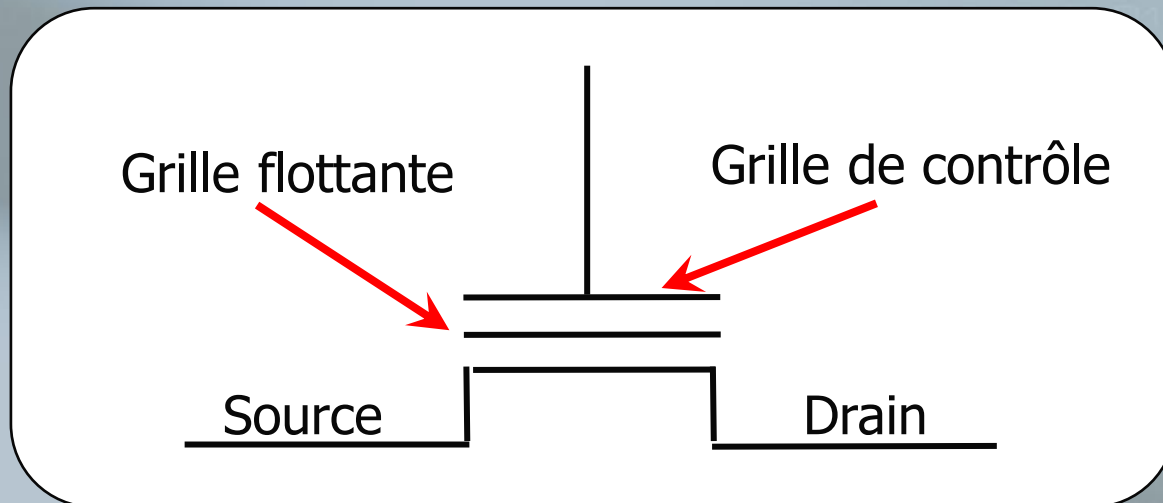
- Mise à 1 de tous les WL sauf celui de la ligne à lire

C5

65

Reprogrammabilité: Technologies EPROM

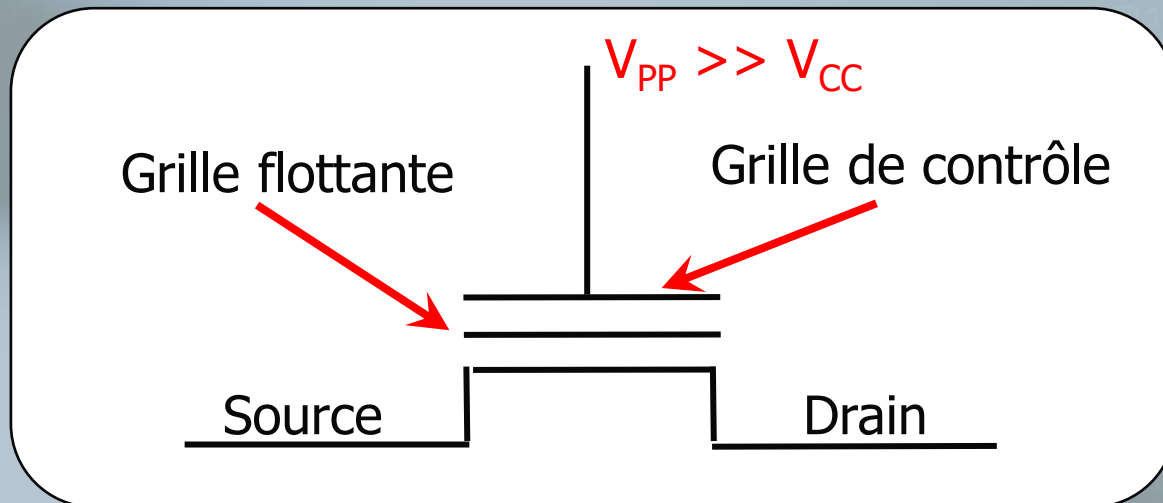
- Cellule mémoire
- Transistor à Grille Flottante (GF)



- Des électrons peuvent être piégés dans la GF
 - GF sans électrons: Tension de seuil = V_{T1}
 - GF avec électrons: Tension de seuil = $V_{T2} > V_{T1}$

Transistor Grille Flottante

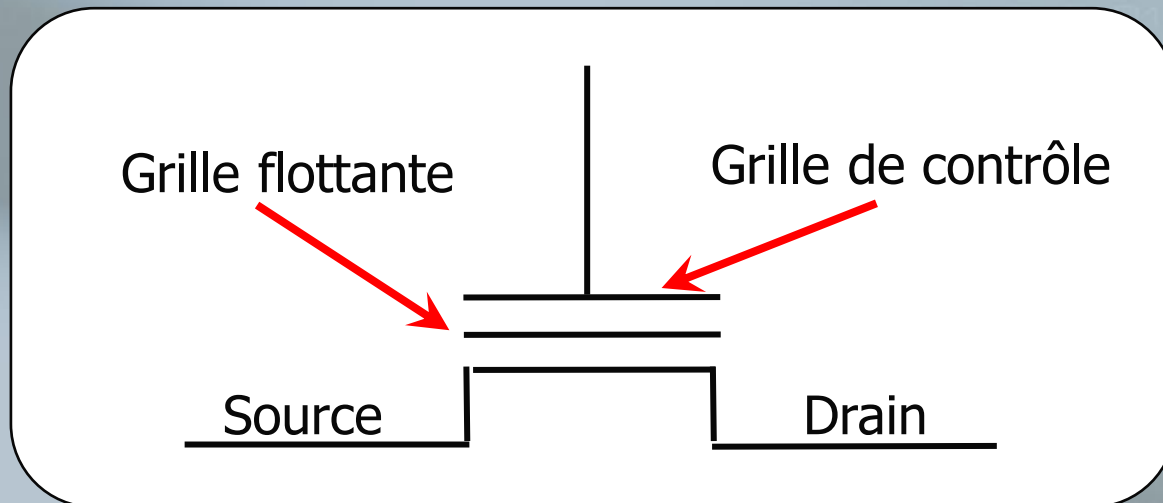
■ Programmation



- Application d'une "forte" tension V_{PP}
 - Transistor passant → Courant entre la source et le drain
 - Des électrons passent dans la grille flottante

Transistor Grille Flottante

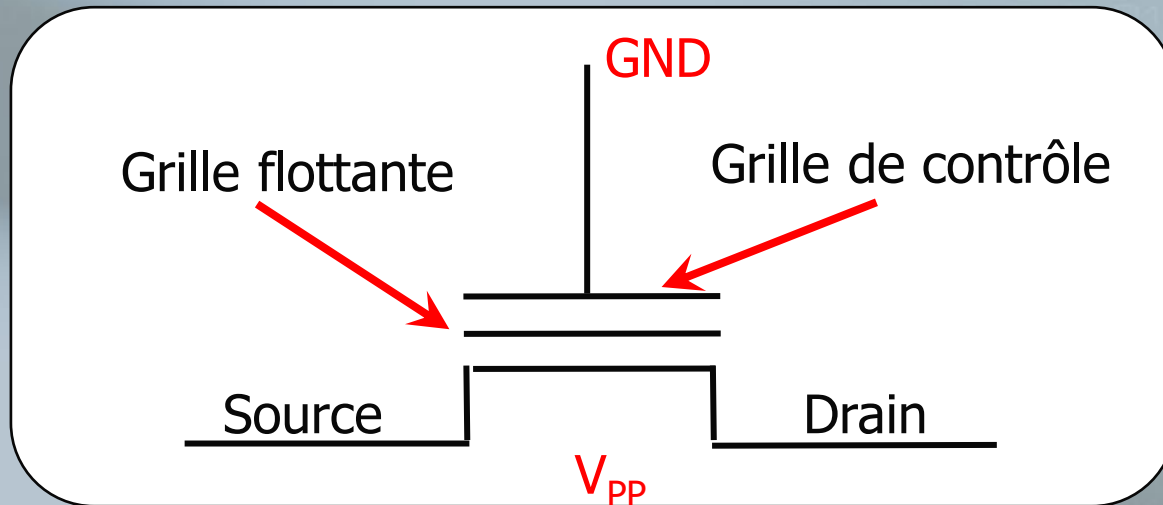
■ Lecture



- La grille est mise à une tension V_G ($V_{T1} < V_G < V_{T2}$)
- Selon la tension de seuil programmée, le transistor est passant ou bloqué

Transistor Grille Flottante

■ Effacement



- Les électrons de la GF partent dans le substrat du transistor

Technologies EPROM

- UV-EPROM:

- Effacement complet par ultra-violets

- EEPROM:

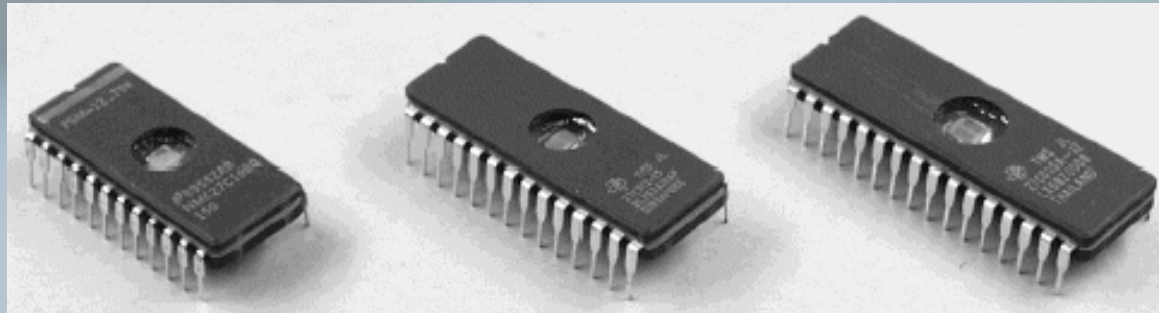
- Effacement électrique par mot mémoire

C5

70

UV-EPROM

- Programmation et effacement hors système à l'aide d'un appareil spécifique



C5

- Les temps de programmation/effacement sont importants (plusieurs minutes) et concernent l'intégralité du tableau mémoire

71

EEPROM

- Programmation et effacement in situ
- Programmation/Effacement par mot
 - Logique de contrôle pour chaque mot pour gérer lecture, écriture, effacement
- Temps d'effacement rapide
- Coût plus élevé
 - Comparé aux mémoires Flash

C5

72



Source: Wikipedia

Mémoires Flash

Mémoires Flash

- Programmation et effacement in situ
- Programmation/Effacement par bloc
 - Logique de contrôle commune à un bloc
- Temps d'effacement très rapide
- Coût limité
 - Comparé aux mémoires EEPROM

C5

74

Technologies Flash

■ Deux grandes familles

Source: eetimes.com

	NAND	NOR
Cell Array		
Layout		
Cross-section		
Cell size	$4F^2$	$10F^2$

C5

75

Technologies Flash

- Possibilité de stocker plusieurs charges par cellule
 - SLC: Single Level Cell
 - MLC: Multi Level Cell

SLC		MLC
Features		
1	Bits per cell	2
3.3V, 1.8V	Voltage	3.3V
x8, x16	Data width (bits)	x8
Architecture		
1 or 2	Number of planes	2
2KB or 4KB	Page size	2KB or 4KB
64	Pages per block	128
Reliability		
4	NOP (partial page programming)	1
1	ECC (per 528 bytes)	4+
~100K	Endurance (ERASE / PROGRAM cycles)	~10K
Array Operations		
25µs	tR (Max)	50µs
200–300µs	tPROG (Typ)	600–900µs
1.5–2ms	tBERS (Typ)	3ms

MLC Density is 2 times that of similar SLC

SLC Requires less ECC

SLC Reliability is 10 times better !

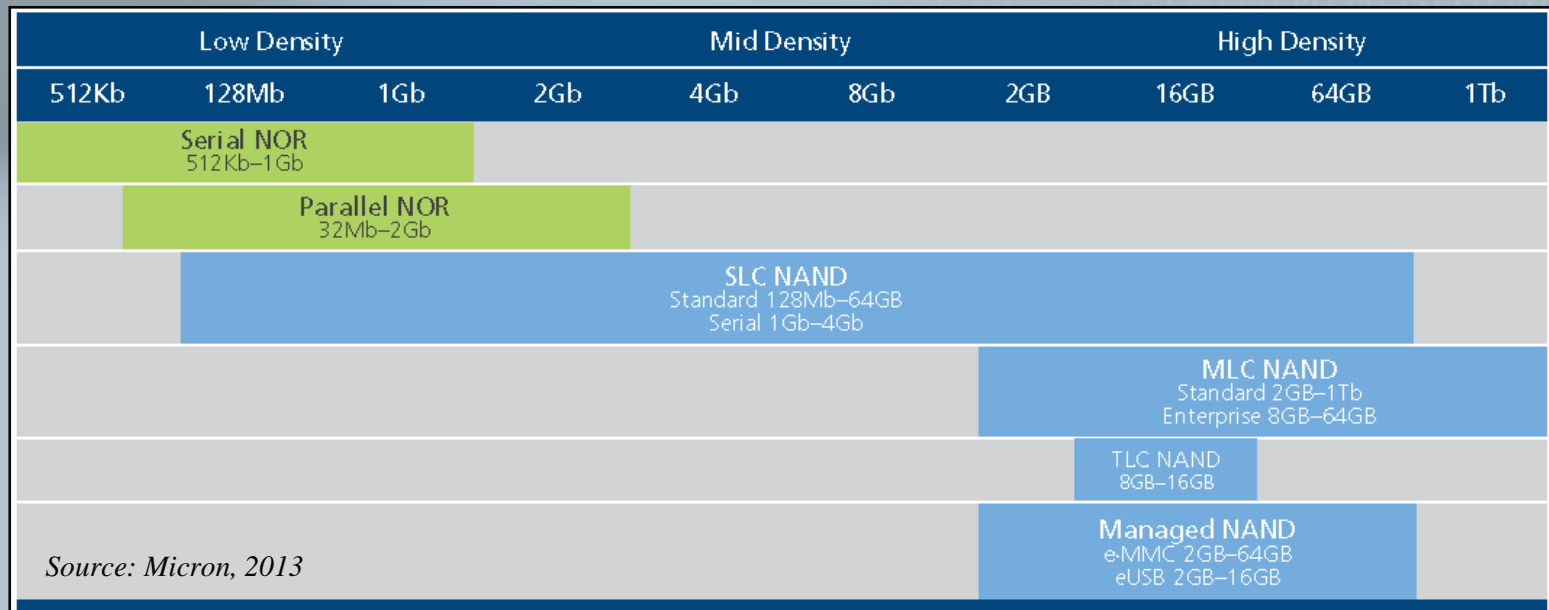
SLC Performance is ~3 times better

Source: Micron

C5

76

Technologies Flash



C5

77

Technologies Flash

Table 1. NAND SLC vs. NOR Typical Characteristics

Characteristic	NOR	SLC NAND
Density	1Mb -2Gb	512Mb – 8Gb
Random Read Latency	0.1us ^{Note 1}	25us ^{Note 2}
Sustained Read Speed (x8 I/O)	30MB/s	30MB/s
Read Page buffer	16B ^{Note 1}	2048B ^{Note 2}
Random Write speed	11us ^{Note 1}	250us ^{Note 2}
Write page	64B ^{Note 1}	2048B ^{Note 2}
Sustained Write Throughput	0.5MB/s	8MB/s
Erase speed	0.6s ^{Note 1}	2ms ^{Note 2}
Unit Cell area	10F ²	4F ²

Note 1: MX29GL512F Note 2: MX30LF1G08AA

Table 3. SLC vs. MLC NAND

	SLC	MLC
Density	512Mb – 8Gb	8Gb-64Gb
Tprog (Page Program Time)	250us	900us
Tread (1 st Byte Read Latency)	25us	50us
NOP (No Partial Prgm in Pg)	4	1
Endurance (W/E cycles)	100K	5K
ECC bits	1 - 8+	4 - 20+

Source: Macronix 2014

NOR

- Lecture aléatoire
- Stockage code (OS)

NAND

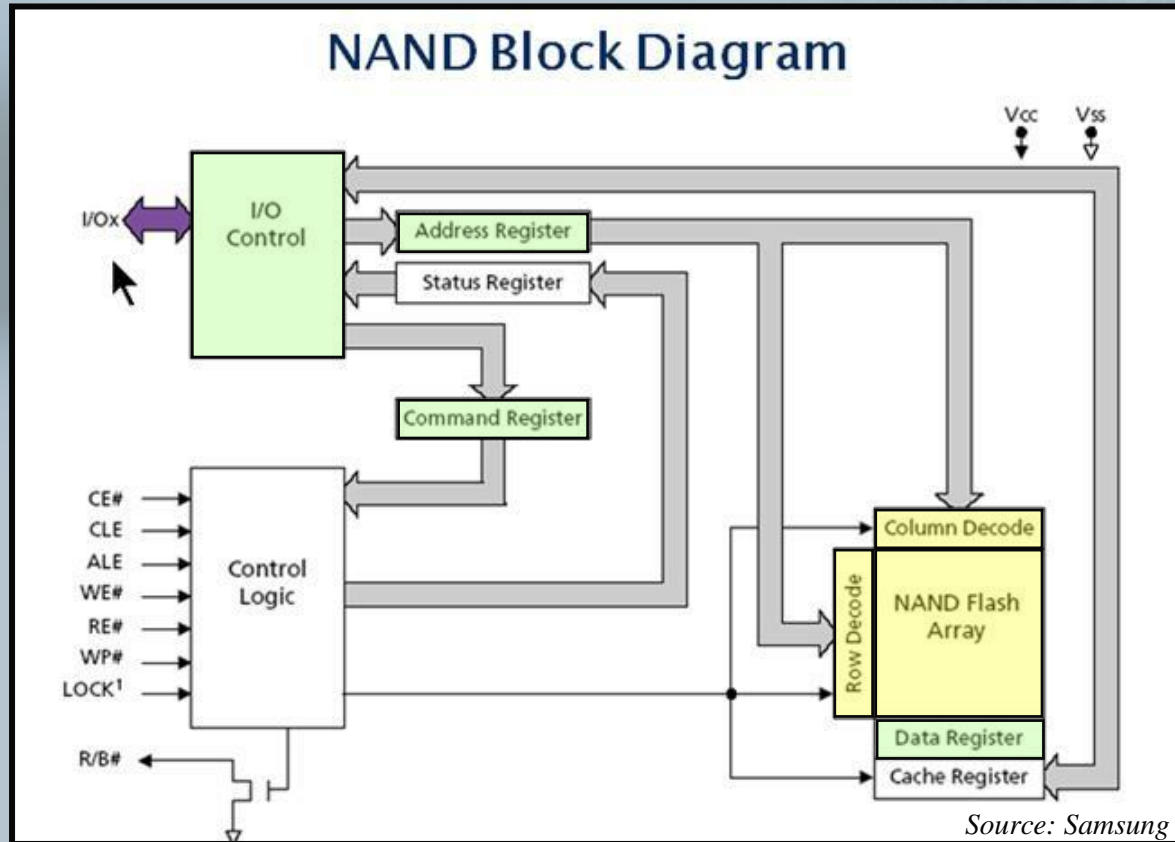
- Lecture séquentielle
- Stockage données

C5

78

Flash NAND

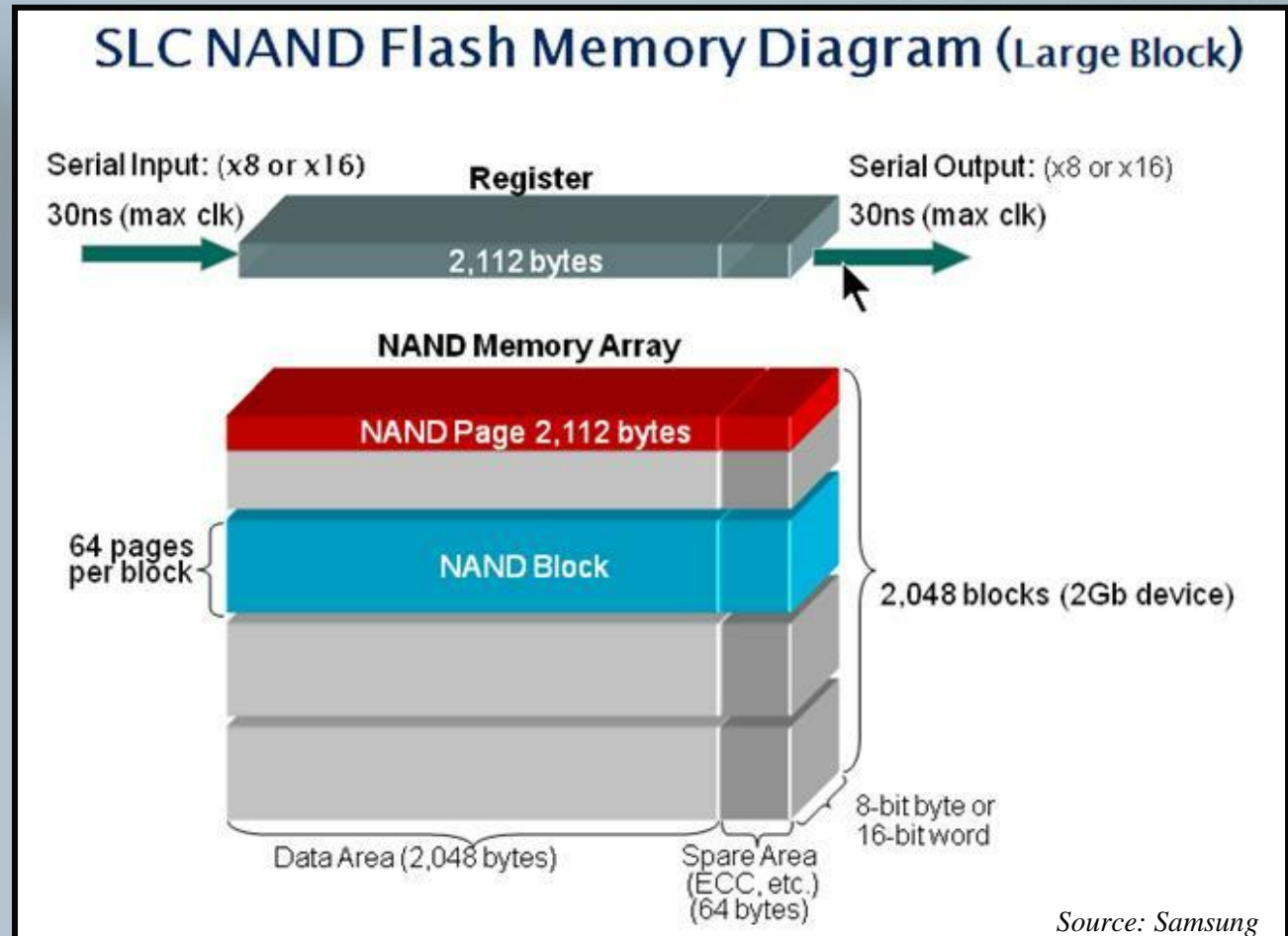
■ Architecture



- Multiplexage bus adresses/données/commandes
- Bus I/O sur 8 ou 16 bits

Flash NAND

■ Tableau mémoire



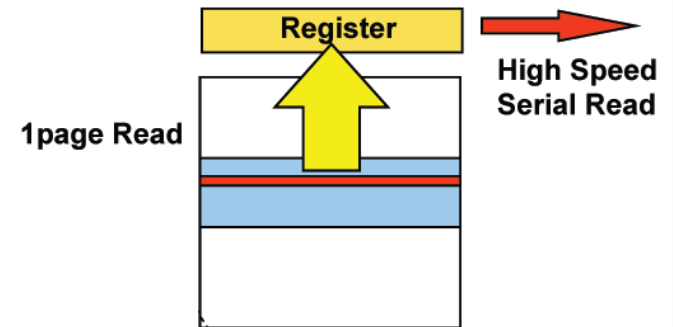
C5

80

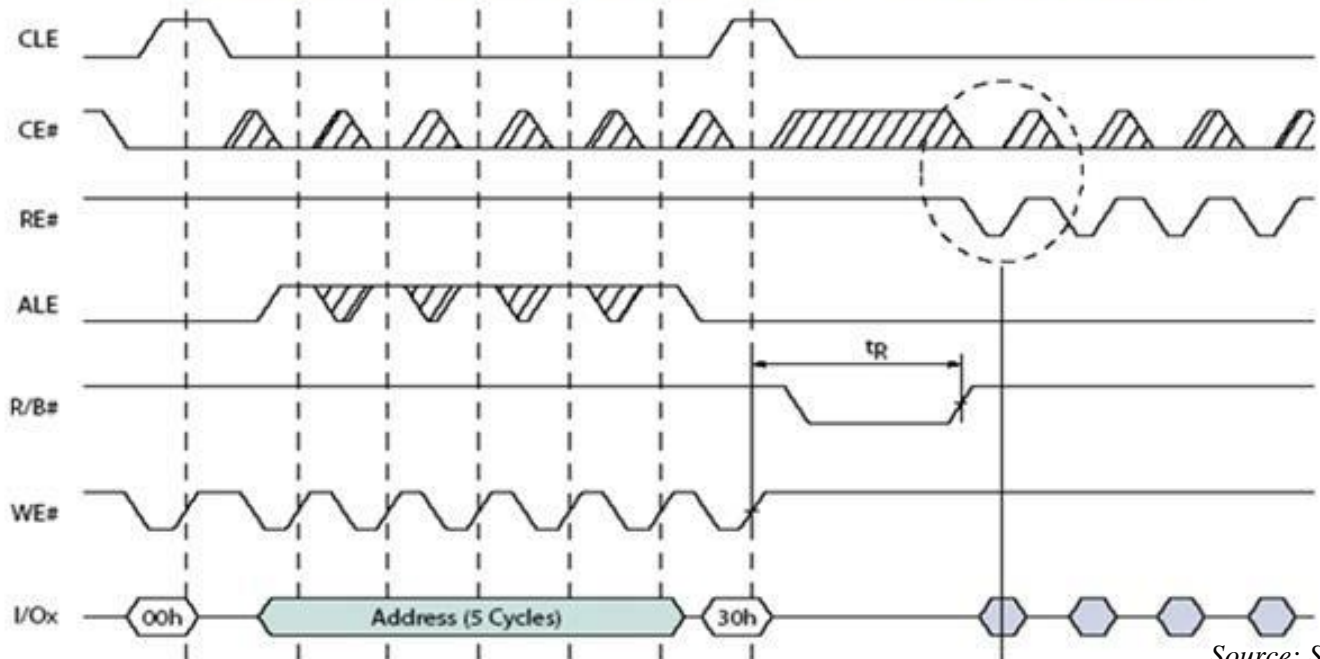
Flash NAND

■ Lecture

Image: B.Granado



Basic Access and “



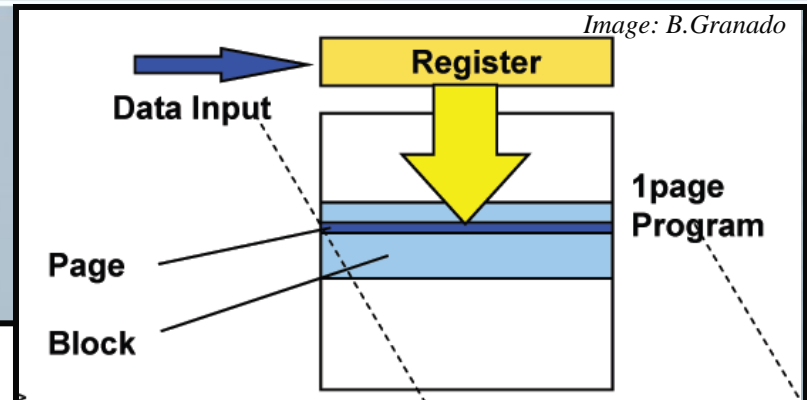
Source: Samsung

C5

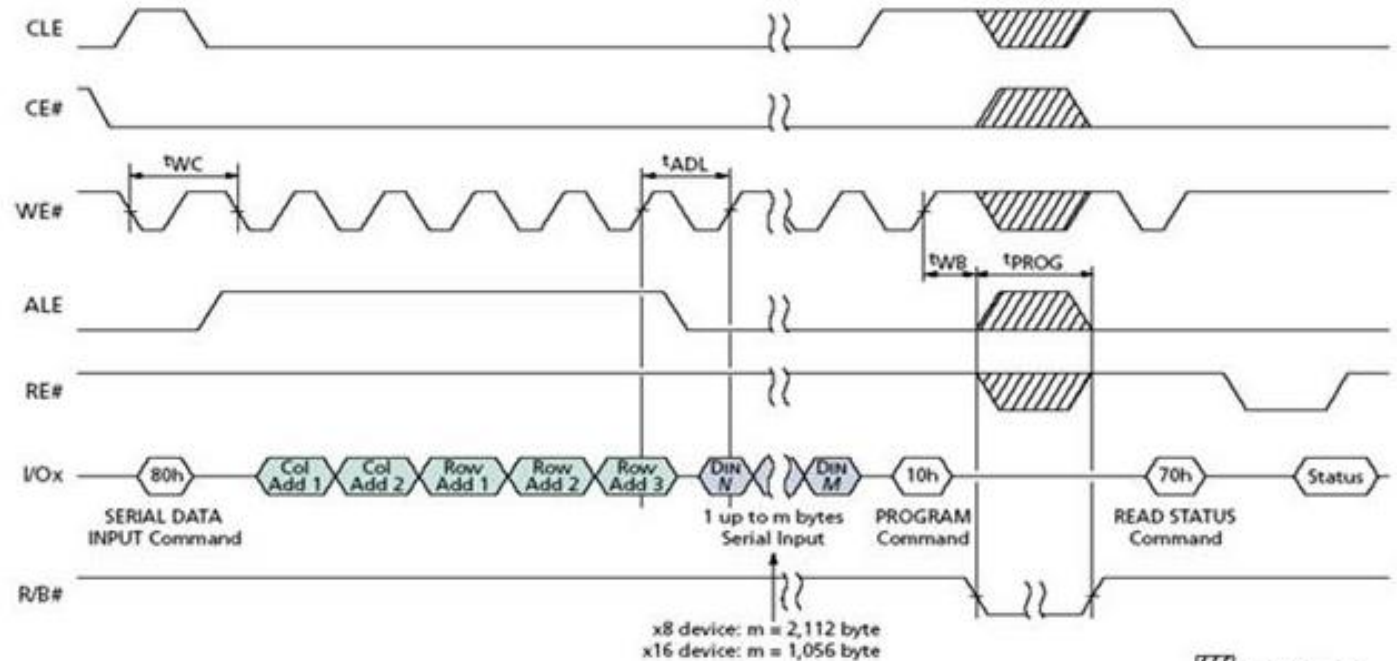
81

Flash NAND

■ Programmation



Program Operation



Source: Samsung

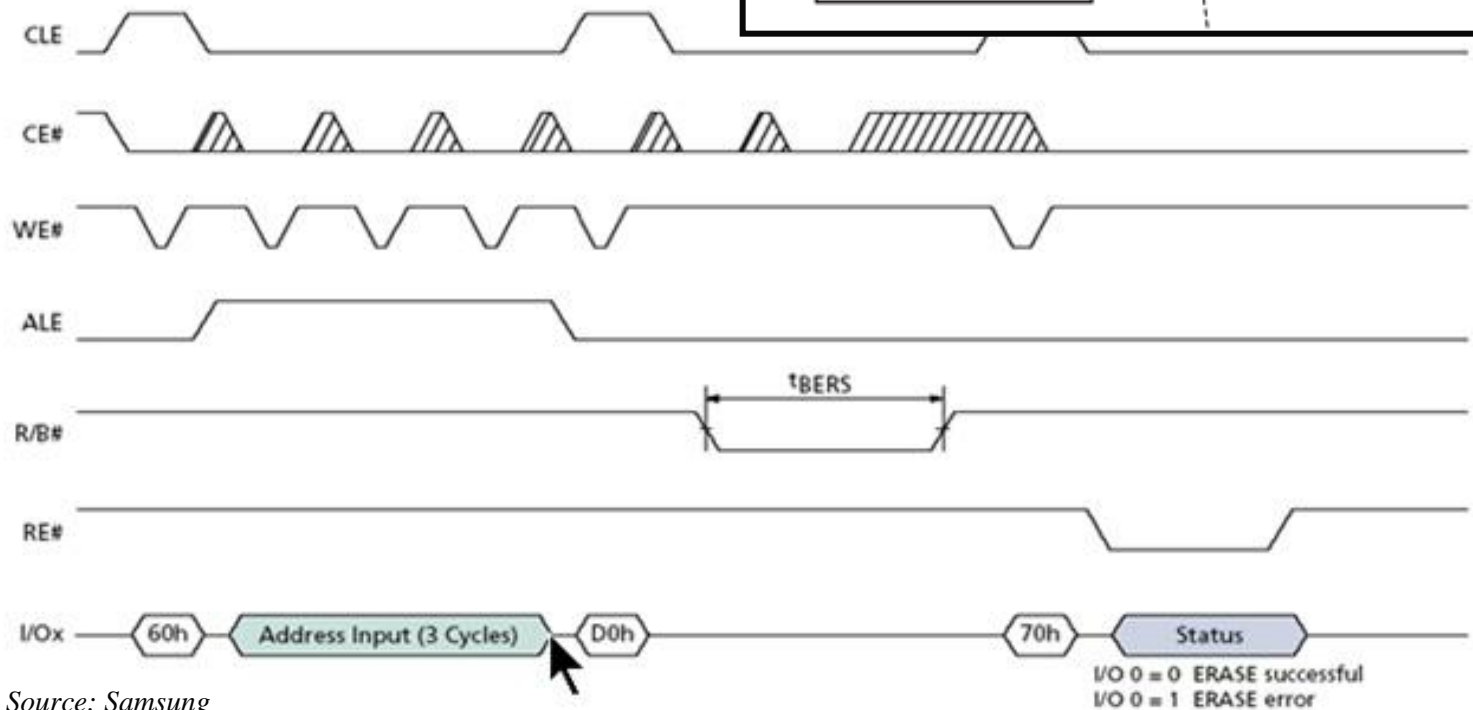
C5

82

Flash NAND

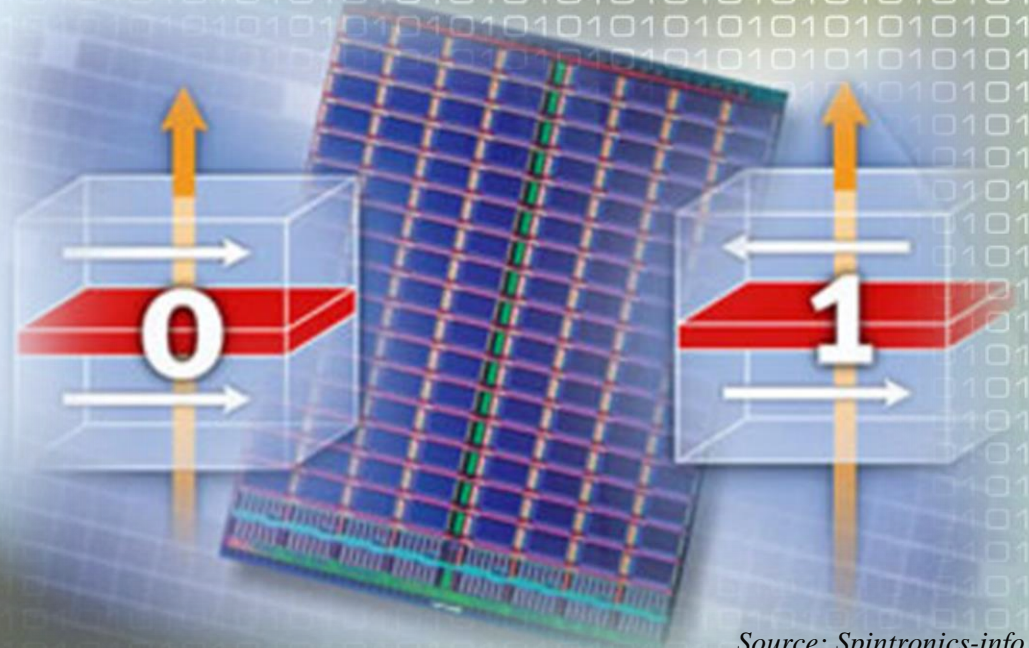
■ Effacement

Block Erase Operation



C5

83



Source: Spintronics-info

Mémoires MRAM

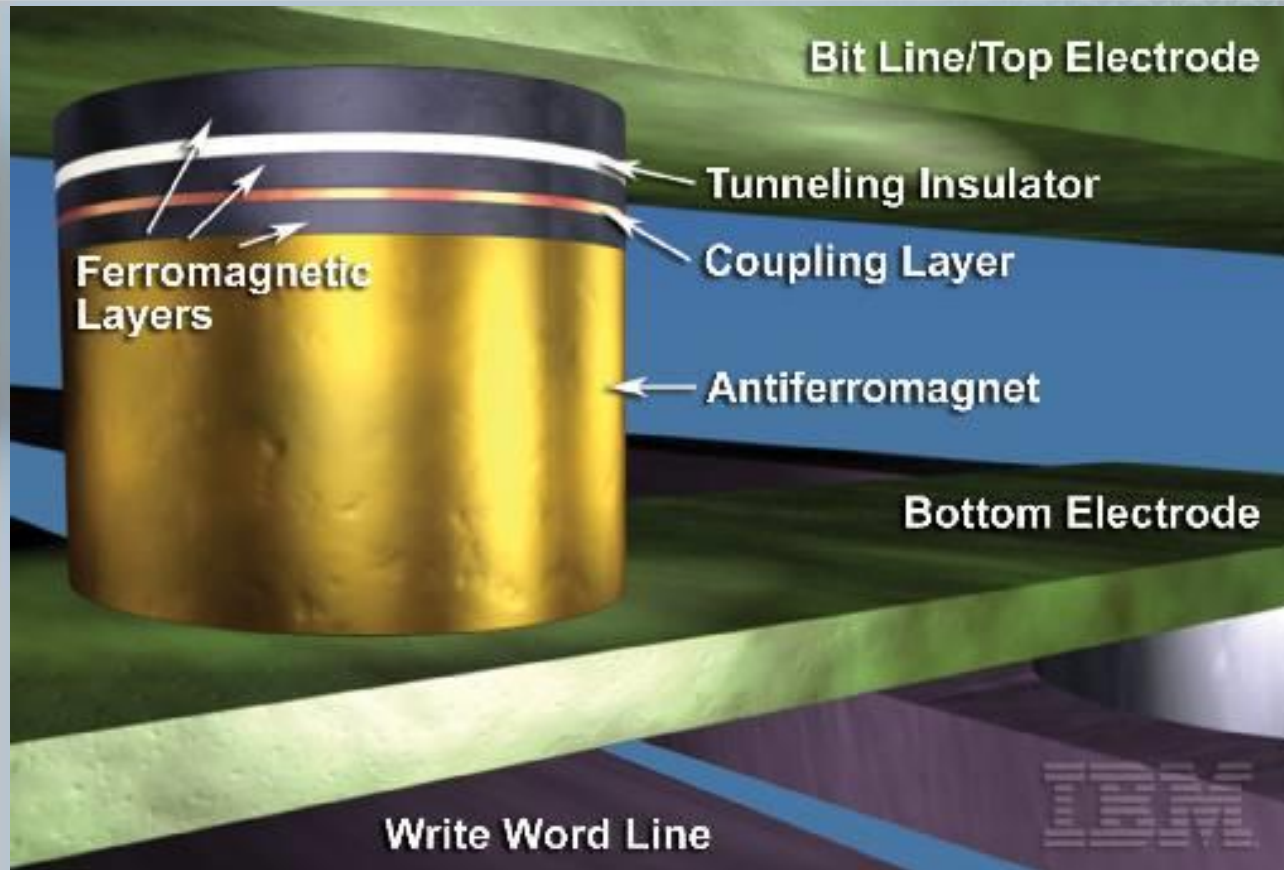
Technologies MRAM

- Concept initié dans les années 1990
- Objectif: allier les avantages des
 - SRAM: rapidité
 - DRAM: densité
 - Flash: non volatilité

C5

85

Principe

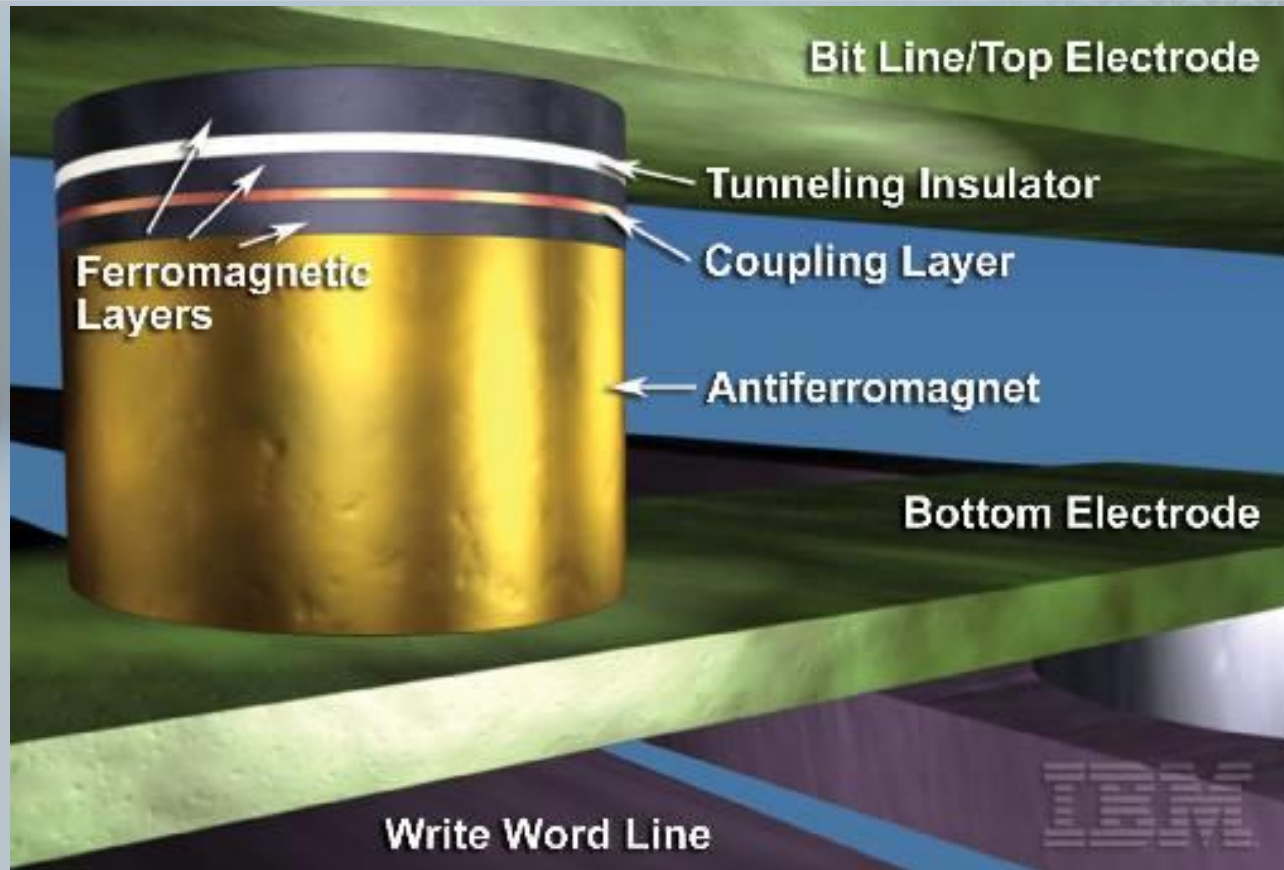


C5

86

- Cellule mémoire → 2 Couches ferromagnétiques
 - Polarisation fixe pour l'une (Word Line)
 - Polarisation variable pour l'autre (Bit Line)

Principe

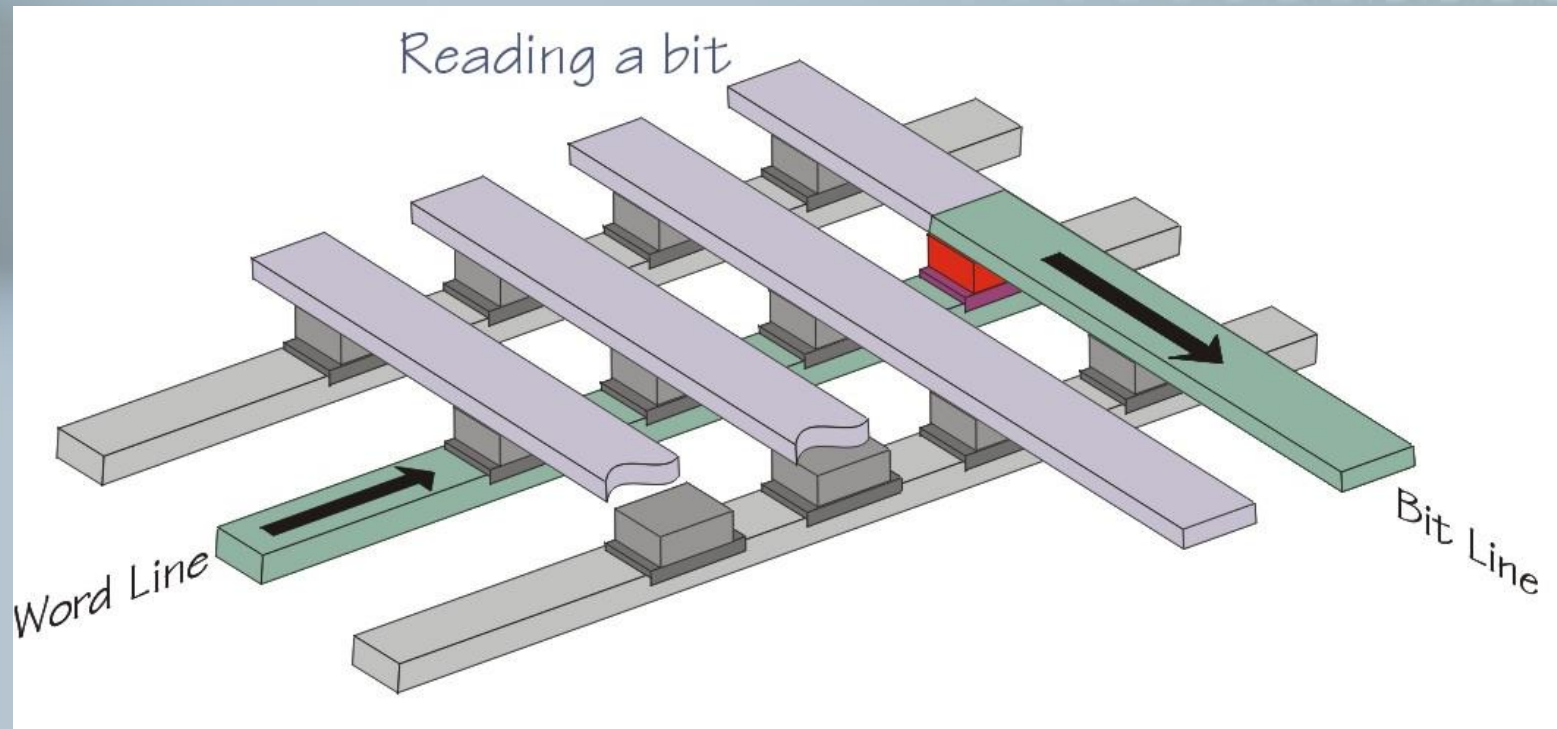


C5

87

- La polarisation fait varier la résistivité de la cellule
 - Le calcul de la résistance (ou du courant qui va traverser le point mémoire) indique son état logique

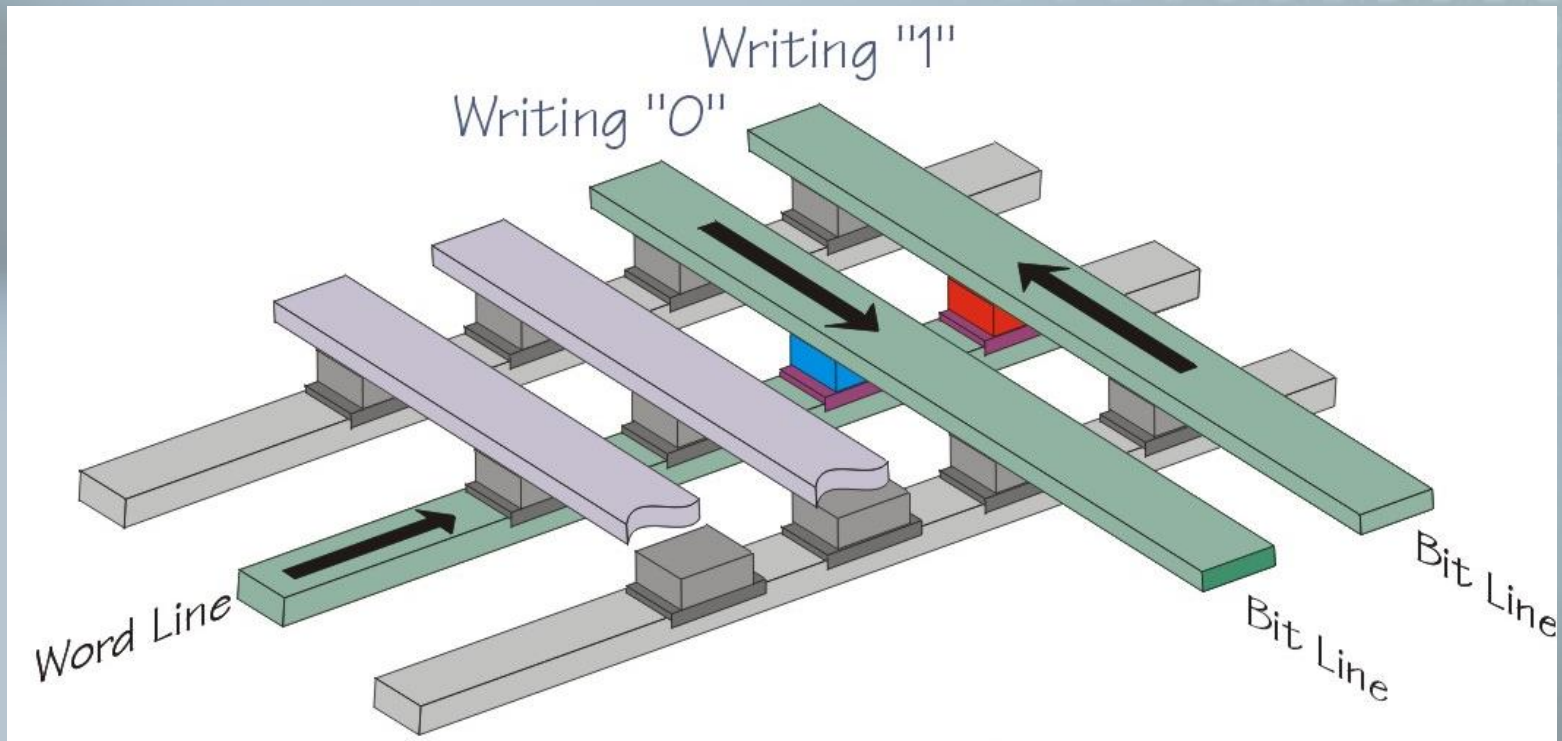
Lecture MRAM



C5

88

Ecriture MRAM



C5

89