2 L'information binarisée

Peter Schlagheck

Université de Liège

2 L'information binarisée

2.1 Le système décimal et le système binaire

Dans le système décimal habituel, les nombres sont encodés avec les chiffres 0, 1, 2, . . . , 9

Exemple:

$$\begin{aligned} 325 &= 3 \times 100 + 2 \times 10 + 5 \times 1 \\ &= 3 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 \end{aligned}$$

2.1 Le système décimal et le système binaire

Dans le système décimal habituel, les nombres sont encodés avec les chiffres 0, 1, 2, . . . , 9

Exemple:

$$\begin{aligned} 325,172 &= 3 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 \\ &\quad + 1 \times 10^{-1} + 7 \times 10^{-2} + 2 \times 10^{-3} \end{aligned}$$

2.1 Le système décimal et le système binaire

Dans le système décimal habituel, les nombres sont encodés avec les chiffres 0, 1, 2, . . . , 9

Exemple:

$$\begin{aligned} 325,172 &= 3 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 \\ &\quad + 1 \times 10^{-1} + 7 \times 10^{-2} + 2 \times 10^{-3} \end{aligned}$$

Représentation “scientifique” sur l’écran d’une calculatrice:

$$\texttt{3,25172 02} \equiv 3,25172 \times 10^2$$

$\longrightarrow$ pratique pour des très grands nombres . . .

$$\texttt{3,25172 20} = 325172000000000000000$$

2.1 Le système décimal et le système binaire

Dans le système décimal habituel, les nombres sont encodés avec les chiffres 0, 1, 2, . . . , 9

Exemple:

$$\begin{aligned} 325,172 &= 3 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 \\ &\quad + 1 \times 10^{-1} + 7 \times 10^{-2} + 2 \times 10^{-3} \end{aligned}$$

Représentation “scientifique” sur l’écran d’une calculatrice:

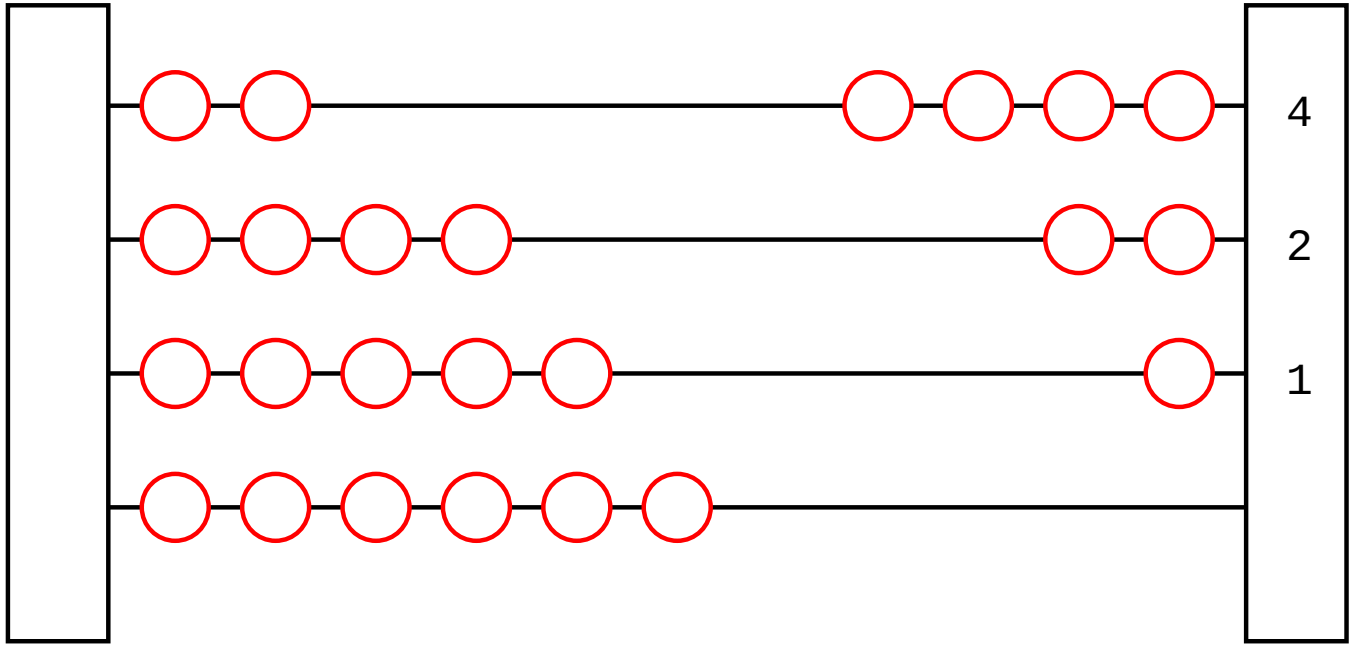
$$\texttt{3,25172 02} \equiv 3,25172 \times 10^2$$

$\longrightarrow$ pratique pour des très grands nombres . . .
et des très petits nombres:

$$\texttt{3,25172 -20} = 0,00000000000000000325172$$

Le système décimal et l'abaque

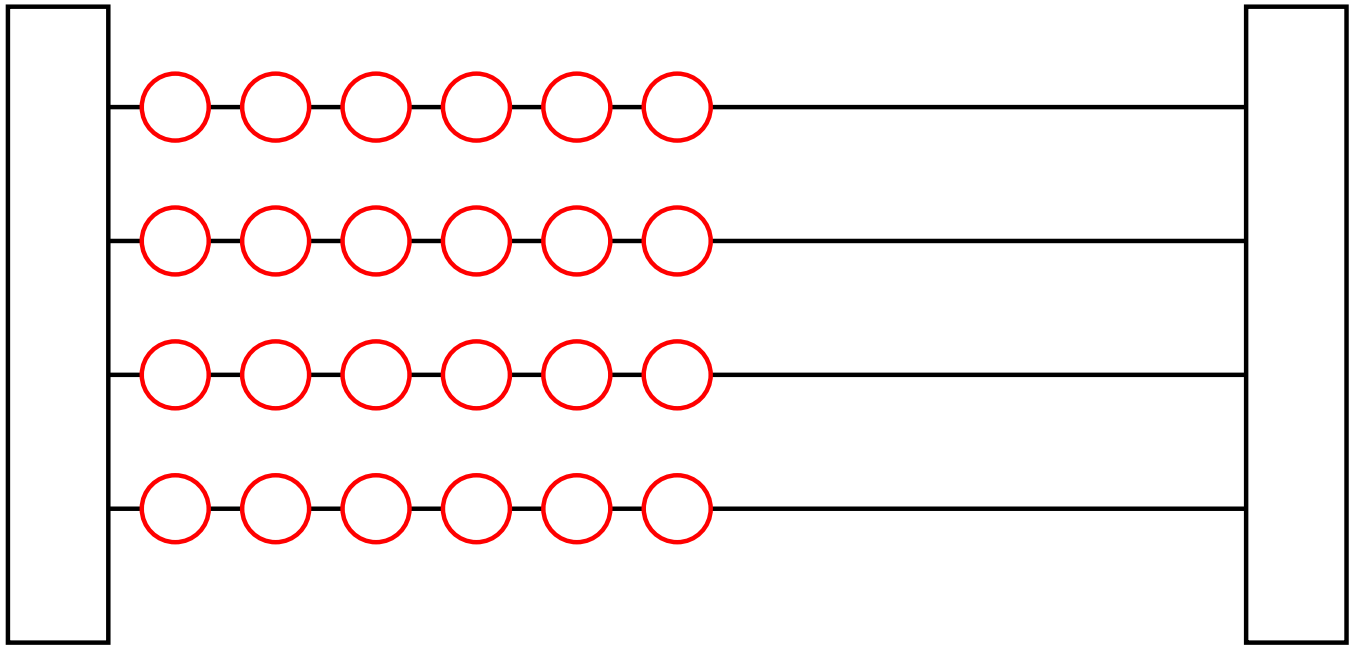
L'abaque à six boules par ligne ?



$$124 \quad = \quad 1 \times 6^2 + 2 \times 6^1 + 4 \times 6^0 = 52$$

Le système hexal

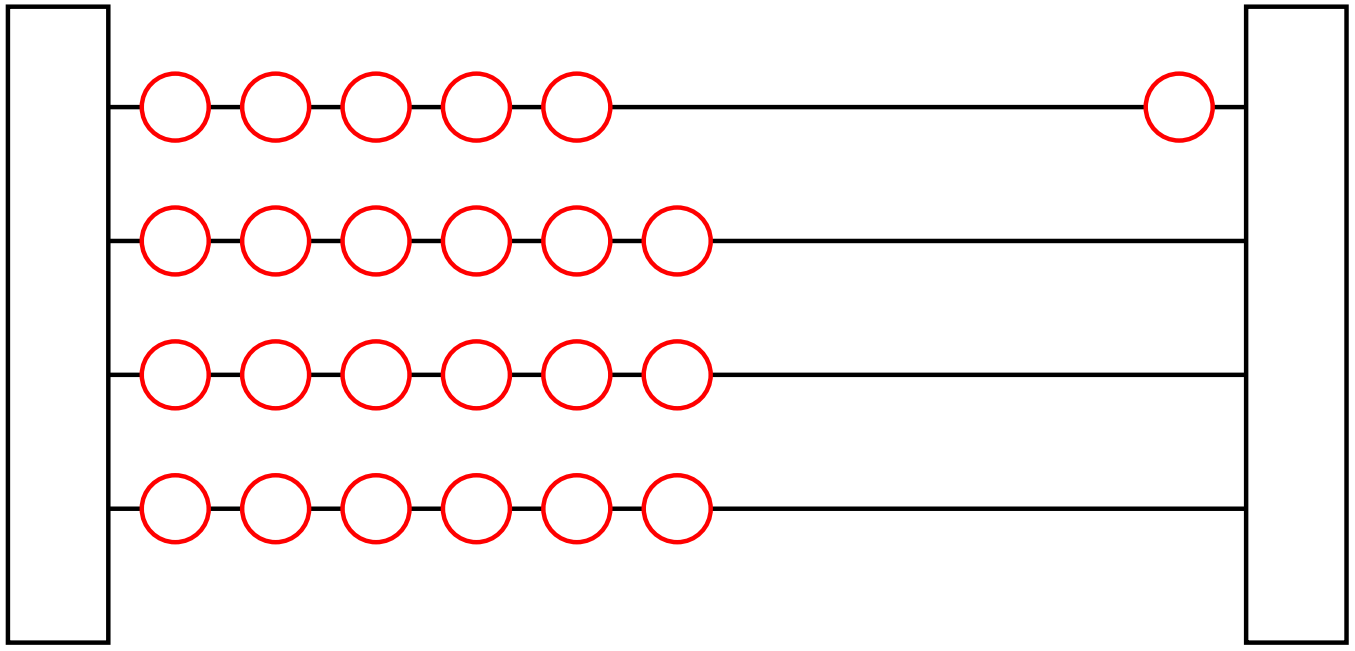
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



0

Le système hexal

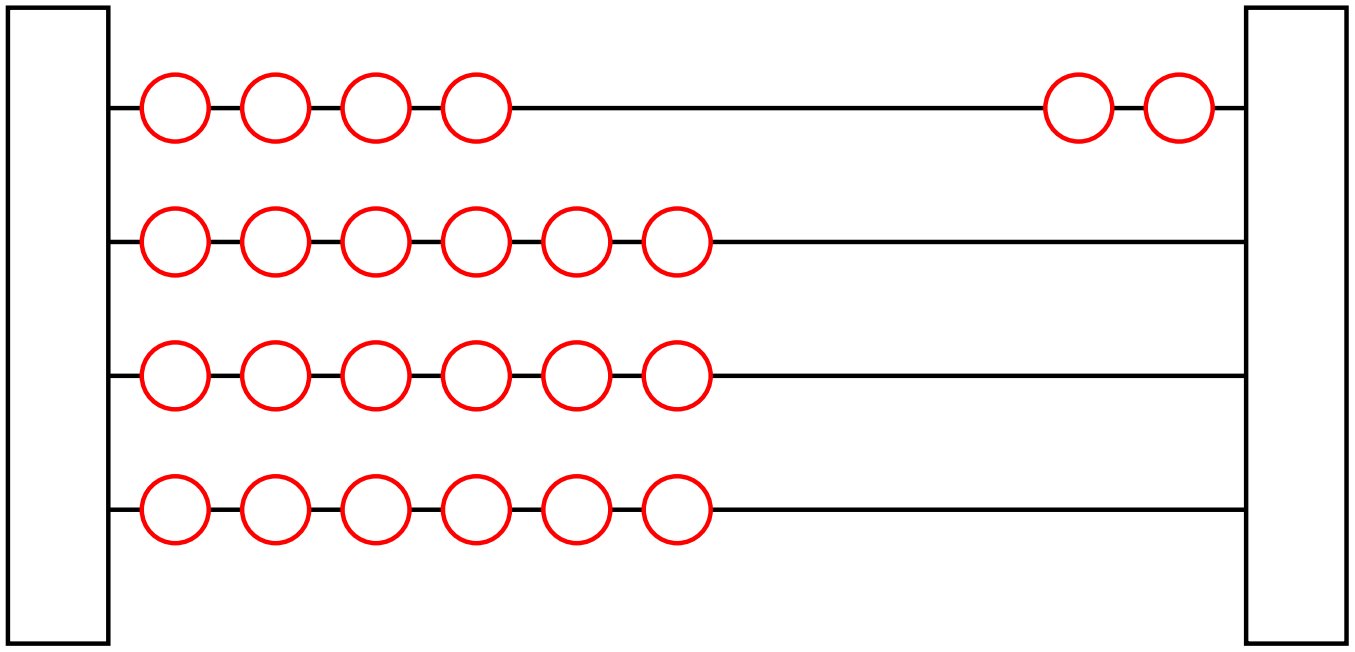
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



1

Le système hexal

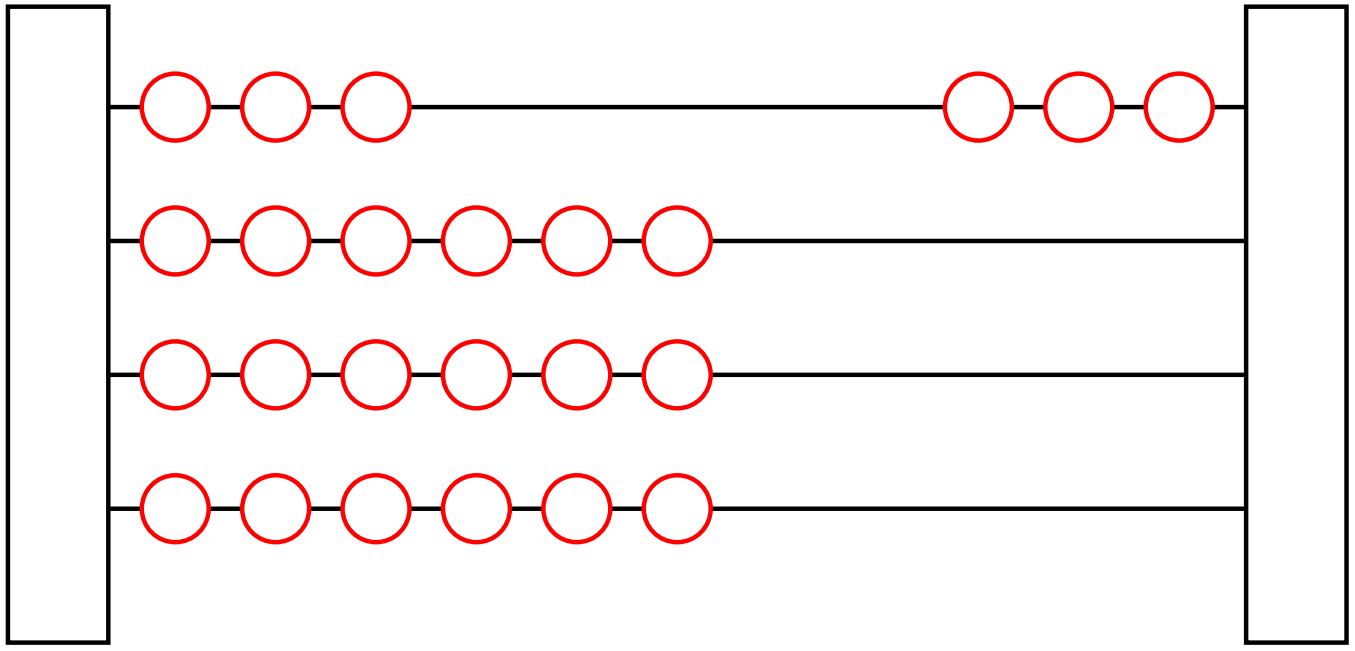
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



2

Le système hexal

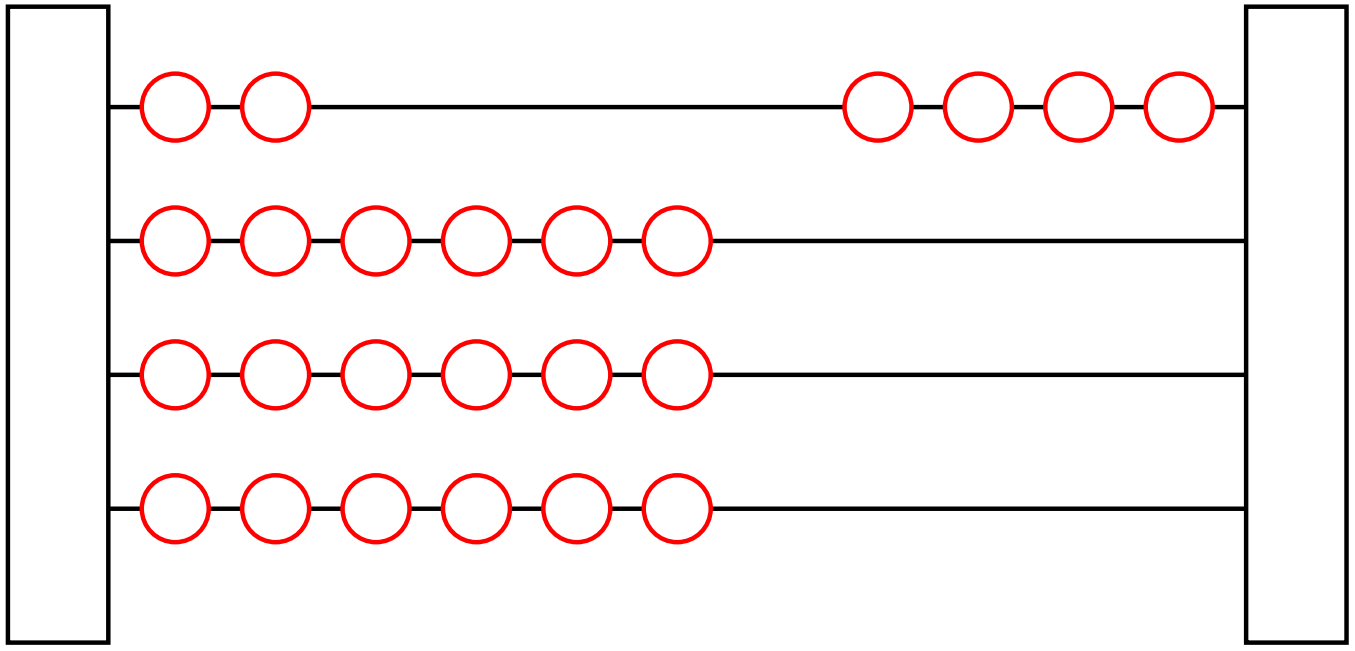
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



3

Le système hexal

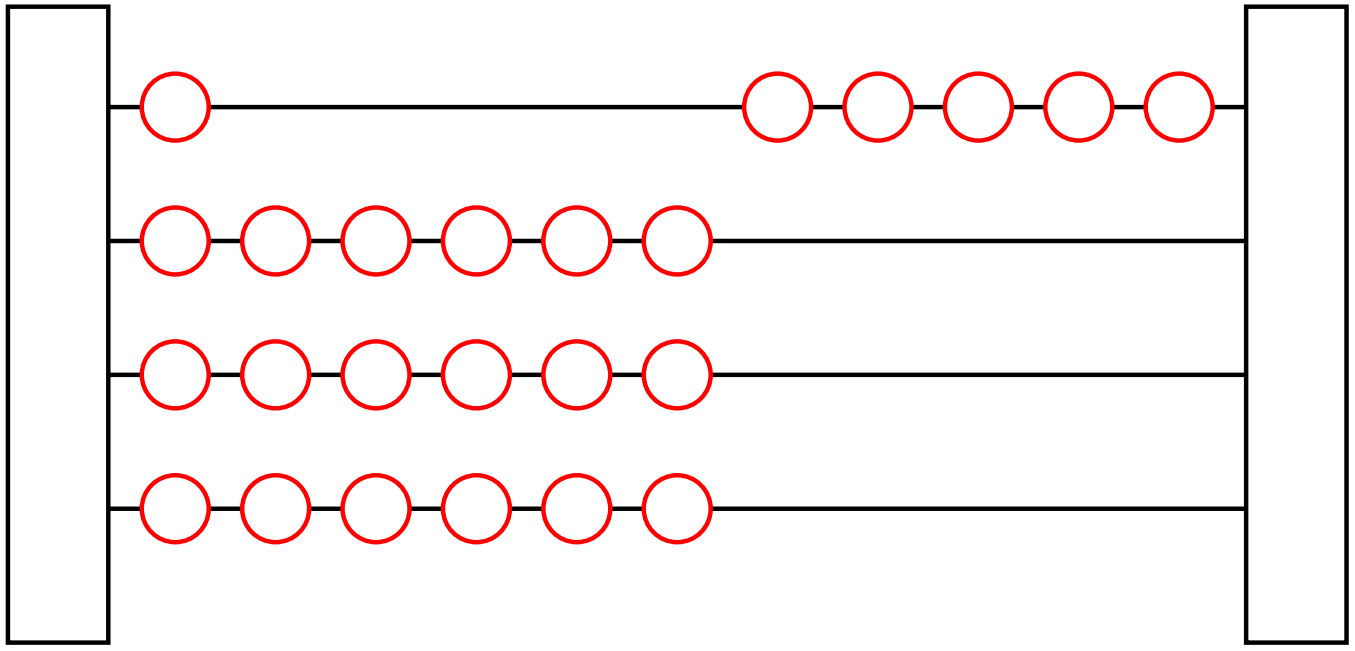
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



4

Le système hexal

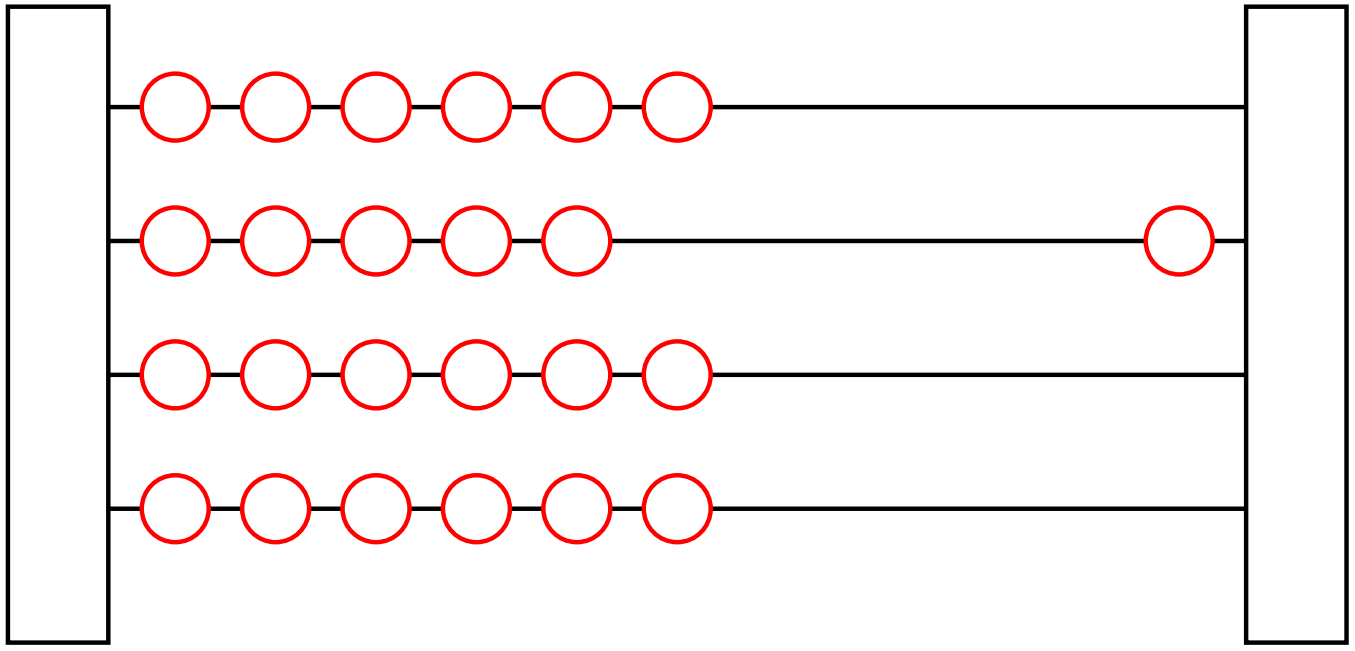
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



5

Le système hexal

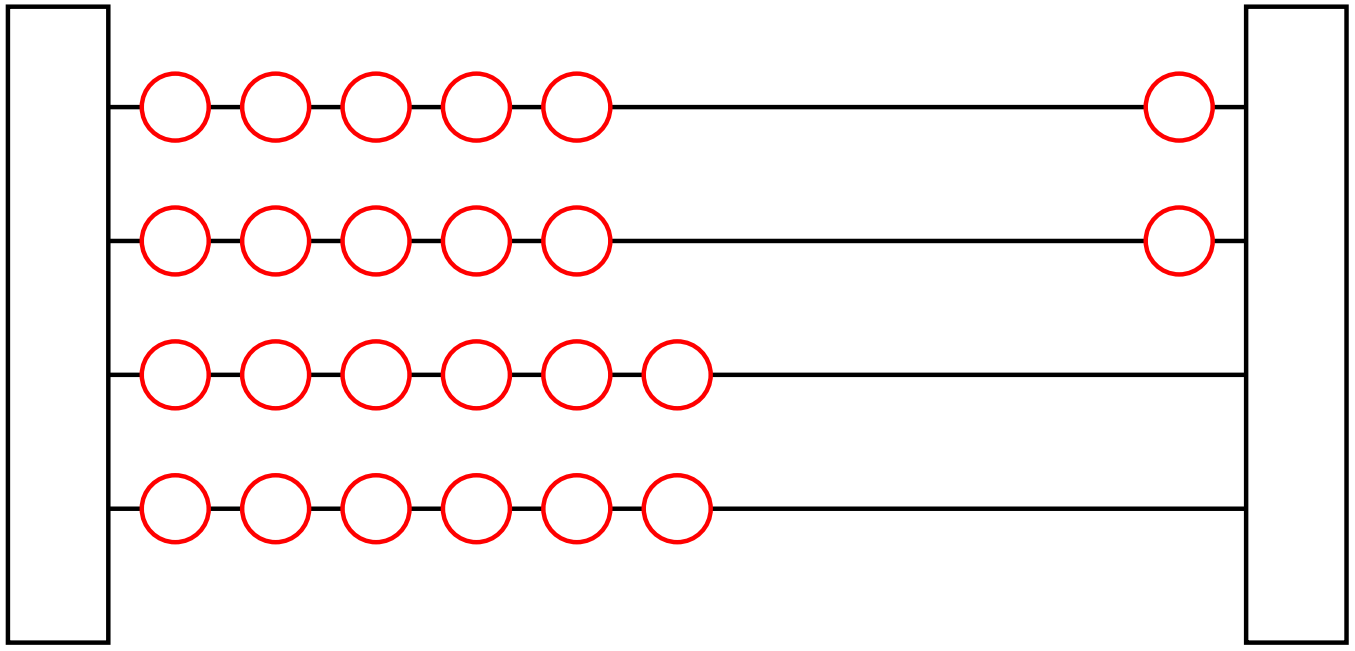
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



10 (6)

Le système hexal

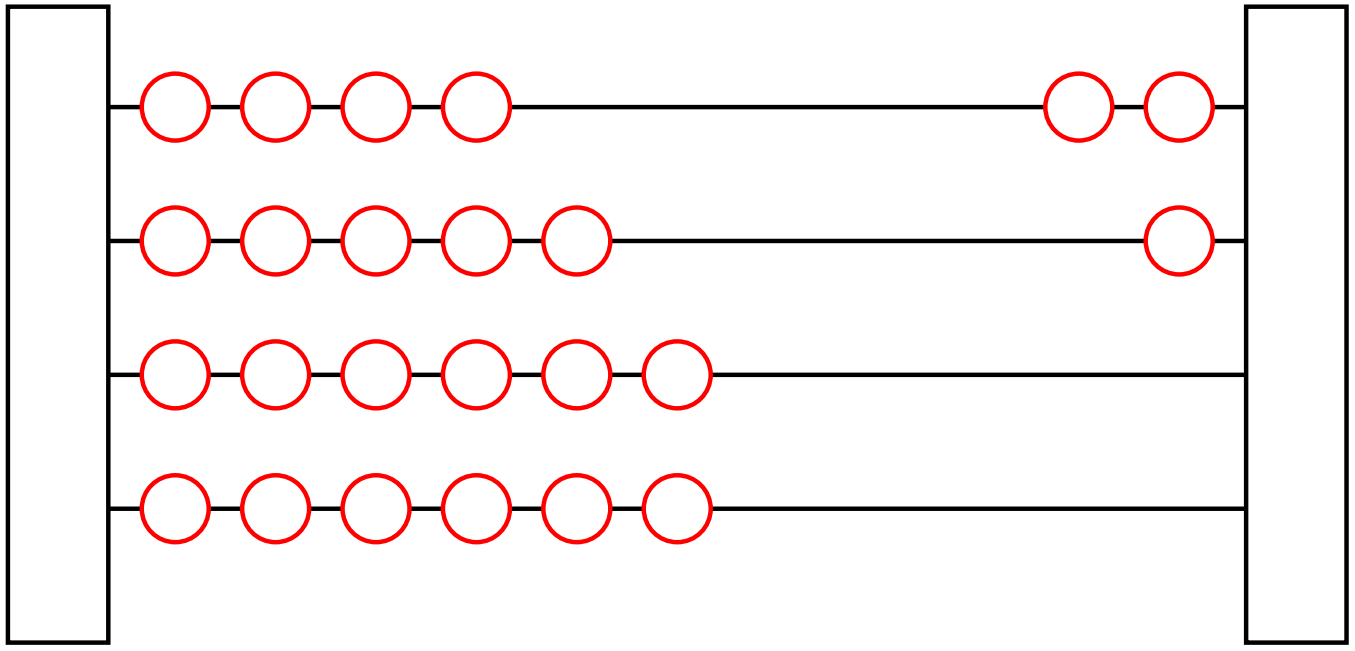
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



11 (7)

Le système hexal

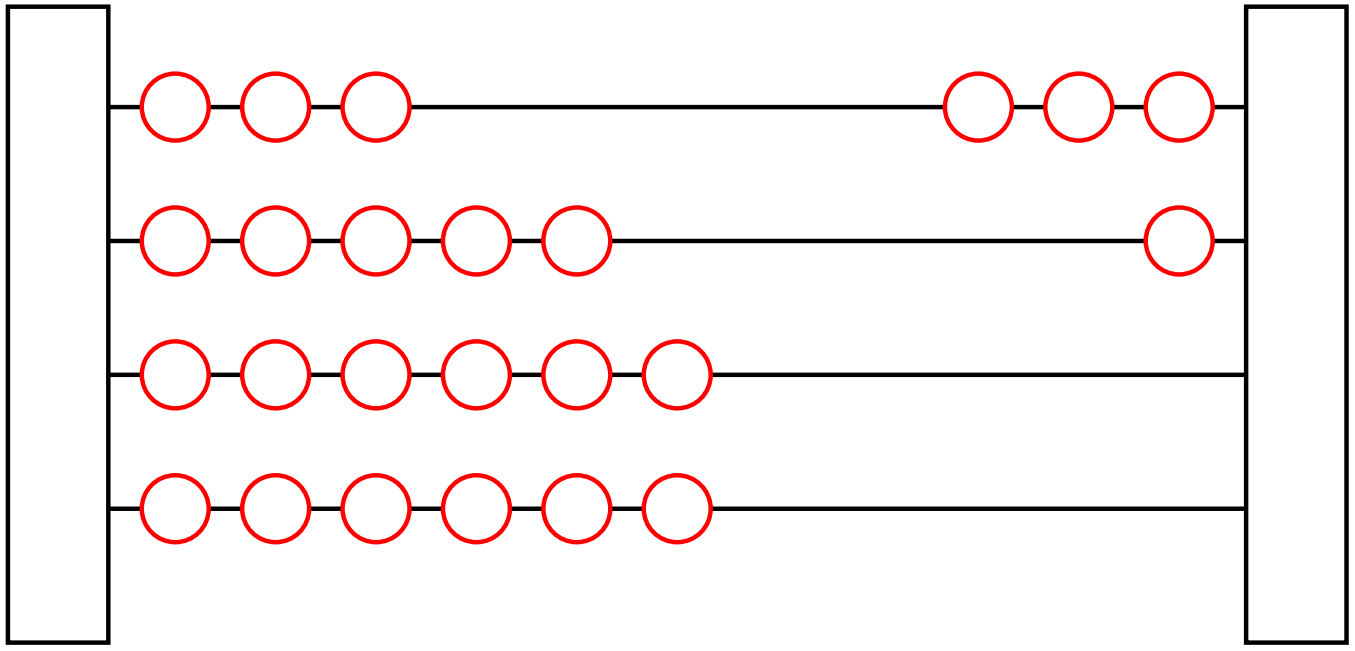
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



12 (8)

Le système hexal

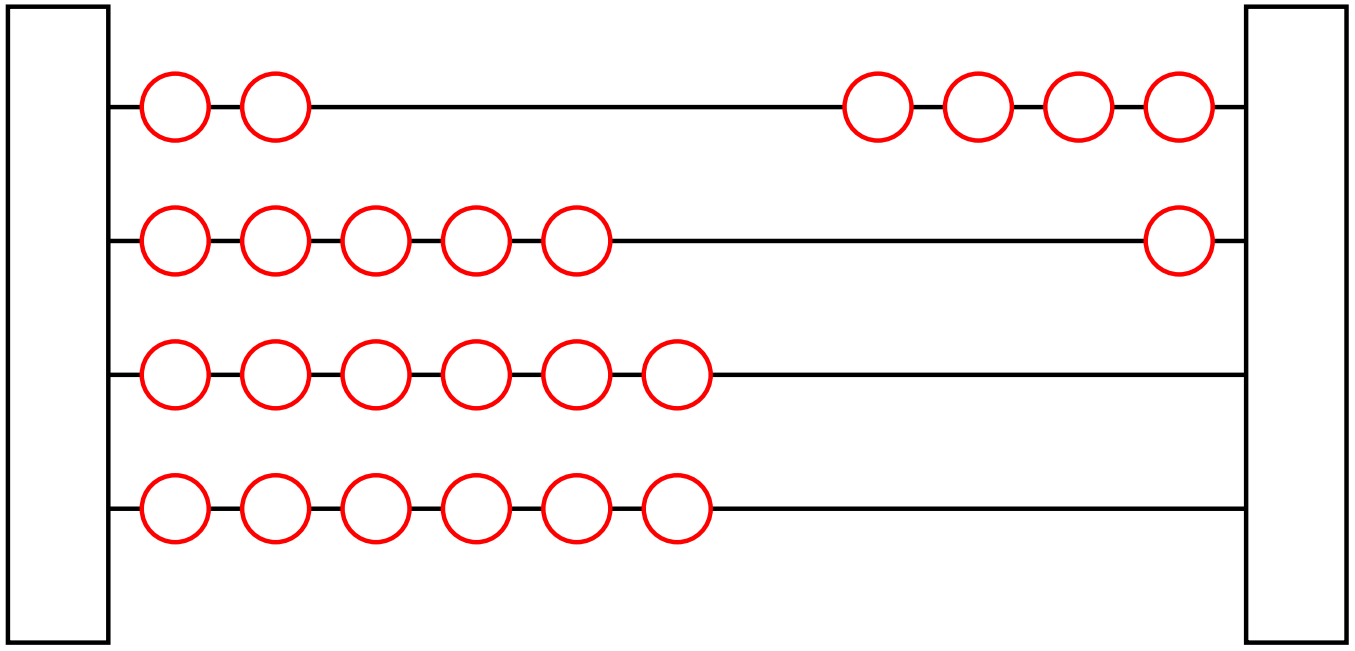
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



13 (9)

Le système hexal

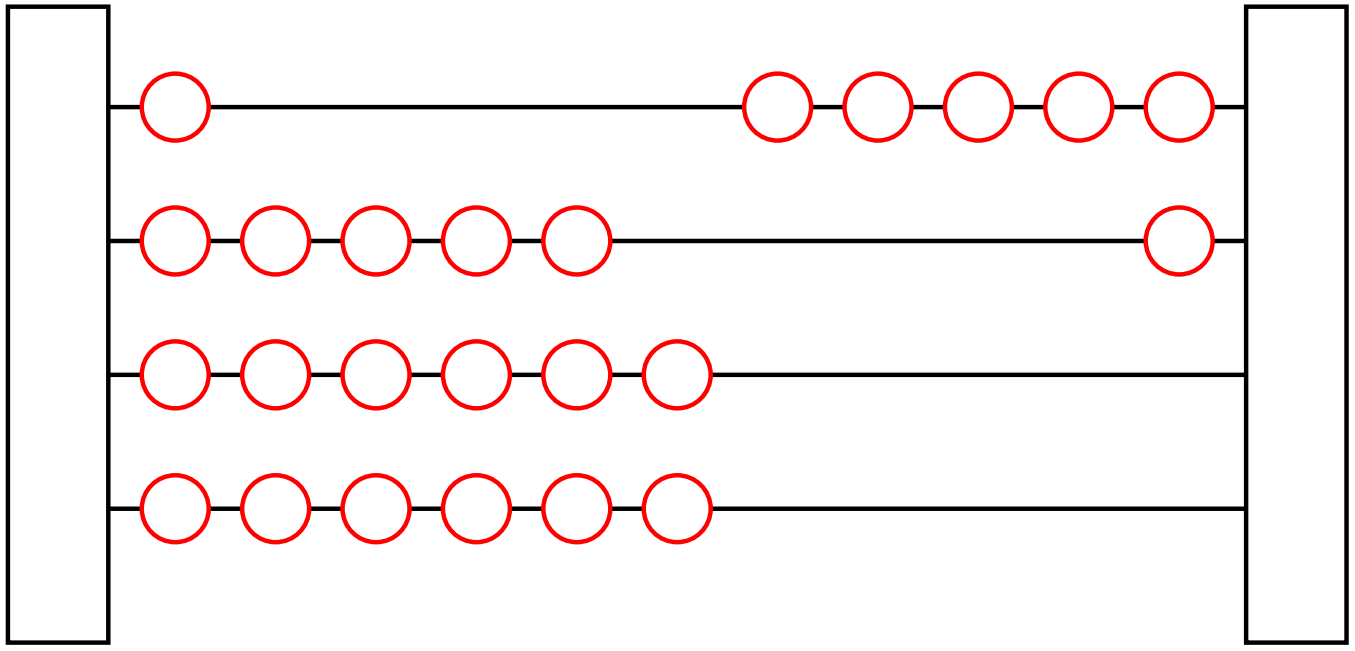
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



$14 \quad (10)$

Le système hexal

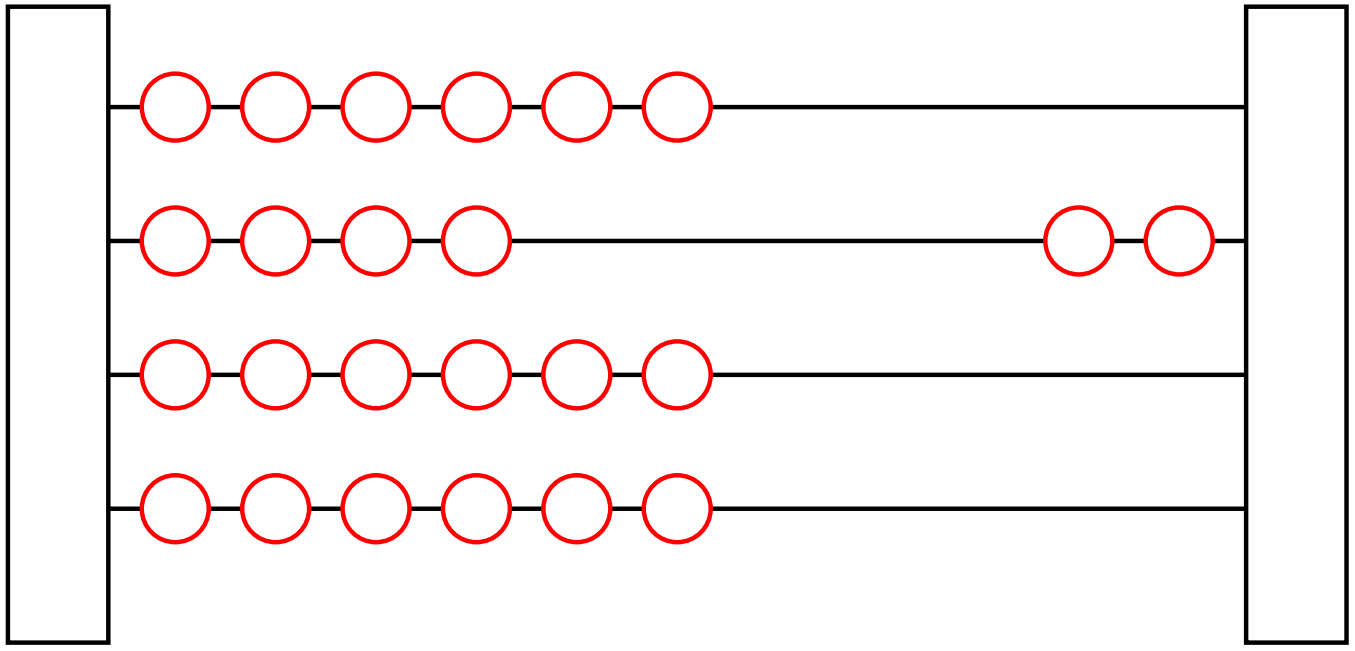
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



15 (11)

Le système hexal

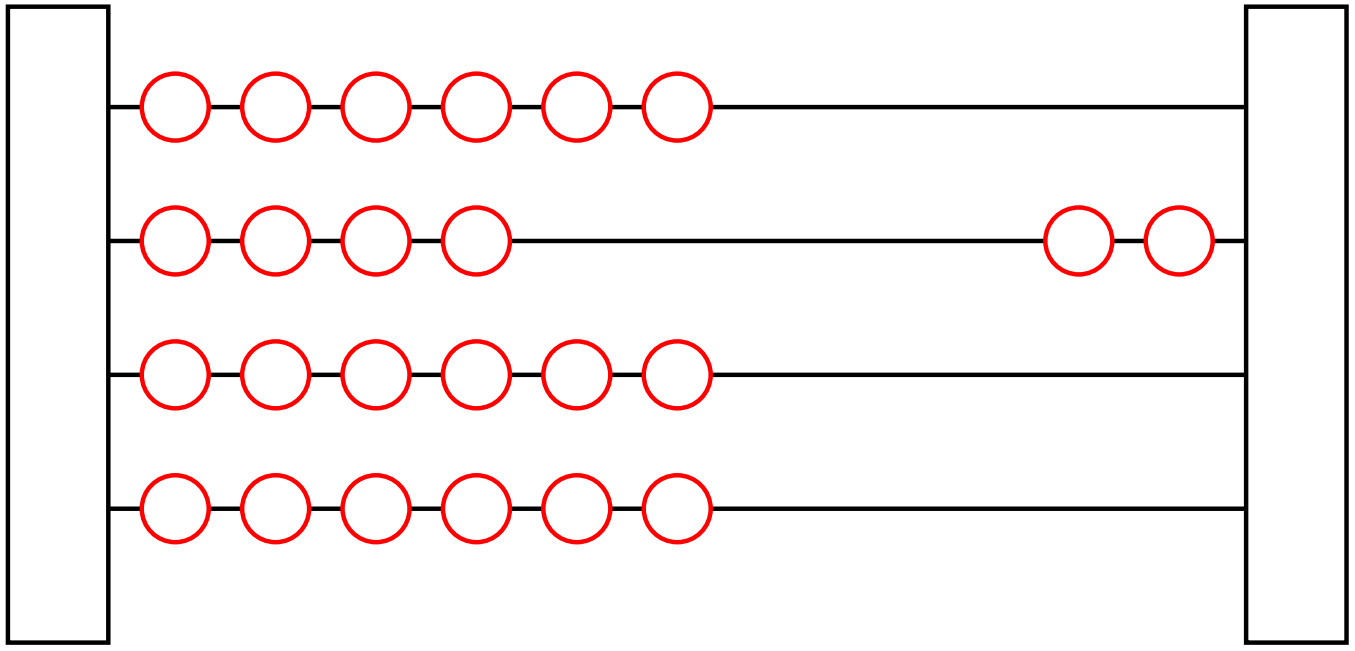
$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



20 (12)

Le système hexal

$\longrightarrow$ encoder des nombres avec les chiffres 0, 1, 2, 3, 4, 5:



$20 \quad (12)$

$\longrightarrow$ ca marche aussi bien que le système décimal

Le système hexadécimal (à la base 16)

$\longrightarrow$ encoder des nombres avec les chiffres

`0, 1, 2, 3, 4, 5, 6, 7, 8, 9`

. . . et avec les lettres

`A, B, C, D, E, F`

(10) (11) (12) (13) (14) (15)

Exemples:

`42` = 4 * 16 + 2 = 66

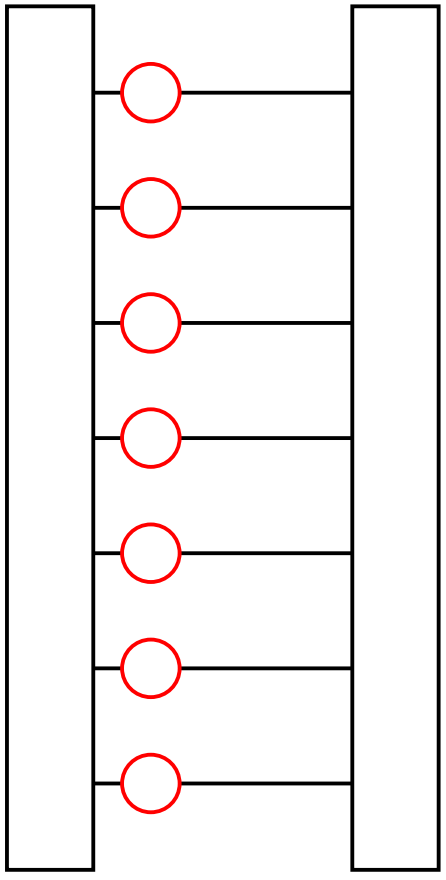
`A3` = 10 * 16 + 3 = 163

`5B` = 5 * 16 + 11 = 91

`FF` = 15 * 16 + 15 = 255

Le système binaire

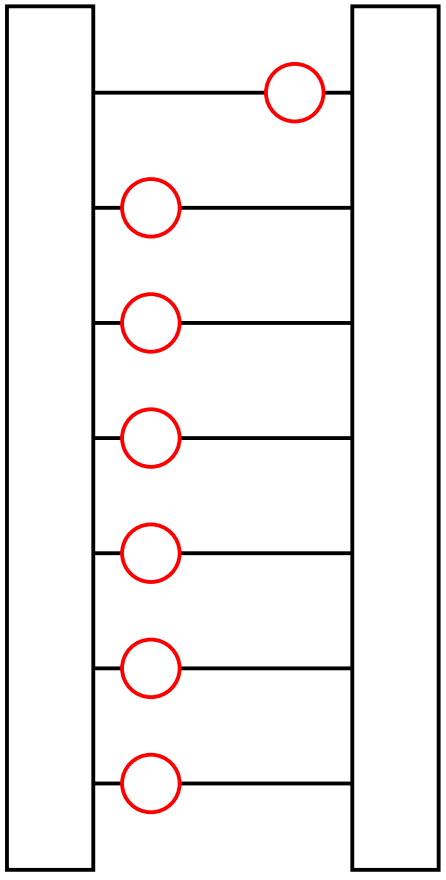
$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



0

Le système binaire

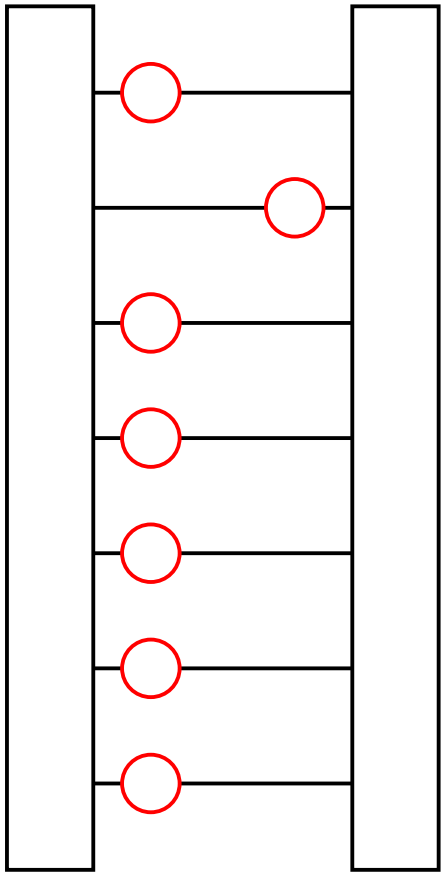
$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



1 (1)

Le système binaire

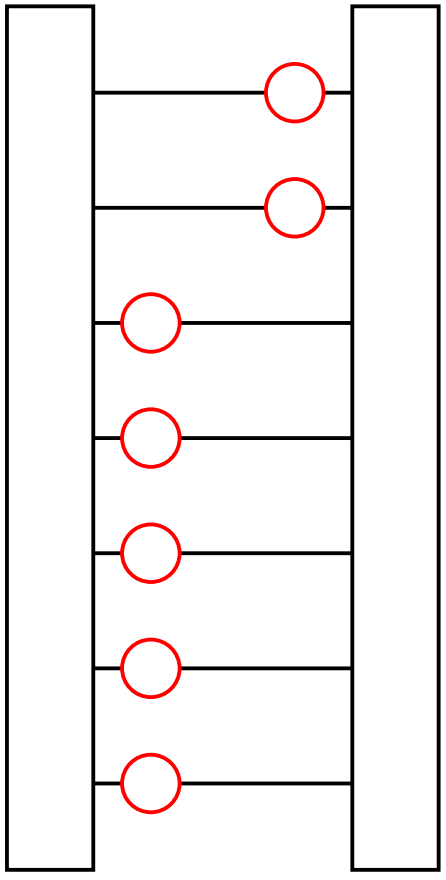
$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



`10` (2)

Le système binaire

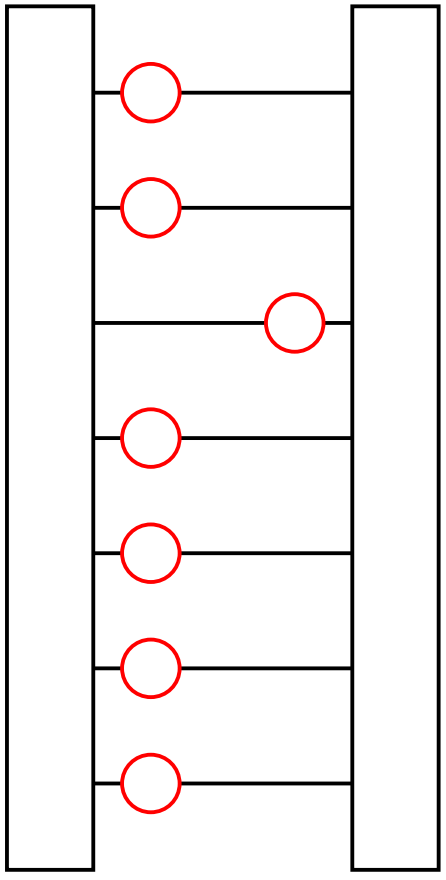
$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



`11` (3)

Le système binaire

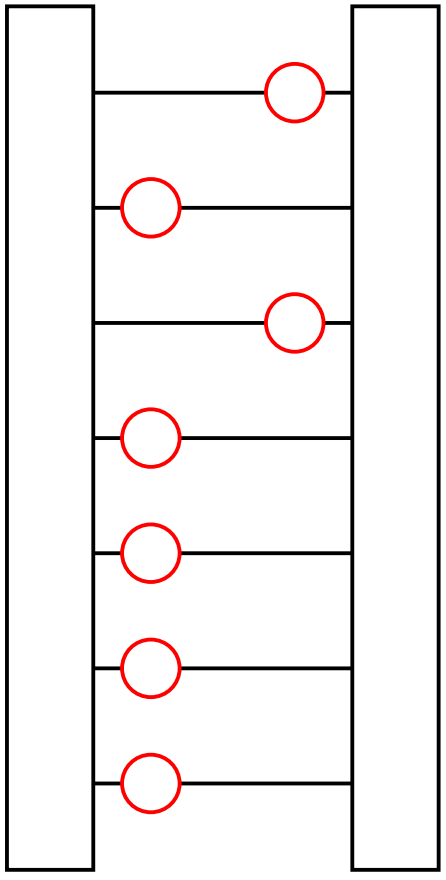
$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



`100` (4)

Le système binaire

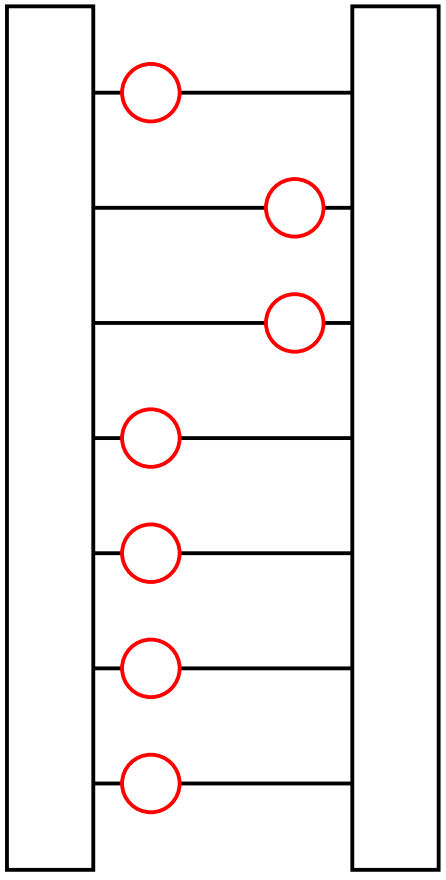
$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



`101` (5)

Le système binaire

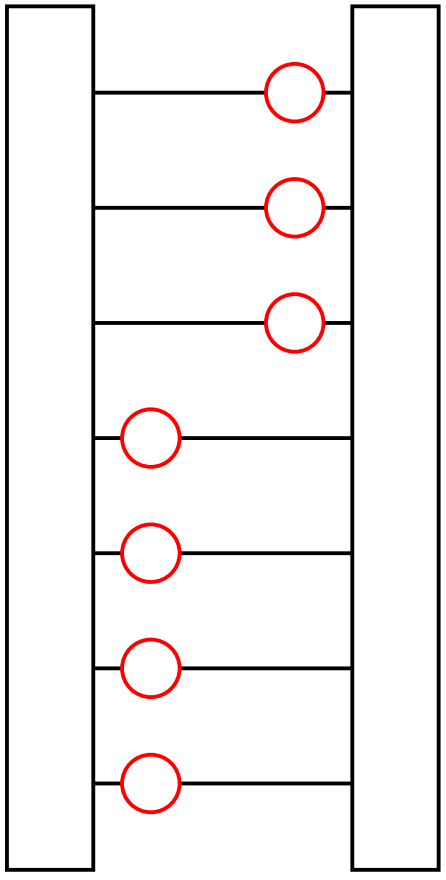
$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



`110` (6)

Le système binaire

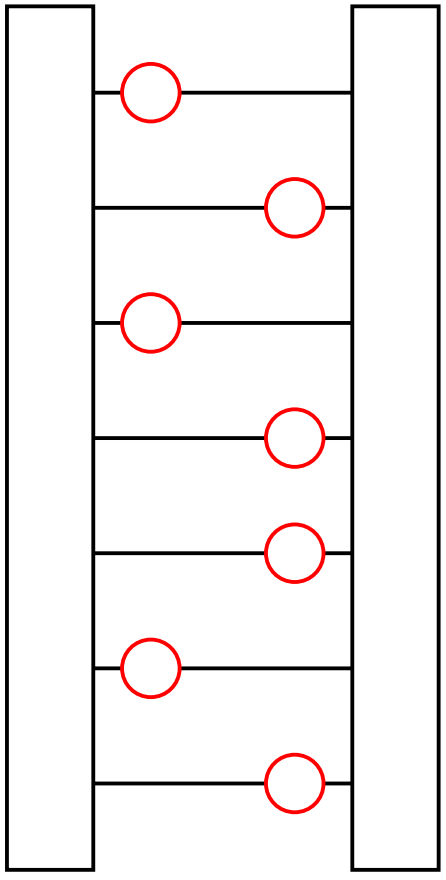
$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



`111` (7)

Le système binaire

$\longrightarrow$ encoder des nombres avec les chiffres 0 et 1



$$1011010 = 2^6 + 2^4 + 2^3 + 2^1 = 90$$

Opérations arithmétiques dans le système binaire

Addition

$$\begin{array}{rcccc} & 1 & 0 & 1 & 1 \\ + & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

$$\begin{array}{rc} & 11 \\ + & 6 \\ \hline & 17 \end{array}$$

Opérations arithmétiques dans le système binaire

Addition

```
    1  0  1  1
+   0  1  1  0
-------------
          1
```

```
   11
+   6
-----
   17
```

Opérations arithmétiques dans le système binaire

Addition

$$\begin{array}{rcccc} & 1 & 0 & 1 & 1 \\ + & 0 & 1_1 & 1 & 0 \\ \hline & & & \cancel{2} & 1 \\ & & & 0 & \end{array}$$

$$\begin{array}{rc} & 11 \\ + & 6 \\ \hline & 17 \end{array}$$

Opérations arithmétiques dans le système binaire

Addition

$$\begin{array}{rcccc} & 1 & 0 & 1 & 1 \\ + & 0_1 & 1 & 1 & 0 \\ \hline & & \cancel{2} & 0 & 1 \\ & & 0 & & \end{array}$$

$$\begin{array}{rc} & 11 \\ + & 6 \\ \hline & 17 \end{array}$$

Opérations arithmétiques dans le système binaire

Addition

	1	0	1	1
+	0	1	1	0
1	0	0	0	1

	11
+	6
	17

Opérations arithmétiques dans le système binaire

Soustraction

$$
\begin{array}{rcccc}
 & 1 & 0 & 1 & 1 \\
- & 0 & 1 & 1 & 0 \\
\hline
 & & & & 1
\end{array}
$$

Opérations arithmétiques dans le système binaire

Soustraction

$$\begin{array}{rcccc} & 1 & 0 & 1 & 1 \\ - & 0 & 1 & 1 & 0 \\ \hline & & & 0 & 1 \end{array}$$

Opérations arithmétiques dans le système binaire

Soustraction

$$
\begin{array}{rcccc}
 & 1 & 0 & 1 & 1 \\
- & 0_1 & 1 & 1 & 0 \\
\hline
 & & 1 & 0 & 1
\end{array}
$$

Opérations arithmétiques dans le système binaire

Soustraction

$$\begin{array}{rcccc} & 1 & 0 & 1 & 1 \\ - & 0 & 1 & 1 & 0 \\ \hline & & 1 & 0 & 1 \end{array}$$

$$\begin{array}{rr} & 11 \\ - & 6 \\ \hline & 5 \end{array}$$

Opérations arithmétiques dans le système binaire

Multiplication

1 0 1 1 × 0 1 1 0 =

Opérations arithmétiques dans le système binaire

Multiplication

$$1011 \times 0110 = 0110$$

$$1011$$

$$1011$$

$$0000$$

$$01100110$$

Opérations arithmétiques dans le système binaire

Multiplication

1	0	1	1	×	0	1	1	0	=				0	0	0	0
											1	0	1	1		
										1	0	1	1			
									0	0	0	0				
																0

Opérations arithmétiques dans le système binaire

Multiplication

1	0	1	1	×	0	1	1	0	=				0	0	0	0
											1	0	1	1		
										1	0	1	1			
									0	0	0	0				
															1	0

Opérations arithmétiques dans le système binaire

Multiplication

$$1011 \times 0110 = 0110$$

$$1011$$

$$1011$$

$$0000_{1}$$

$$0010$$

Opérations arithmétiques dans le système binaire

Multiplication

$$
\begin{array}{cccccccccccccccc}
1 & 0 & 1 & 1 & \times & 0 & 1 & 1 & 0 & = & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & 1 & 0 & 1 & 1 & \\
 & & & & & & & & & & & 1 & 0 & 1 & 1 & & \\
 & & & & & & & & & & 0 & 0 & 0_1 & 0_1 & & & \\
\hline
 & & & & & & & & & & & & & 0 & 0 & 1 & 0
\end{array}
$$

Opérations arithmétiques dans le système binaire

Multiplication

$$
\begin{array}{cccccccccccccccc}
1 & 0 & 1 & 1 & \times & 0 & 1 & 1 & 0 & = & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & 1 & 0 & 1 & 1 & \\
 & & & & & & & & & & & 1 & 0 & 1 & 1 & & \\
 & & & & & & & & & & 0 & 0_1 & 0_1 & 0_1 & & & \\
\hline
 & & & & & & & & & & & & 0 & 0 & 0 & 1 & 0
\end{array}
$$

Opérations arithmétiques dans le système binaire

Multiplication

$$
\begin{array}{cccccccccccccc}
1 & 0 & 1 & 1 & \times & 0 & 1 & 1 & 0 & = & & & & \\
 & & & & & & & & & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & 1 & 0 & 1 & 1 & \\
 & & & & & & & & & & 1 & 0 & 1 & 1 & & \\
 & & & & & & & & & 0_1 & 0_1 & 0_1 & 0_1 & & & \\
\hline
 & & & & & & & & & & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

Opérations arithmétiques dans le système binaire

Multiplication

$$
\begin{array}{cccccccccccccc}
1 & 0 & 1 & 1 & \times & 0 & 1 & 1 & 0 & = & & & & \\
 & & & & & & & & & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & 1 & 0 & 1 & 1 & \\
 & & & & & & & & & & 1 & 0 & 1 & 1 & & \\
 & & & & & & & & & 0 & 0 & 0 & 0 & & & \\
\hline
 & & & & & & & & & 1 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

$$= 66 = 11 \times 6$$

Pourquoi le système binaire ?

. . . et non pas un système ternaire, hexal, décimal . . . ?

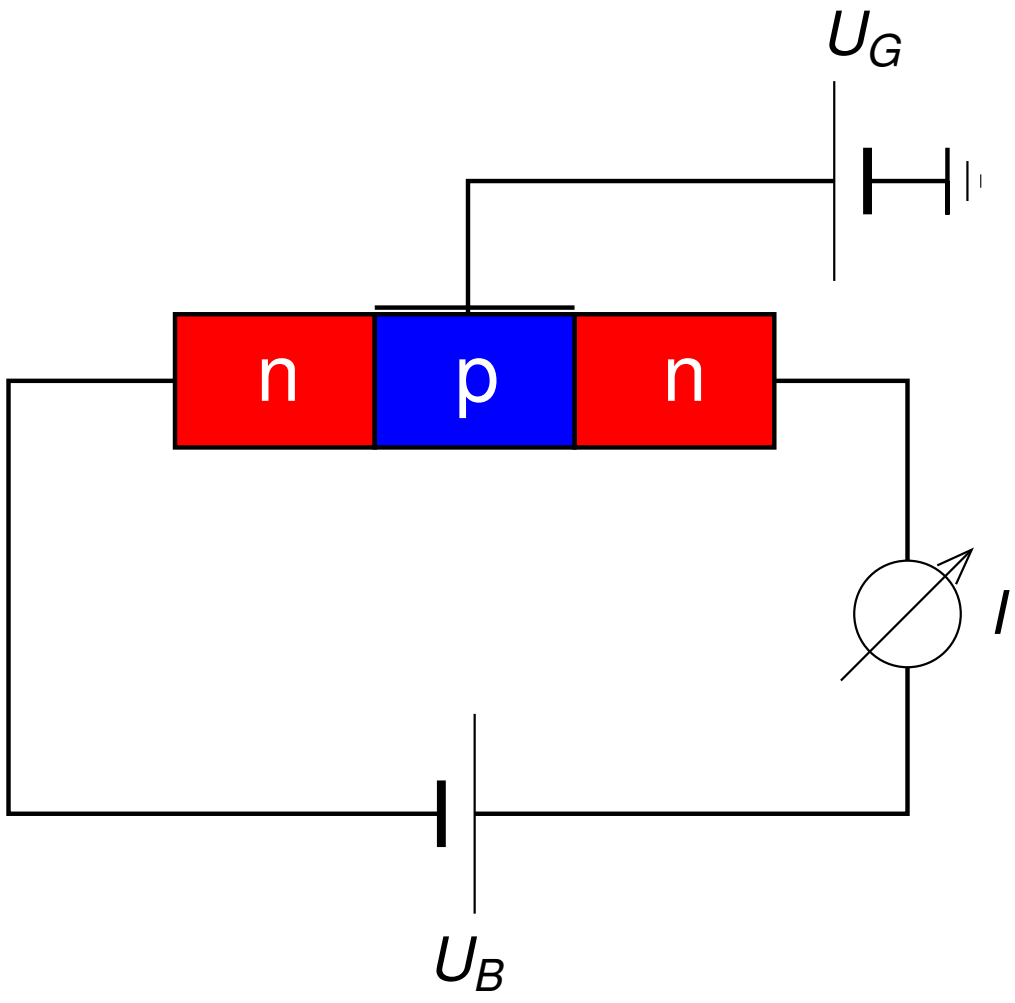
$\rightarrow$ plus facile à réaliser

$\rightarrow$ facile à encoder des informations plus générales

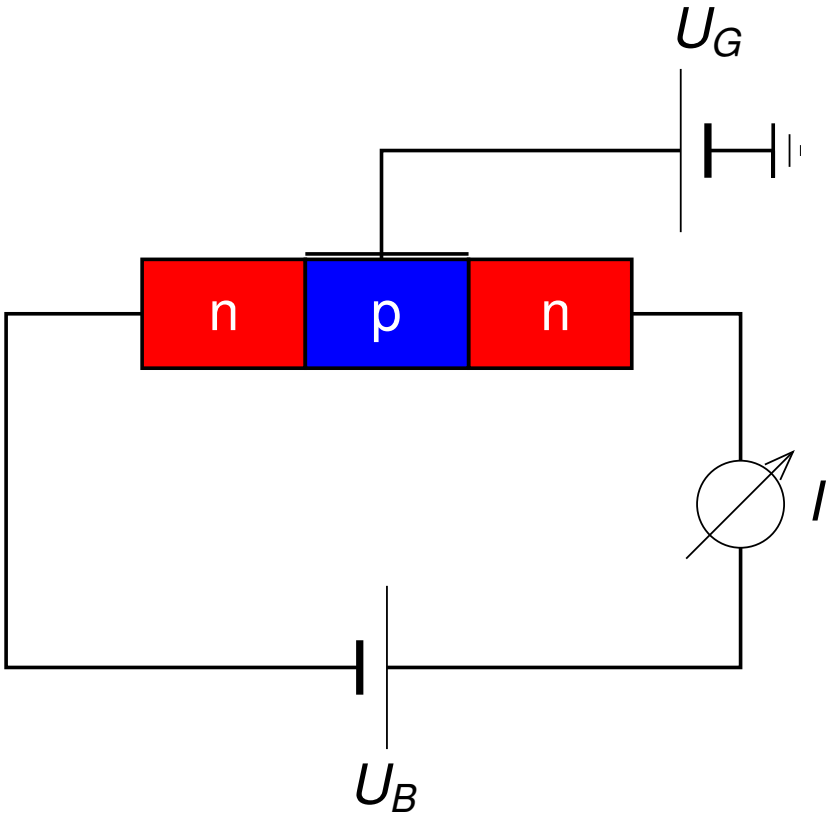
$\rightarrow$ plus robuste que des systèmes plus compliqués

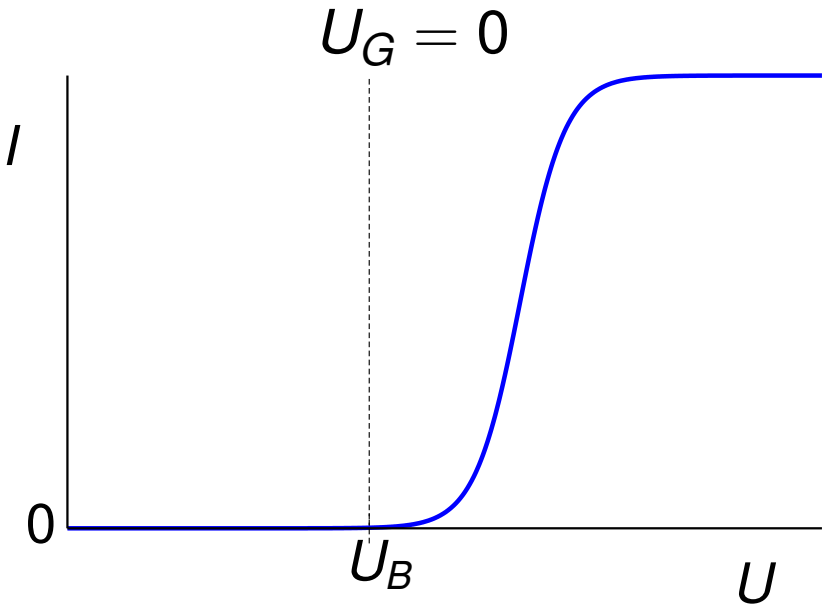
$\rightarrow$ longue histoire :
télégraphie (code de Morse), cartes perforés, CDs . . .

2.2 La réalisation du bit (binary digit)



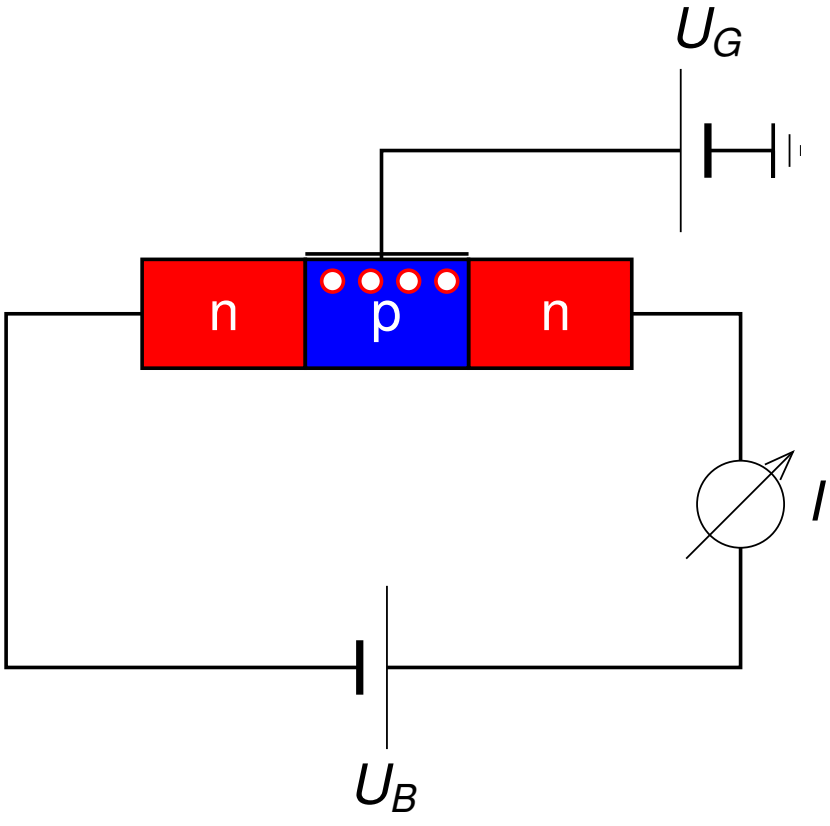
2.2 La réalisation du bit (binary digit)

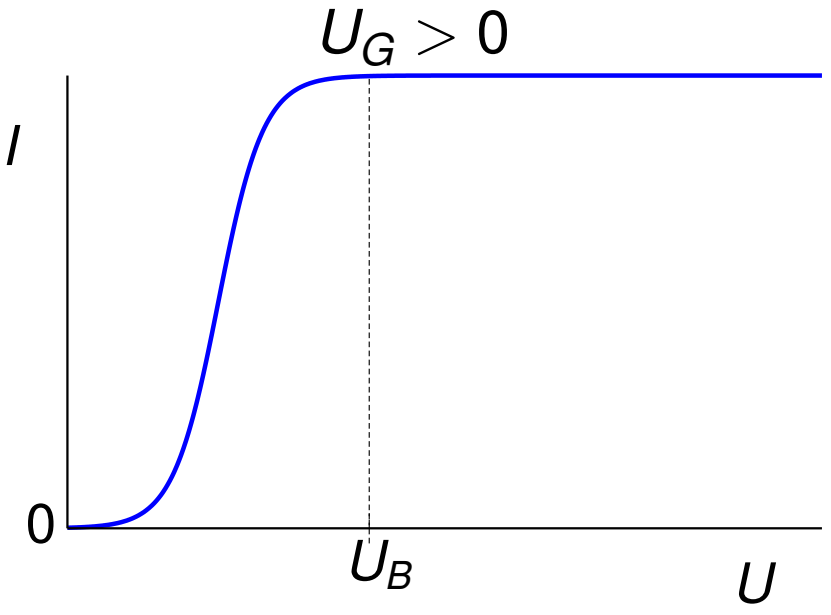




$\rightarrow$ pas de courant : bit 0

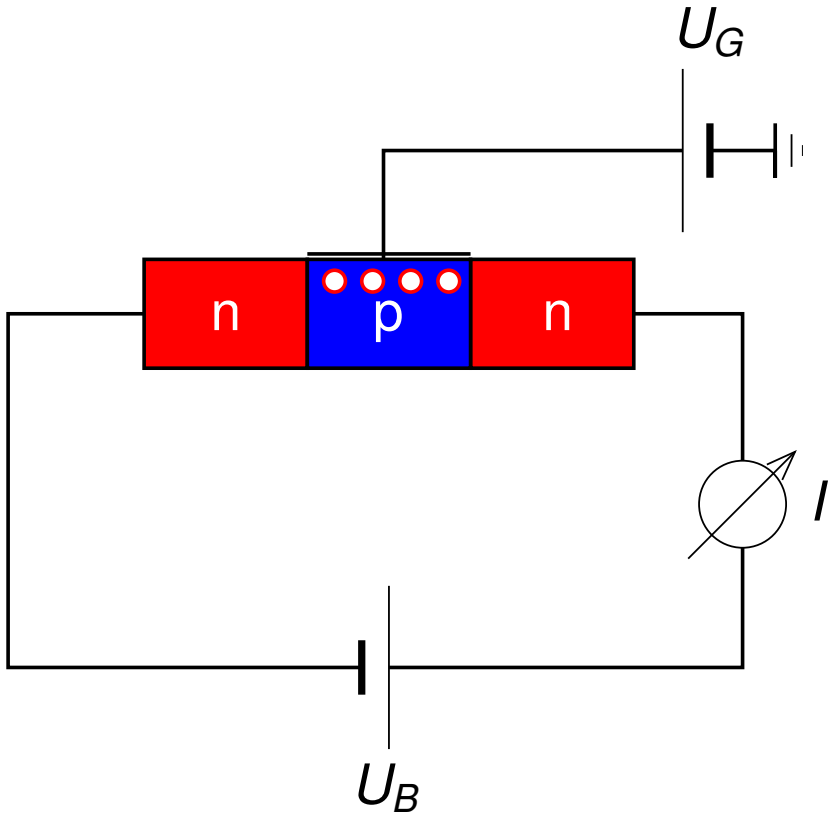
2.2 La réalisation du bit (binary digit)

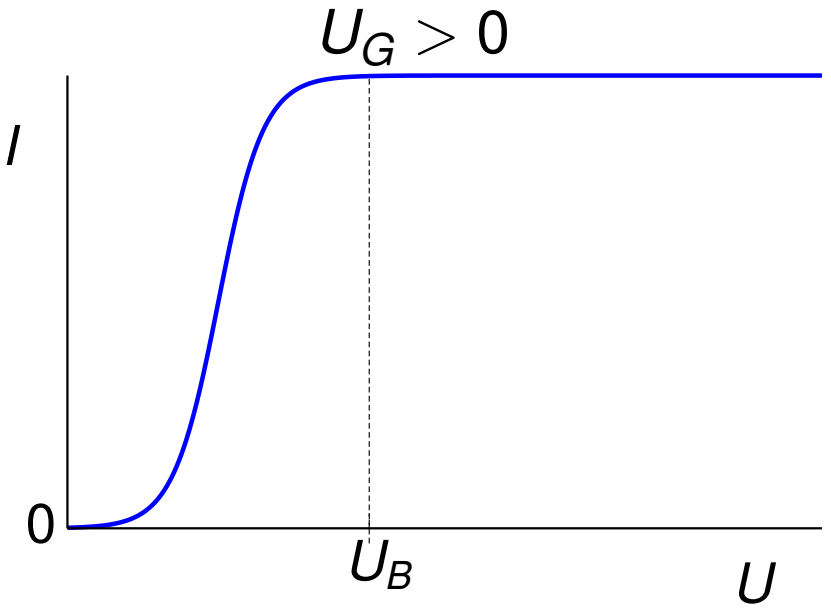




$\rightarrow$ courant : bit 1

2.2 La réalisation du bit (binary digit)





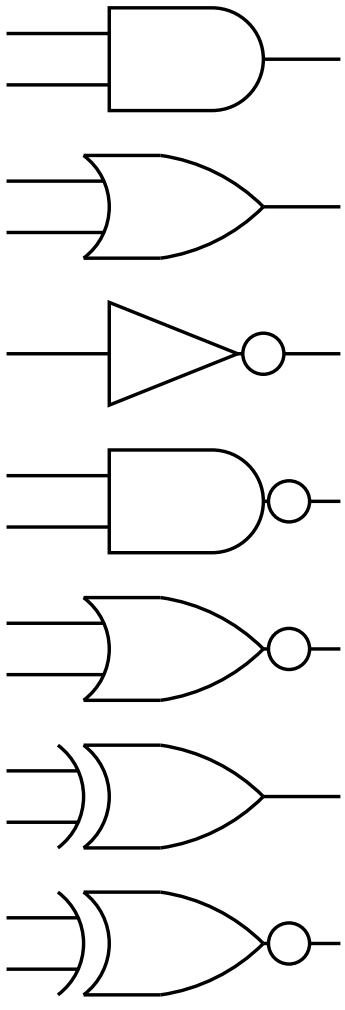
$\rightarrow$ courant : bit 1

$\longrightarrow$ comment garder de l'information binaire sur un circuit ?

Les mémoires binaires sur les circuits intégrés

Rappel: les opérations logiques élémentaires

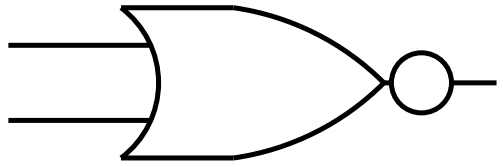
- AND : $A = B \wedge C$
- OR : $A = B \vee C$
- NOT : $A = \overline{B}$
- NAND : $A = \overline{B \wedge C}$
- NOR : $A = \overline{B \vee C}$
- XOR : $A = (B \wedge \overline{C}) \vee (C \wedge \overline{B})$
- XNOR : $A = (B \wedge C) \vee (\overline{B} \wedge \overline{C})$

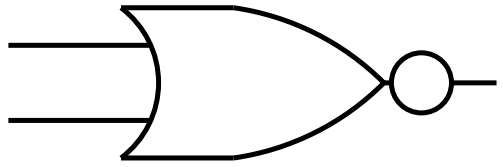


Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

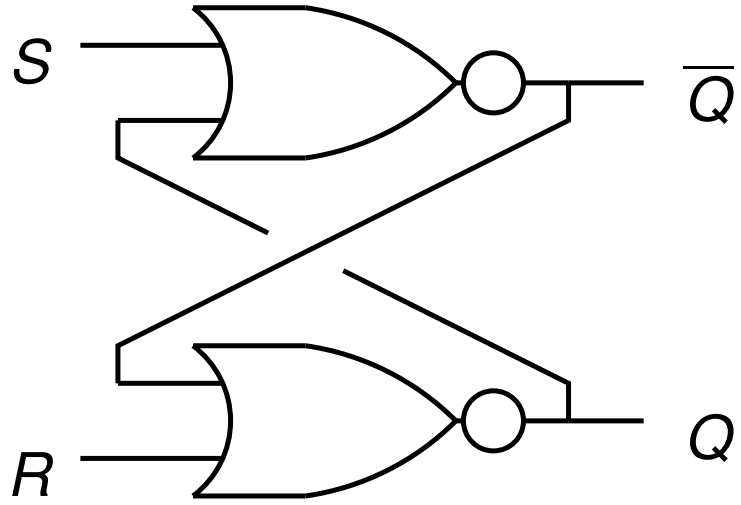




Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR



input S: activation (set)

input R: désactivation (reset)

output Q: lire le contenu

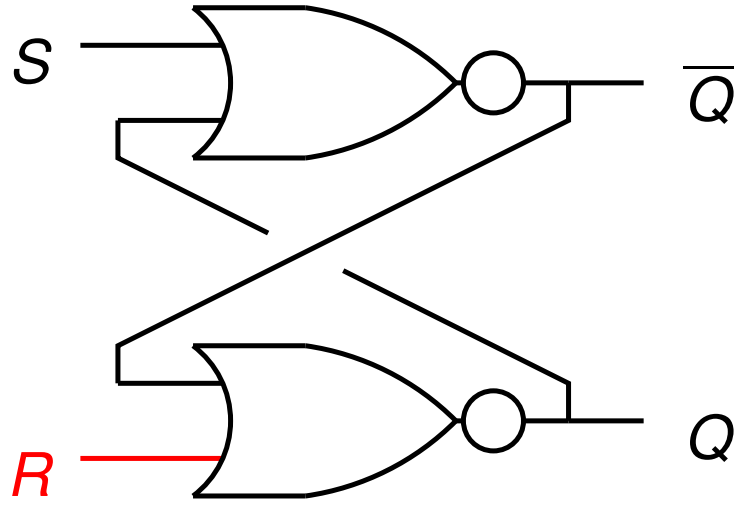
output $\overline{Q}$: le contraire de Q

Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

désactivation avec le contact R



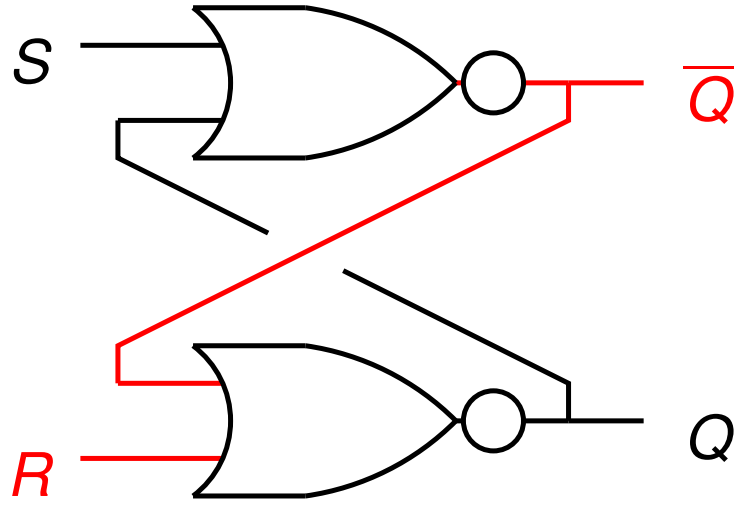
Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

désactivation avec le contact R

$\rightarrow$ Q “est faux”
$\rightarrow$ $\overline{Q}$ “est vrai”



Les mémoires binaires sur des circuits intégrés

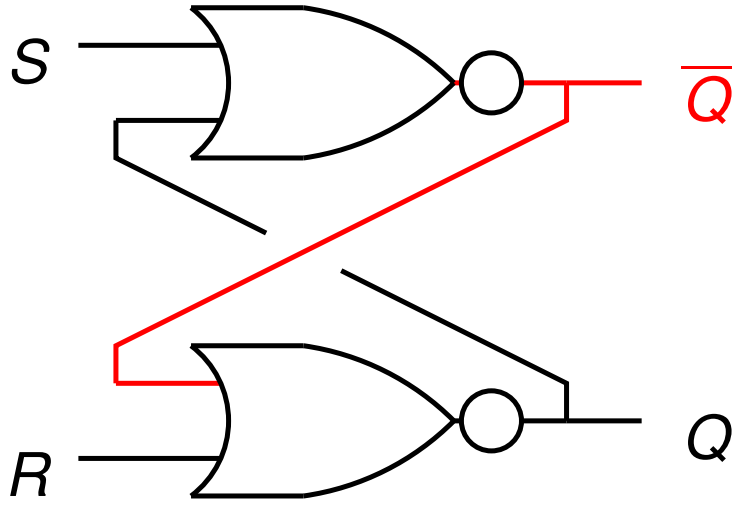
$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

désactivation avec le contact R

$\rightarrow$ Q “est faux”
$\rightarrow$ $\overline{Q}$ “est vrai”

$\longrightarrow$ ça reste comme ça
même si R est découpé

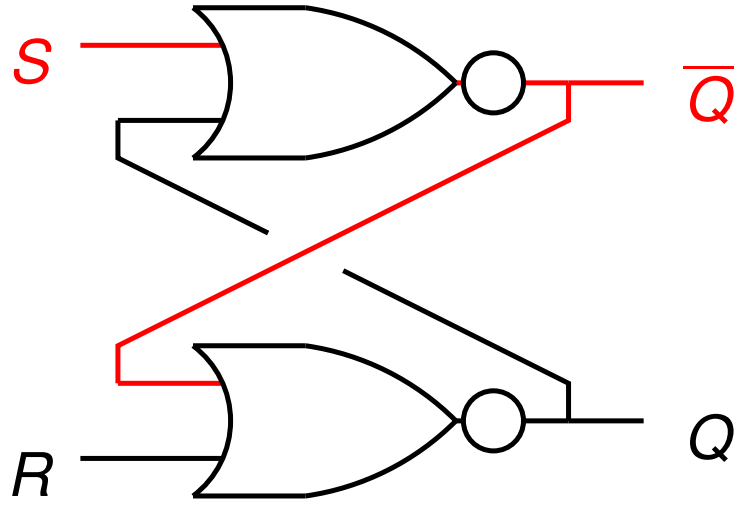


Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

activation avec le contact S



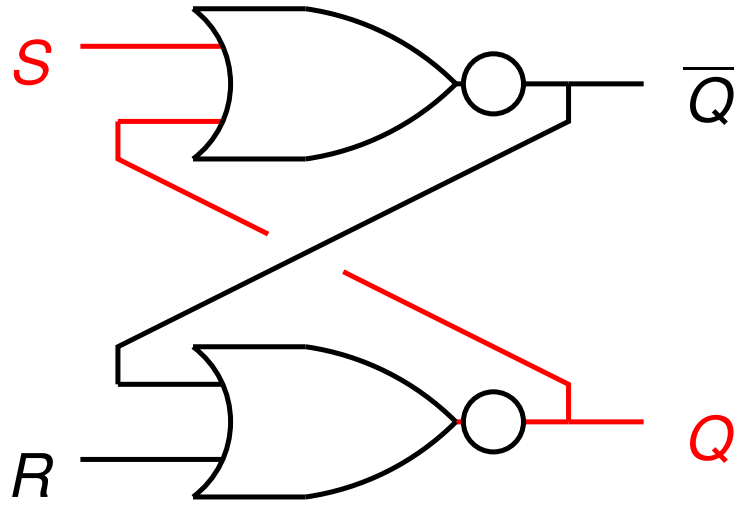
Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

activation avec le contact S

$\rightarrow$ Q “est vrai”
$\rightarrow$ $\overline{Q}$ “est faux”



Les mémoires binaires sur des circuits intégrés

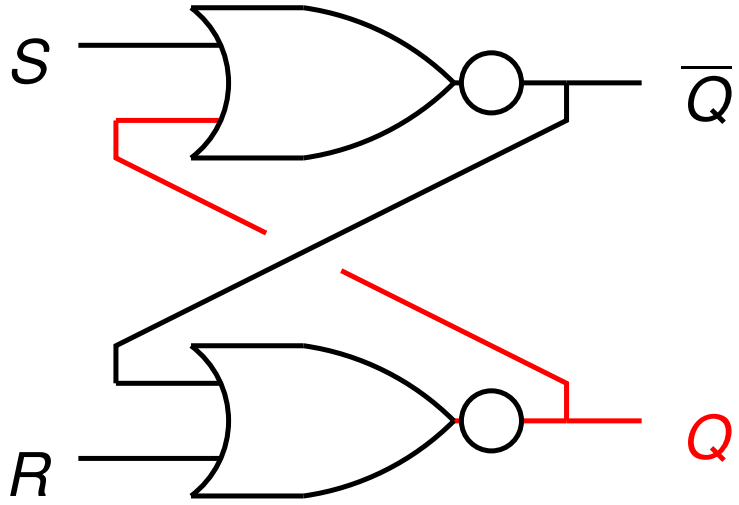
$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

activation avec le contact S

$\rightarrow Q$ “est vrai”
$\rightarrow \overline{Q}$ “est faux”

$\longrightarrow$ ça reste comme ça
même si S est découpé

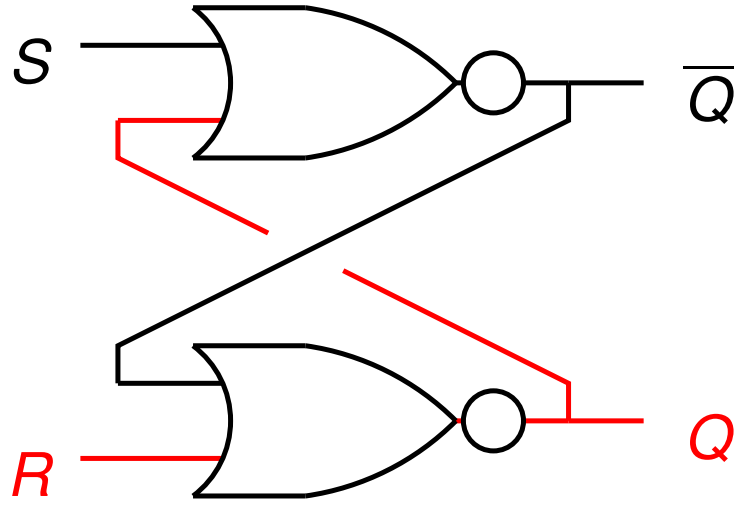


Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

désactivation avec le contact R



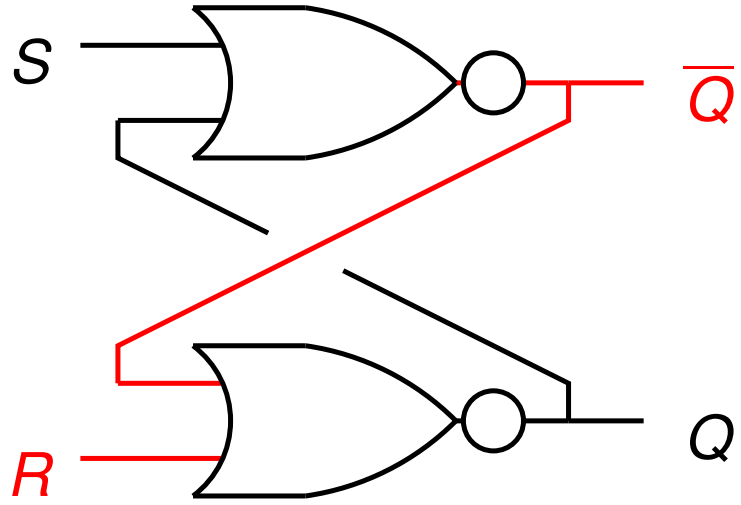
Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

désactivation avec le contact R

$\rightarrow$ Q “est faux”
$\rightarrow$ $\overline{Q}$ “est vrai”



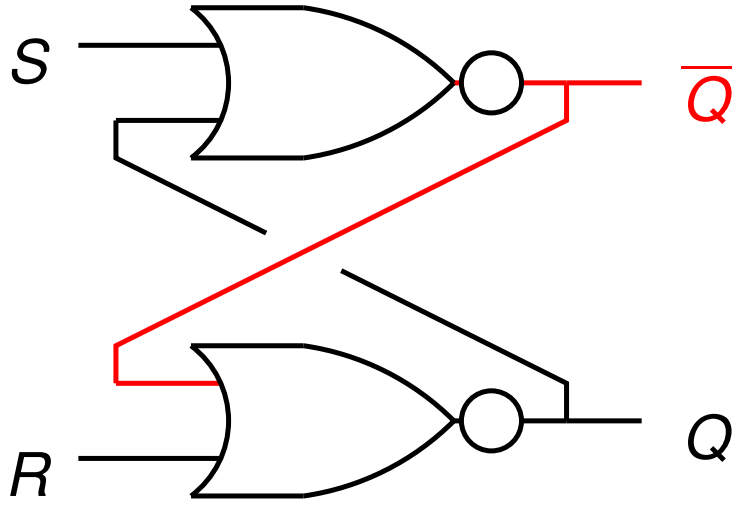
Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

désactivation avec le contact R

$\rightarrow$ Q “est faux”
$\rightarrow$ $\overline{Q}$ “est vrai”

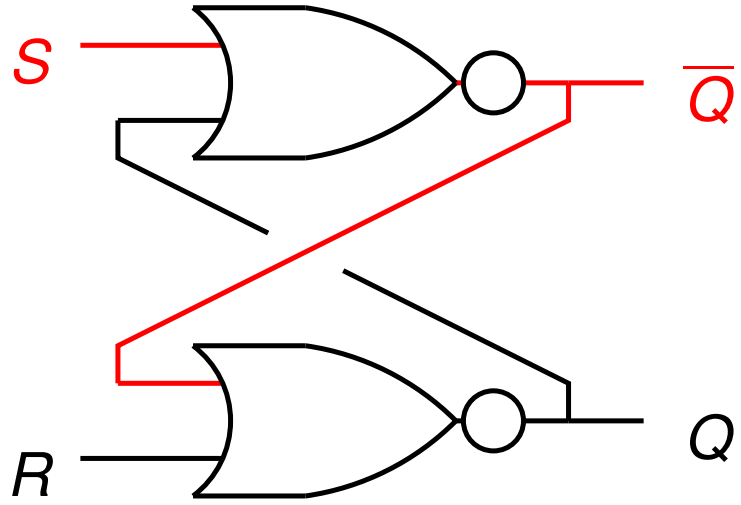


Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

activation avec le contact S



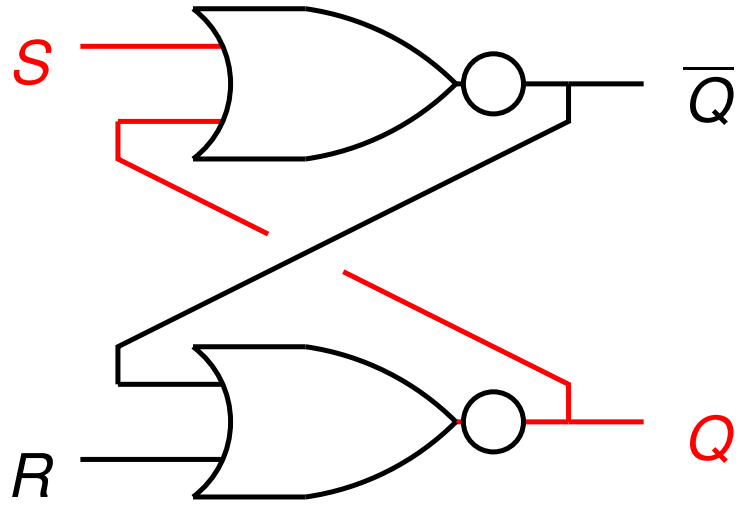
Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR

activation avec le contact S

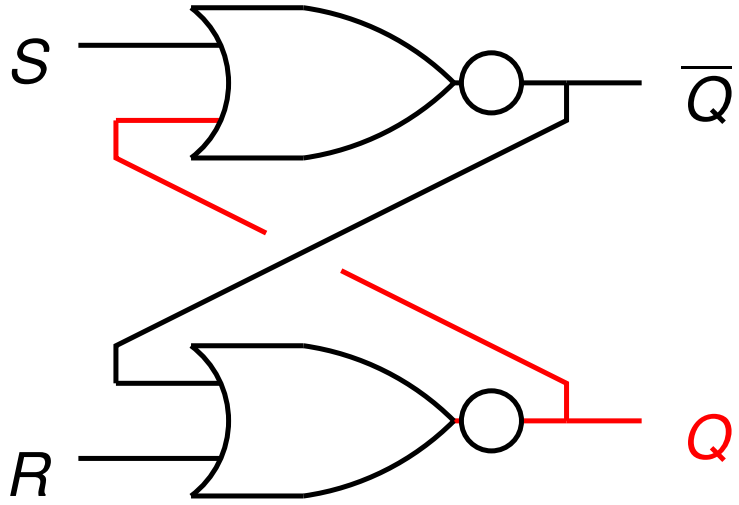
$\rightarrow$ Q “est vrai”
$\rightarrow$ $\overline{Q}$ “est faux”



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les circuits bistables (flip-flop)

On utilise deux élements NOR



activation avec le contact S

$\rightarrow$ Q “est vrai”
$\rightarrow$ $\overline{Q}$ “est faux”

$\longrightarrow$ rapidité et stabilité
(tant que le circuit est connecté à la source de tension)

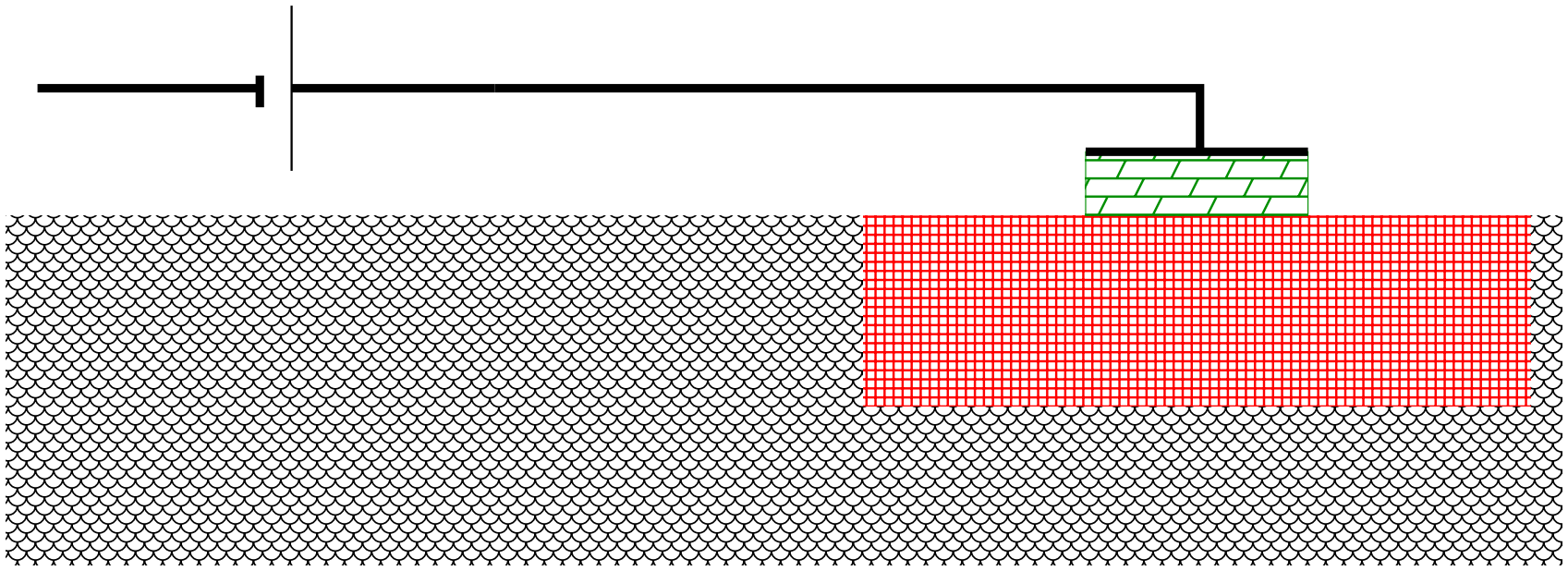
$\longrightarrow$ six transistors nécessaires pour maintenir un bit

$\longrightarrow$ SRAM (Static Random Access Memory)

Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires DRAM (Dynamic Random Access Memory)

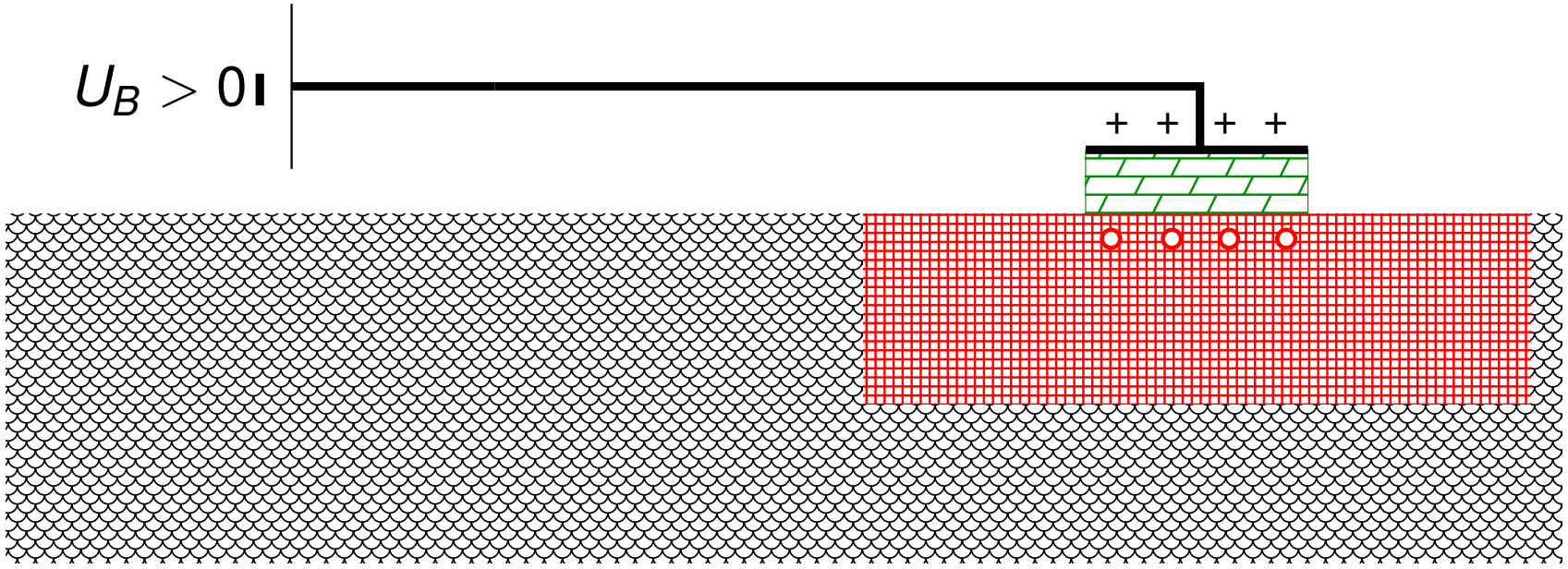
On utilise des capacitances pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires DRAM (Dynamic Random Access Memory)

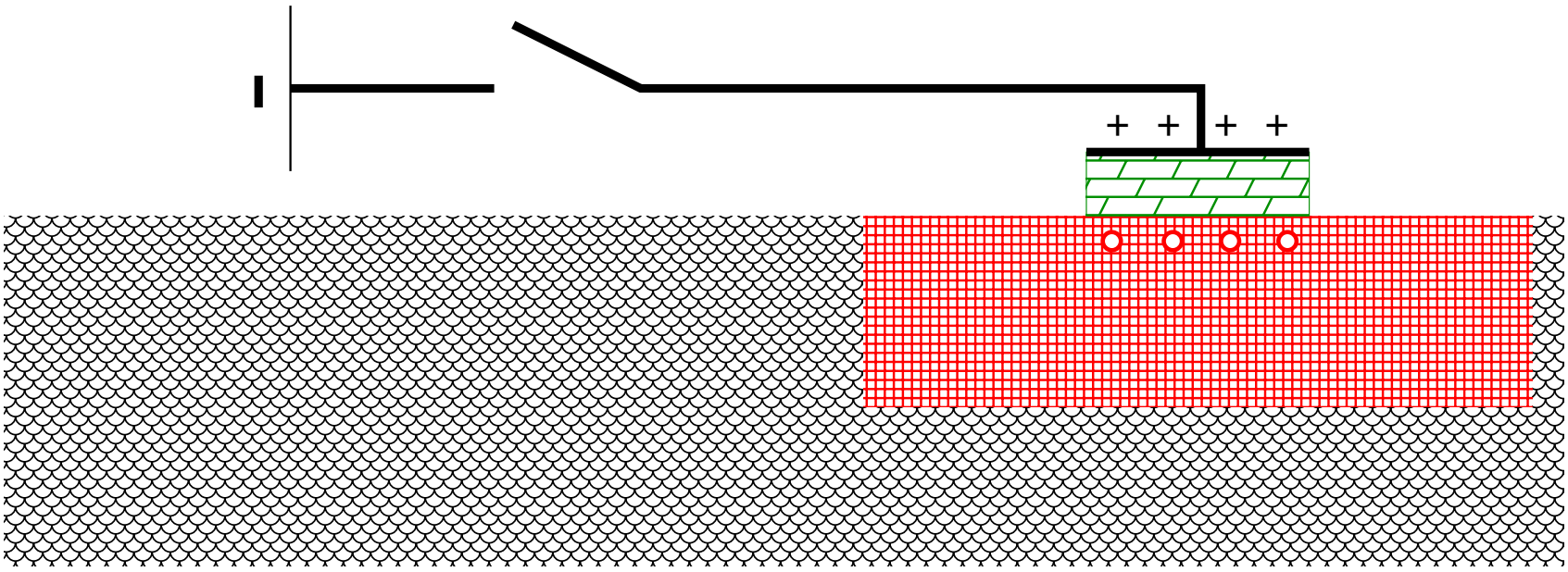
On utilise des capacitances pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires DRAM (Dynamic Random Access Memory)

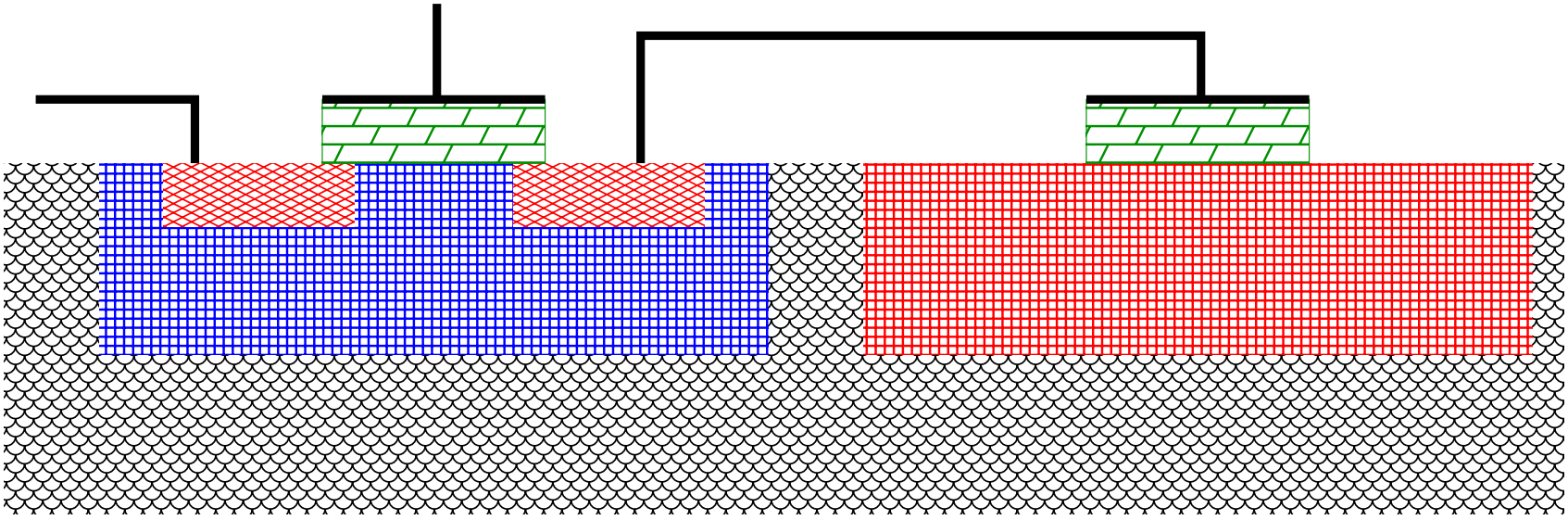
On utilise des capacitances pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires DRAM (Dynamic Random Access Memory)

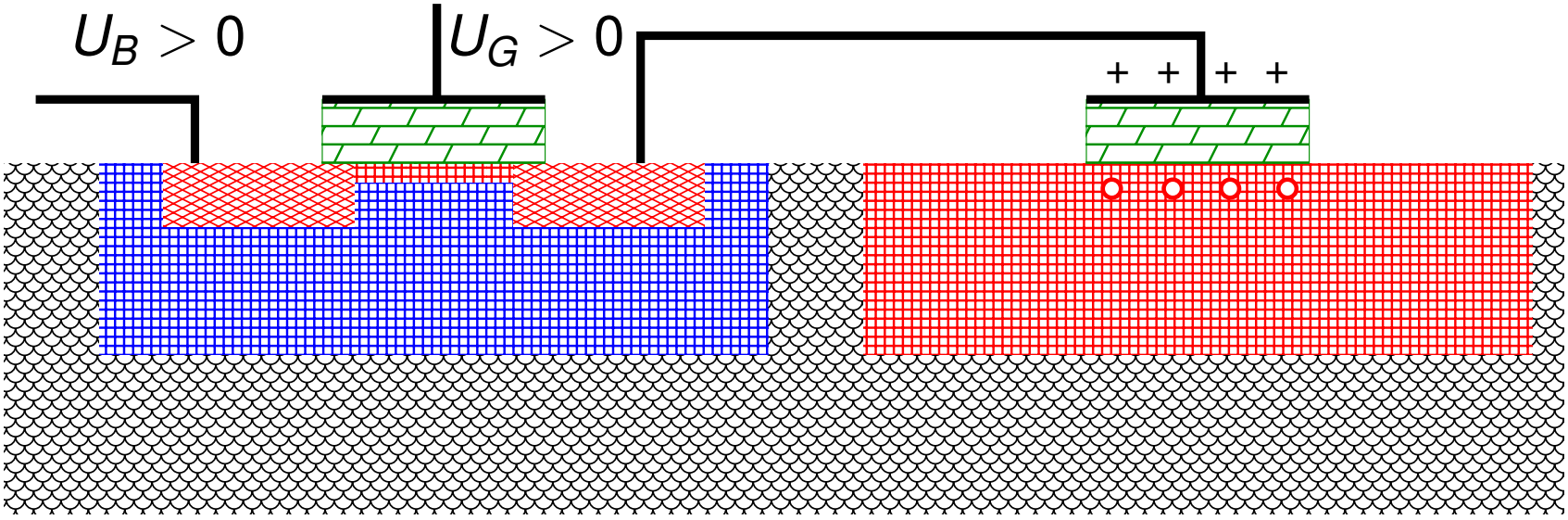
On utilise des capacitances pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires DRAM (Dynamic Random Access Memory)

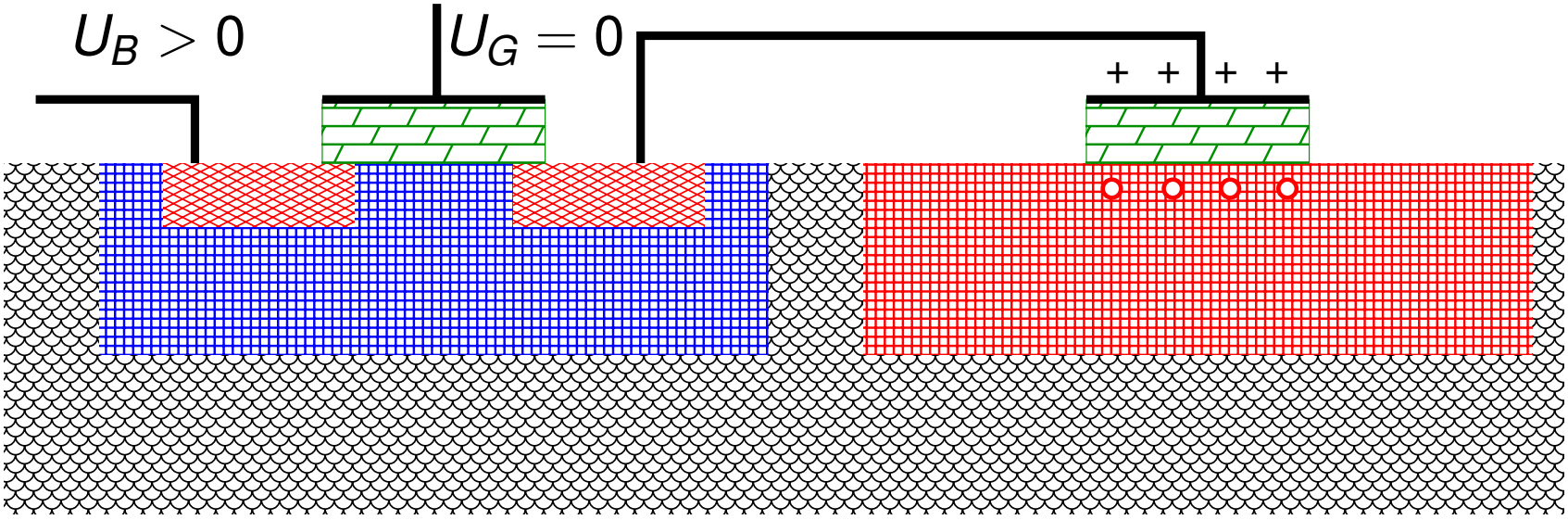
On utilise des capacitances pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires DRAM (Dynamic Random Access Memory)

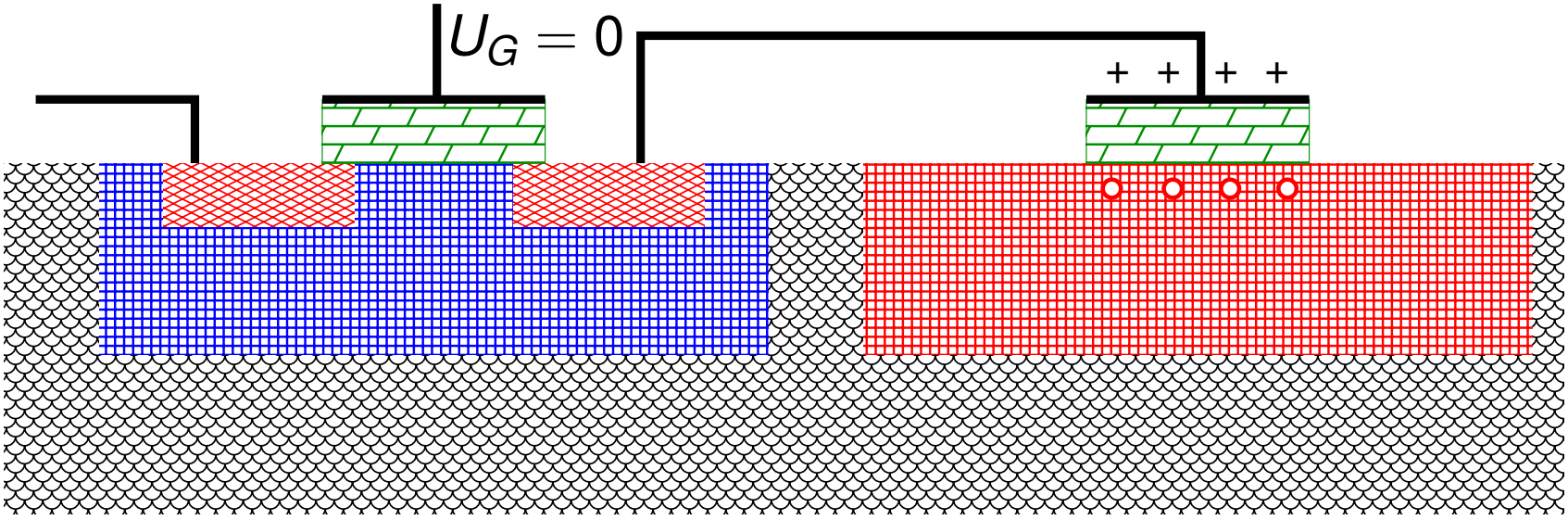
On utilise des capacitances pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires DRAM (Dynamic Random Access Memory)

On utilise des capacitances pour sauvegarder des bits



$\longrightarrow$ deux transistors nécessaires pour maintenir un bit

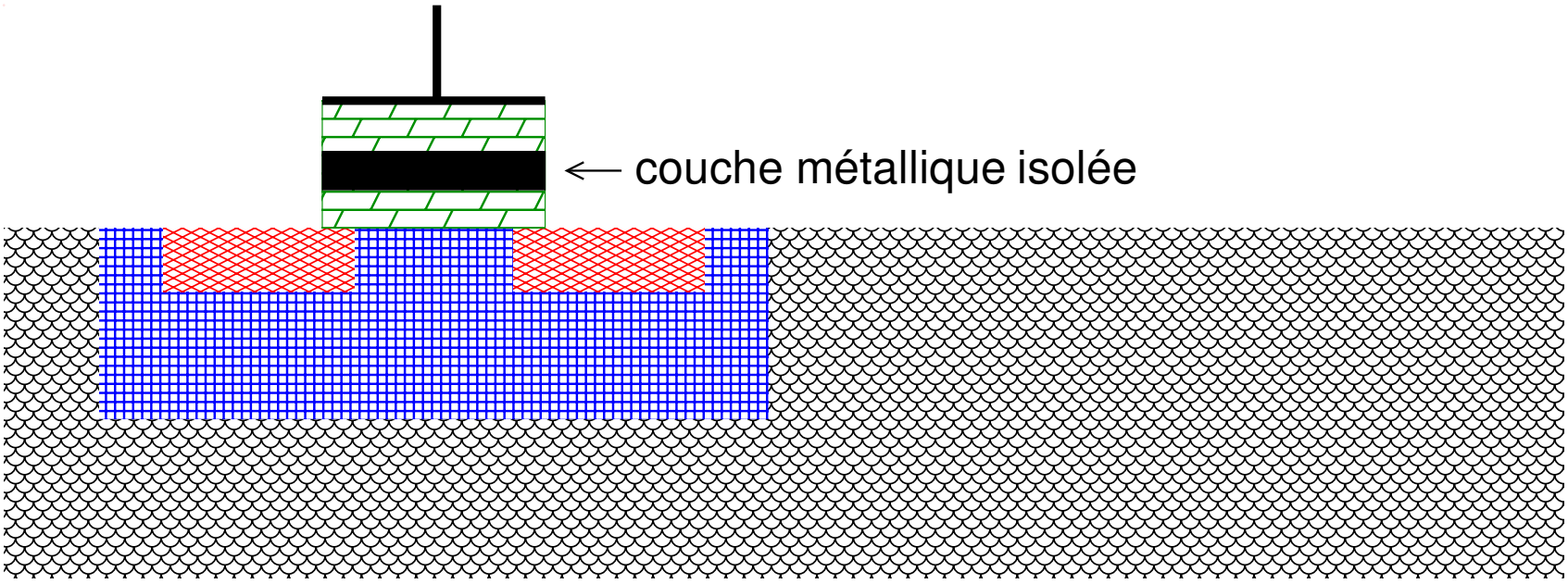
$\longrightarrow$ pas de stabilité, à cause de courants de fuite

$\rightarrow$ circuits de rafraîchissement nécessaires

Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires Flash (Toshiba, 1980)

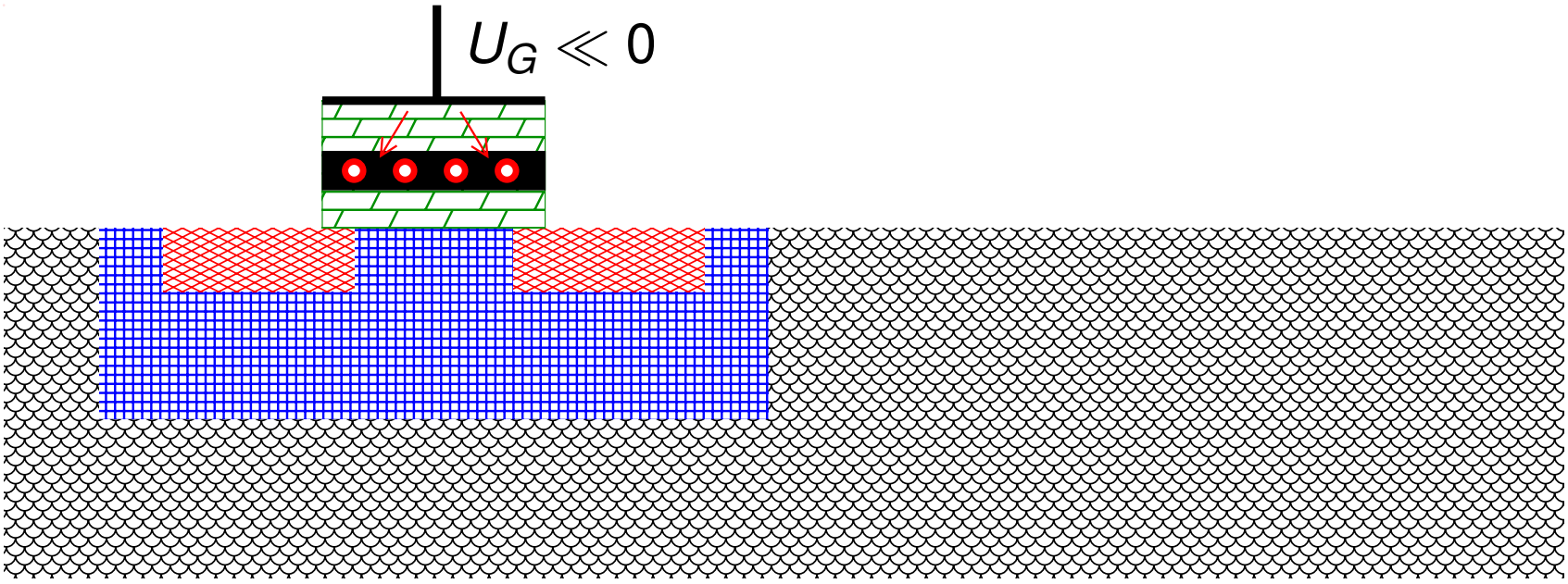
On utilise des capacitances isolées pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires Flash (Toshiba, 1980)

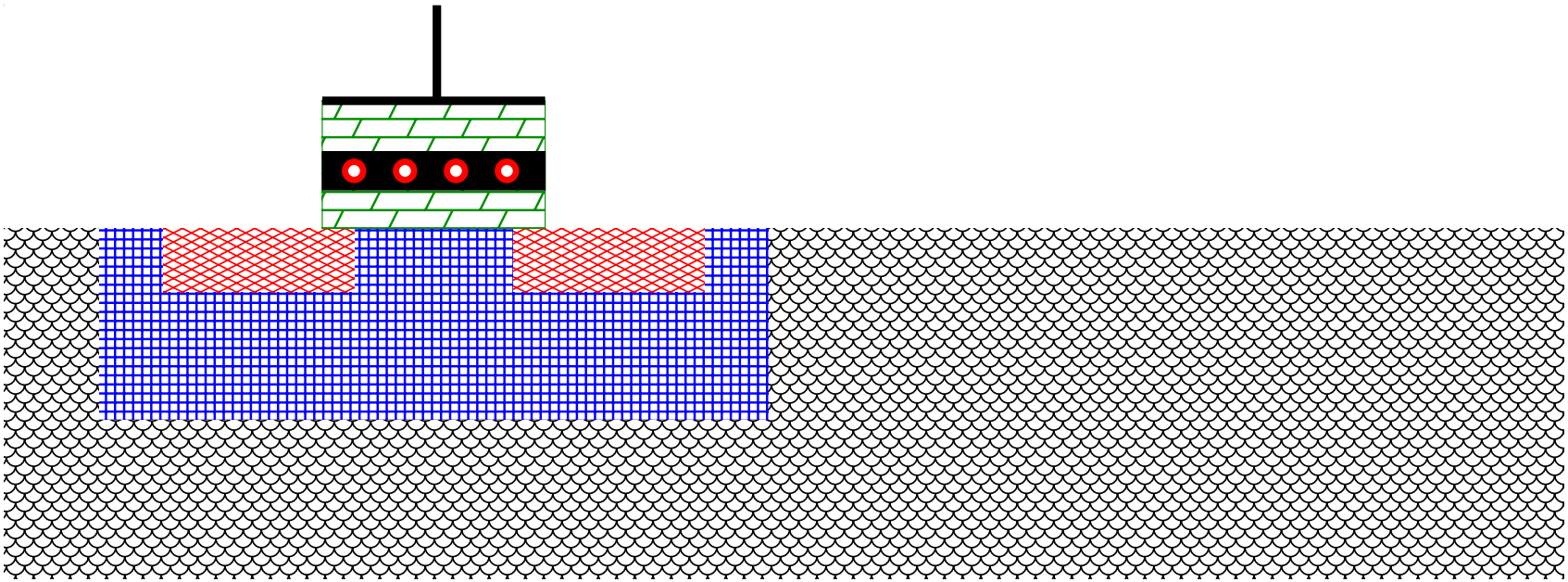
On utilise des capacitances isolées pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires Flash (Toshiba, 1980)

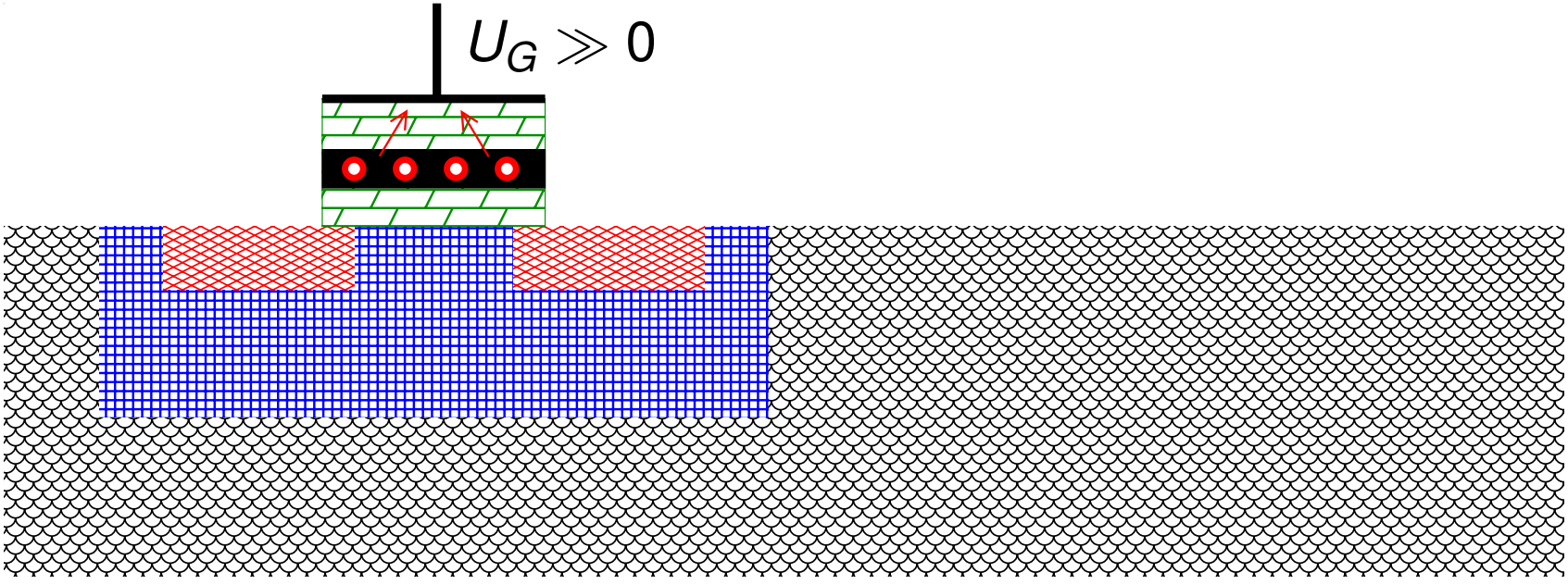
On utilise des capacitances isolées pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires Flash (Toshiba, 1980)

On utilise des capacitances isolées pour sauvegarder des bits

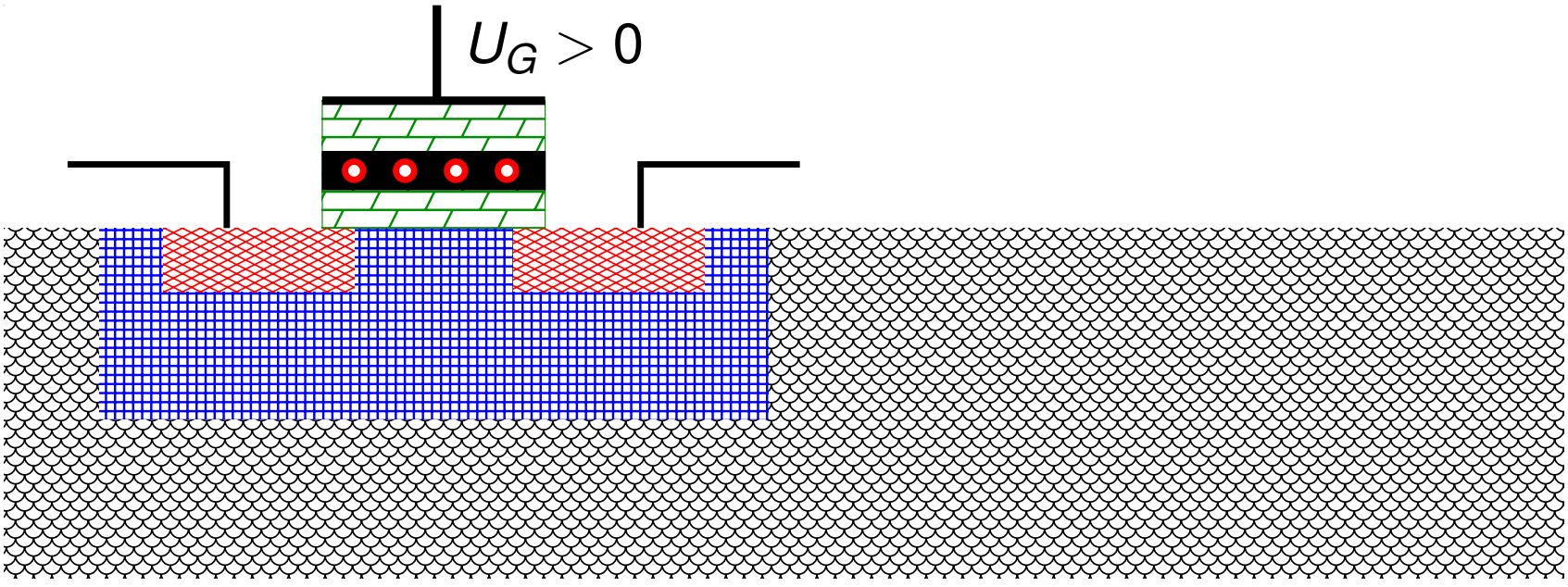


$\longrightarrow$ charger et décharger avec des tensions U_G très fortes (effet tunnel quantique)

Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires Flash (Toshiba, 1980)

On utilise des capacitances isolées pour sauvegarder des bits

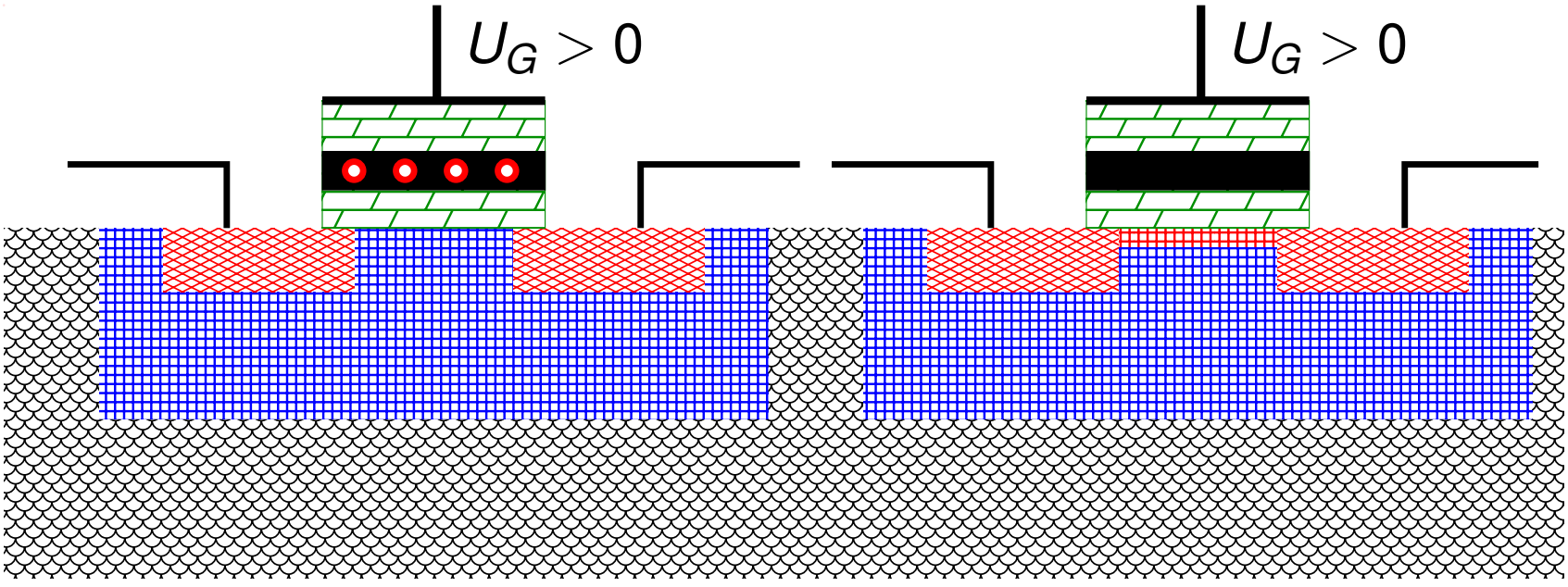


$\longrightarrow$ pas de courant au travers du transistor
si la capacitance est chargée (effet de screening)

Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires Flash (Toshiba, 1980)

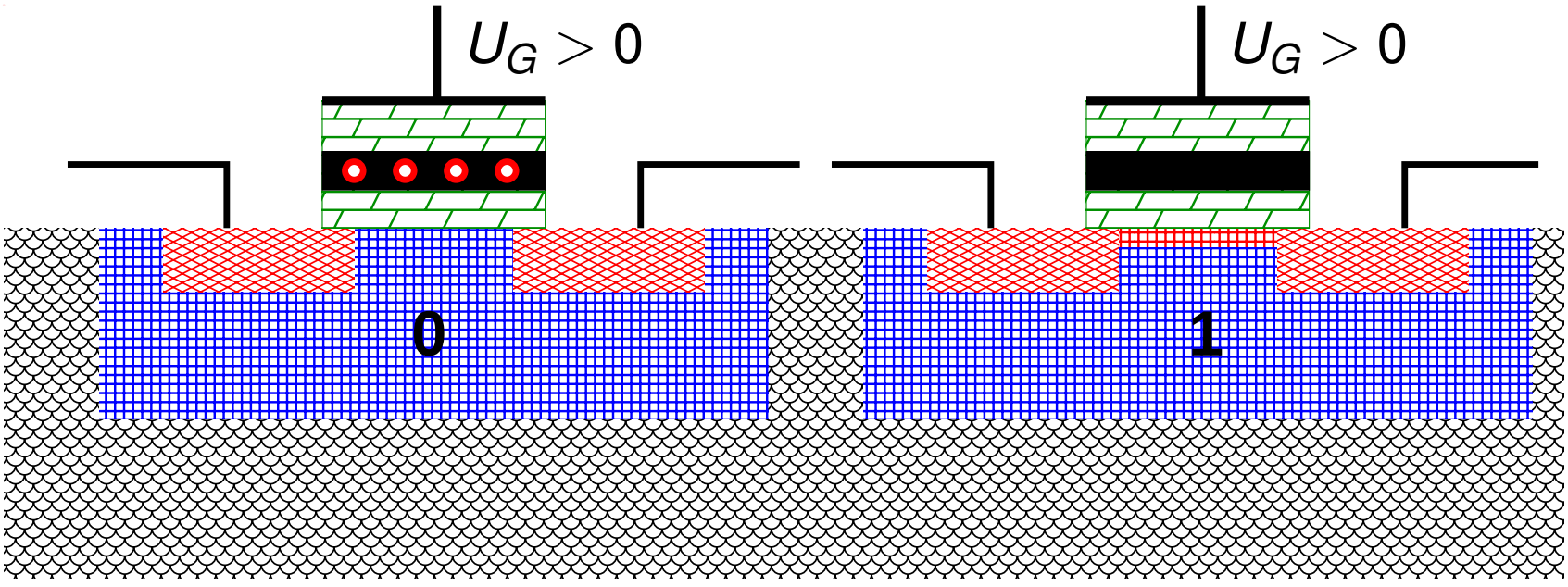
On utilise des capacitances isolées pour sauvegarder des bits



Les mémoires binaires sur des circuits intégrés

$\longrightarrow$ les mémoires Flash (Toshiba, 1980)

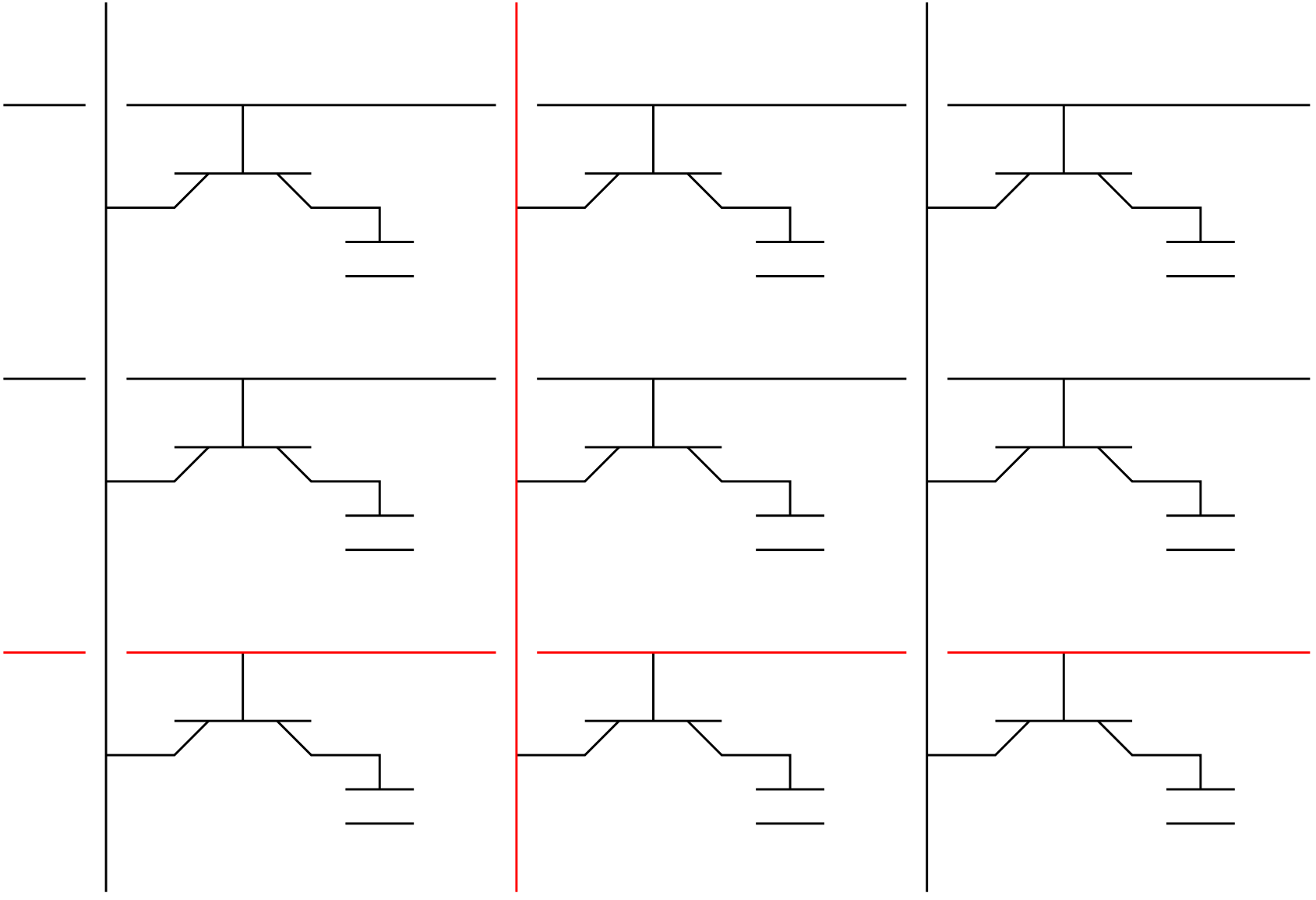
On utilise des capacitances isolées pour sauvegarder des bits



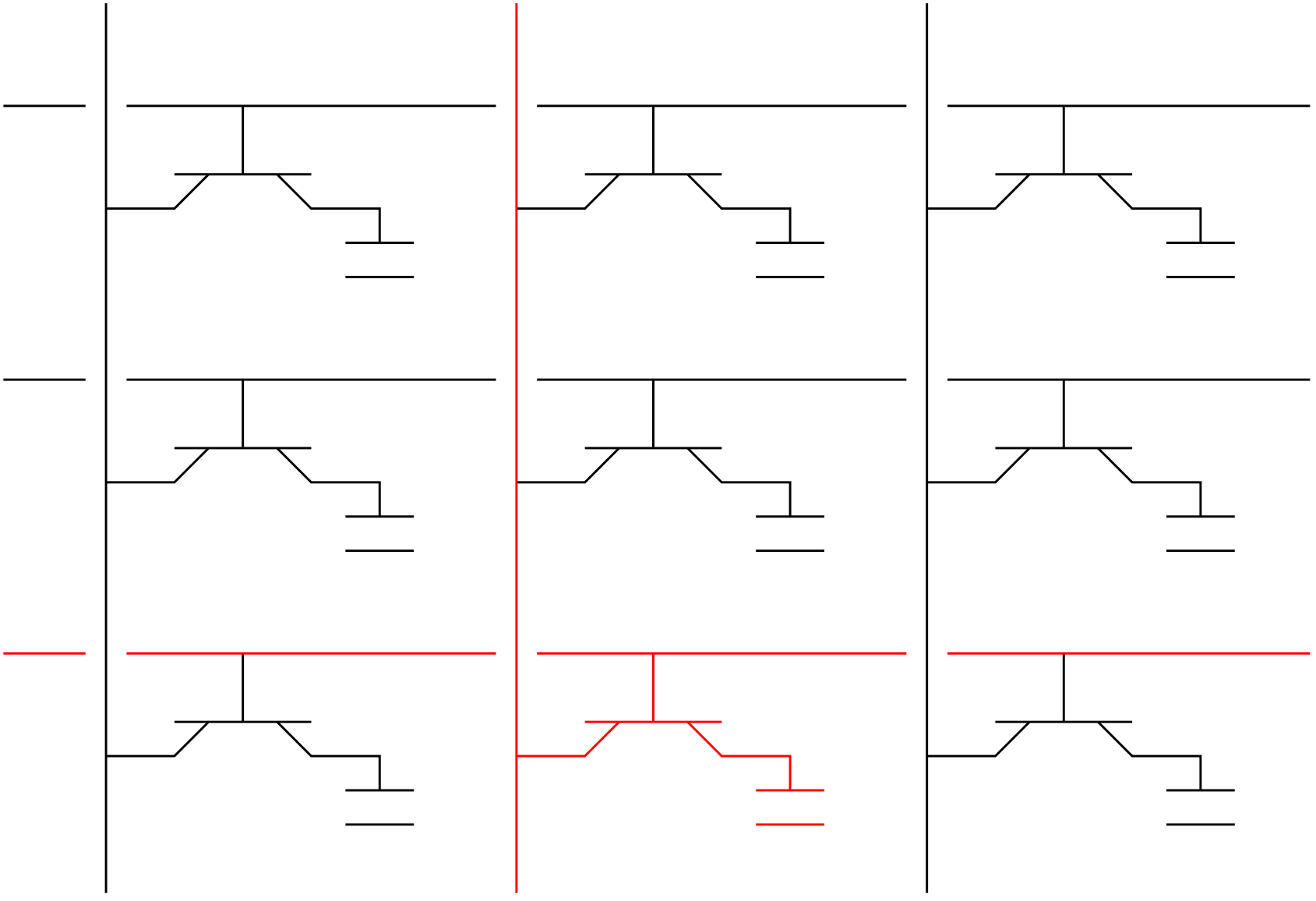
$\longrightarrow$ très longue durée de vie de la mémoire
même si le circuit est découpé de la source de tension

$\longrightarrow$ utilisé dans les GSM, les clés USB, les caméras ...

Les mémoires binaires sur des circuits intégrés



Les mémoires binaires sur des circuits intégrés



2.3 L'octet (byte)

$=$ la combinaison de 8 bits

0 1 0 1 1 0 1 1

$\longrightarrow$ l'unité de base pour des mémoires informatiques:

1 ko (kB) $=$ 1 kilooctet $=$ 10^3 octets = 1000 octets

1 Mo (MB) $=$ 1 mégaoctet $=$ 10^6 octets = 1 000 000 octets

1 Go (GB) $=$ 1 gigaoctet $=$ 10^9 octets = 1 000 000 000 octets

1 To (TB) $=$ 1 téraoctet $=$ 10^{12} octets = 1 000 000 000 000 octets

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

0 0 0 0 0 0 0 0

0

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

0 0 0 0 0 0 0 1

1

2.3 L'octet (byte)

= la combinaison de 8 bits

0 0 0 0 0 0 1 0

2

2.3 L'octet (byte)

= la combinaison de 8 bits

0 0 0 0 0 0 1 1

3

2.3 L'octet (byte)

= la combinaison de 8 bits

0 0 0 0 0 1 0 0

4

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

0 0 0 0 1 0 0 0

8

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

0 0 0 1 0 0 0 0

16

2.3 L'octet (byte)

= la combinaison de 8 bits

0 0 1 0 0 0 0 0

32

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

0 1 0 0 0 0 0 0

64

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

1 0 0 0 0 0 0 0

128

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$$

$\longrightarrow$ codage des nombres entiers entre 0 et $2^8 - 1 = 255$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$$

$\longrightarrow$ codage des nombres entiers entre 0 et $2^8 - 1 = 255$

. . . ou des nombres entiers, positifs et négatifs, entre

$$-2^7 = -128 \text{ et } 2^7 - 1 = \mathbf{127}$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

$\longrightarrow$ codage des nombres entiers entre 0 et $2^8 - 1 = 255$

. . . ou des nombres entiers, positifs et négatifs, entre

$$-2^7 = -\mathbf{128} \text{ et } 2^7 - 1 = 127$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$$

$$-1$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1\ 1\ 1\ 1\ 1\ 1\ 1\ 0$$

$$-2$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1\ 1\ 1\ 1\ 1\ 1\ 0\ 1$$

$$-3$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0$$

-4

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1\ 1\ 1\ 1\ 1\ 0\ 1\ 1$$

$$-5$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1\ 1\ 1\ 1\ 1\ 0\ 1\ 0$$

$$-6$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1$$

$$-7$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0$$

-8

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0$$

-16

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

$$-32$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

$$-64$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

$$-128$$

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1\ 0\ 1\ 1\ 0\ 1\ 0\ 1$$

Détermination des nombres négatifs:

$\longrightarrow$ négation de tous les bits

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

0 1 0 0 1 0 1 0

Détermination des nombres négatifs:

$\longrightarrow$ négation de tous les bits

$\longrightarrow$ addition de 1

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1$$

Détermination des nombres négatifs:

$\longrightarrow$ négation de tous les bits

$\longrightarrow$ addition de 1

75

2.3 L'octet (byte)

$=$ la combinaison de 8 bits

$$1\ 0\ 1\ 1\ 0\ 1\ 0\ 1$$

Détermination des nombres négatifs:

$\longrightarrow$ négation de tous les bits

$\longrightarrow$ addition de 1

-75

Addition des octets

$$\begin{array}{rcccccccc} & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ + & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \qquad \begin{array}{rr} & 91 \\ + & -106 \\ \hline & -15 \end{array}$$

Addition des octets

$$\begin{array}{rcccccccc} & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ + & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \qquad \begin{array}{rrl} & 91 & \\ + & 86 & \\ \hline & -79 & ?!? \end{array}$$

. . . c'est exact à l'exception d' erreurs d'overflow

Addition des octets

$$\begin{array}{rcccccccc} & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ + & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}$$

$$\begin{array}{rr} & 91 \\ + & -42 \\ \hline & 49 \quad \text{o.k.} \end{array}$$

Addition des octets

Implémentation sur un circuit intégré:

$$\begin{array}{cccccccccc} & a_8 & a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 \\ + & b_8 & b_7 & b_6 & b_5 & b_4 & b_3 & b_2 & b_1 \\ \hline & c_8 & c_7 & c_6 & c_5 & c_4 & c_3 & c_2 & c_1 \end{array}$$

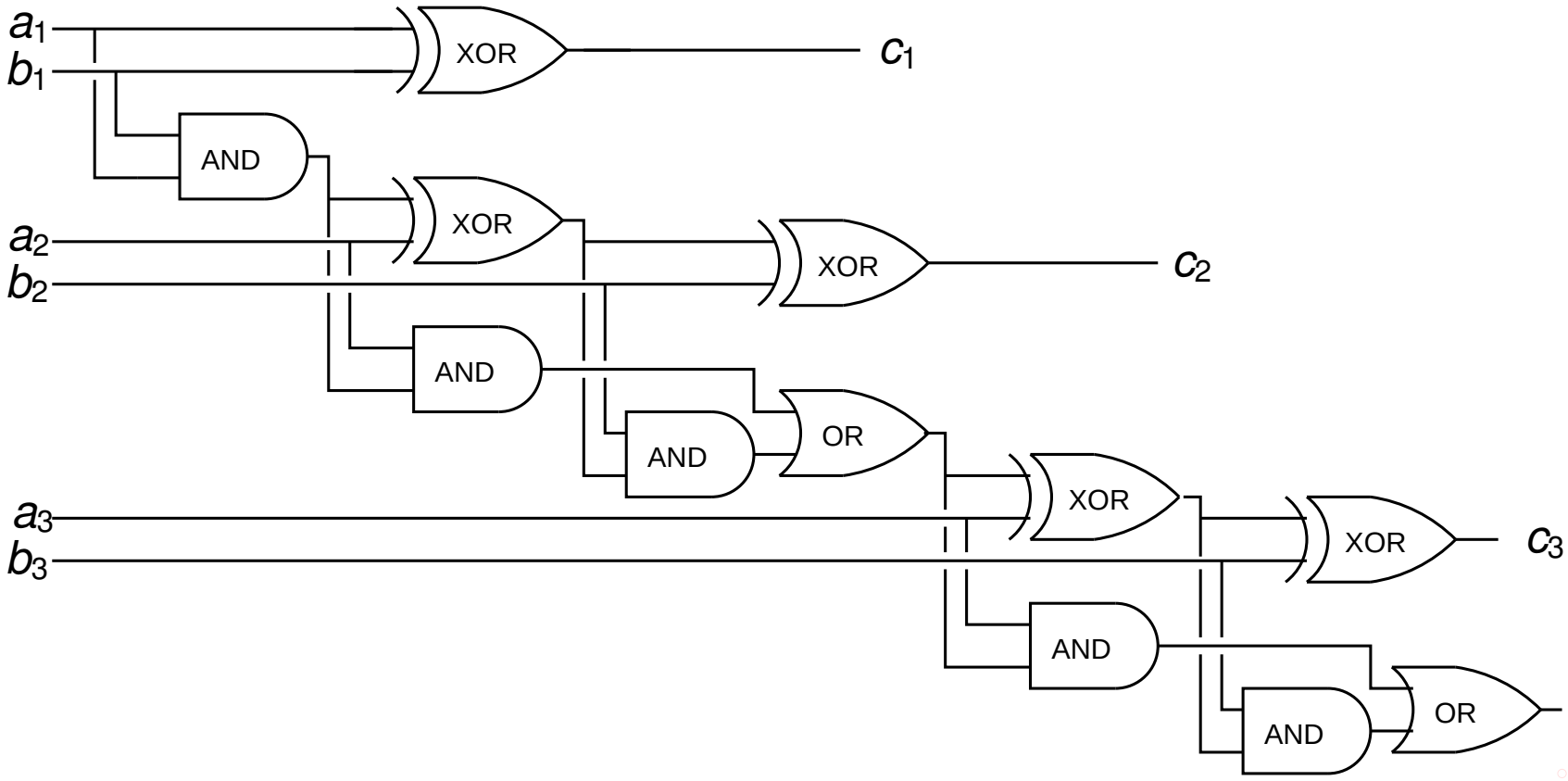
$\longrightarrow$ opération XOR pour l'addition de a_1 et b_1, a_2 et b_2, ...

$\longrightarrow$ opération AND pour ajouter 1 à la colonne suivante si $a_1 = 1$ et $b_1 = 1$ (pareil pour a_2 et b_2, ...)

$\longrightarrow$ opération OR pour faire la même chose dans les colonnes suivantes

Addition des octets

	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1
+	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
	c_8	c_7	c_6	c_5	c_4	c_3	c_2	c_1



Soustraction des octets

$\longrightarrow$ circuit similaire à celui pour les additions . . .

ou

$\longrightarrow$ multiplication du deuxième octet par -1

$\rightarrow$ négation de tous les bits (opération NOT)

$\rightarrow$ addition de 1

$\longrightarrow$ addition ordinaire des deux octets

Représentation des nombres entiers plus grands

$\longrightarrow$ combinaison de plusieurs octets

1 0 0 1 1 0 1 0 0 1 0 0 1 1 1 1

2 octets: représentation des nombres entre -2^{15} et $2^{15} - 1$

c-à-d entre $-32\,768$ et $32\,767$

4 octets: représentation des nombres entre -2^{31} et $2^{31} - 1$

c-à-d entre $-2\,147\,483\,648$ et $2\,147\,483\,647$

Représentation des nombres réels

Représentation “scientifique” du nombre 325 dans le système décimal :

$$3{,}25 \quad 02$$

Représentation des nombres réels

Représentation “scientifique” du nombre 325 dans le système binaire :

`1,01000101  1000`

mantisse exposant (puissance de 2)

$\longrightarrow$ nombres “flottants” (floating point numbers)

Représentation des nombres réels

Représentation “scientifique” du nombre 325 dans le système binaire :

`1,01000101  1000`

mantisse exposant (puissance de 2)

$\longrightarrow$ nombres “flottants” (floating point numbers)

Exemple:

3 octets pour la mantisse et 1 octet pour l’exposant
(type “float” en C/C++)

$\longrightarrow$ représentation des nombres réels entre

$$\pm 1,2 \times 10^{-38} \text{ et } \pm 3,4 \times 10^{38}$$

avec 7 chiffres significatifs

Représentation des nombres réels

Représentation “scientifique” du nombre 325 dans le système binaire :

`1,01000101  1000`

mantisse exposant (puissance de 2)

$\longrightarrow$ nombres “flottants” (floating point numbers)

Exemple:

6,5 octets pour la mantisse et 1,5 octets pour l’exposant (type “double” en C/C++)

$\longrightarrow$ représentation des nombres réels entre

$$\pm 2,2 \times 10^{-308} \text{ et } \pm 1,8 \times 10^{308}$$

avec 14 chiffres significatifs

Addition des nombres réels

Illustration dans le système décimal:

$1435,12 + 0,119421 = 1435,239421$

Calcul “numérique” avec 6 chiffres significatifs:

$1,43512 \times 10^3 + 1,19421 \times 10^{-1} = 1,43523 \times 10^3 = 1435,23$

$\longrightarrow$ attention: les derniers chiffres sont coupés !

$\longrightarrow$ erreur numérique

peut-être pas si grave ?

Addition des nombres réels

Illustration dans le système décimal:

$1435,12 + 0,119421 = 1435,239421$

Calcul “numérique” avec 6 chiffres significatifs:

$1,43512 \times 10^3 + 1,19421 \times 10^{-1} = 1,43523 \times 10^3 = 1435,23$

$\longrightarrow$ attention: les derniers chiffres sont coupés !

$\longrightarrow$ erreur numérique

$$\left[(1 + 10^{-3}) - 1\right] \times 10^3 \;\Rightarrow\; 1$$

Addition des nombres réels

Illustration dans le système décimal:

$1435,12 + 0,119421 = 1435,239421$

Calcul “numérique” avec 6 chiffres significatifs:

$1,43512 \times 10^3 + 1,19421 \times 10^{-1} = 1,43523 \times 10^3 = 1435,23$

$\longrightarrow$ attention: les derniers chiffres sont coupés !

$\longrightarrow$ erreur numérique

$$\left[(1 + 10^{-5}) - 1\right] \times 10^5 \;\Rightarrow\; 1$$

Addition des nombres réels

Illustration dans le système décimal:

$1435,12 + 0,119421 = 1435,239421$

Calcul “numérique” avec 6 chiffres significatifs:

$1,43512 \times 10^3 + 1,19421 \times 10^{-1} = 1,43523 \times 10^3 = 1435,23$

$\longrightarrow$ attention: les derniers chiffres sont coupés !

$\longrightarrow$ erreur numérique

$$\left[(1 + 10^{-7}) - 1\right] \times 10^7 \;\Rightarrow\; 0$$

$\longrightarrow$ attention avec des différences !

Représentation des lettres: le code ASCII

(American Standard Code for Information Interchange)

$\longrightarrow$ représentation des lettres de l'alphabet et des chiffres 0, 1, . . . 9 avec 7 bits:

bin.	d.		bin.	d.		bin.	d.	
1000001	65	A	1100001	97	a	0110000	48	0
1000010	66	B	1100010	98	b	0110001	49	1
1000011	67	C	1100011	99	c	0110010	50	2
1000100	68	D	1100100	100	d	0110011	51	3
⋮	⋮		⋮	⋮		⋮	⋮	
1011010	90	Z	1111010	122	z	0111001	57	9

$\longrightarrow$ les autres nombres sont réservés pour des caractères spéciaux: ! " # % $ & ' ( ) $*$ + , - / ; $<$ $>$? . . .