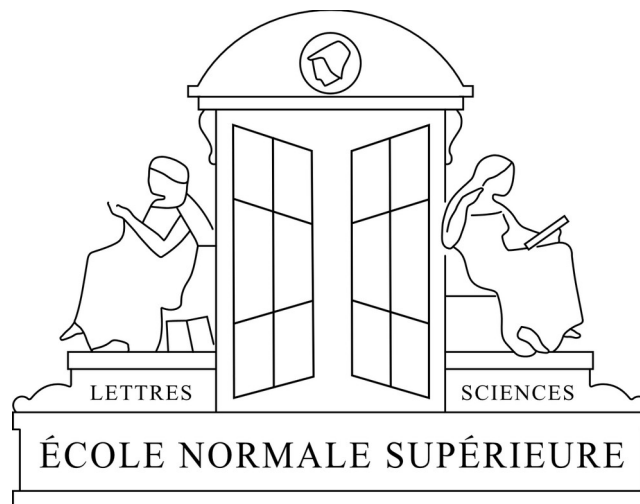


Cours "**Systemes numériques :** **de l'algorithme aux circuits**"

Leçon inaugurale
Septembre

Sylvain GUILLEY

`<sylvain.guilley@telecom-paristech.fr>`



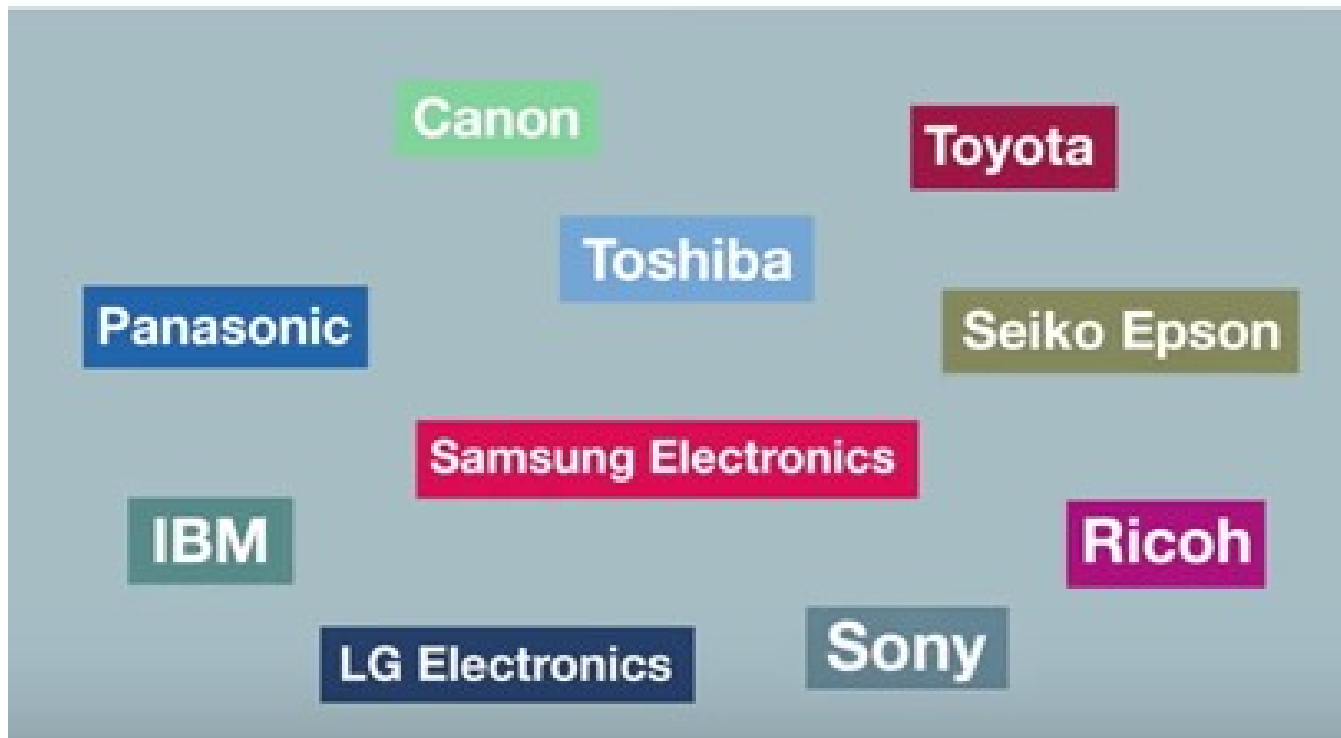
Program

- How to design and use an electronic circuit
- Emphasis on : energy and time
- Technologies : FPGA, ASIC
- Complexity management
- CPU and programmation

- Team:
 - Sylvain GUILLEY
 - Hadrien BARRAL et Théophile WALLEZ,
Sumanta CHAUDHURI et Yves MATHIEU

Electronics : WIPO report

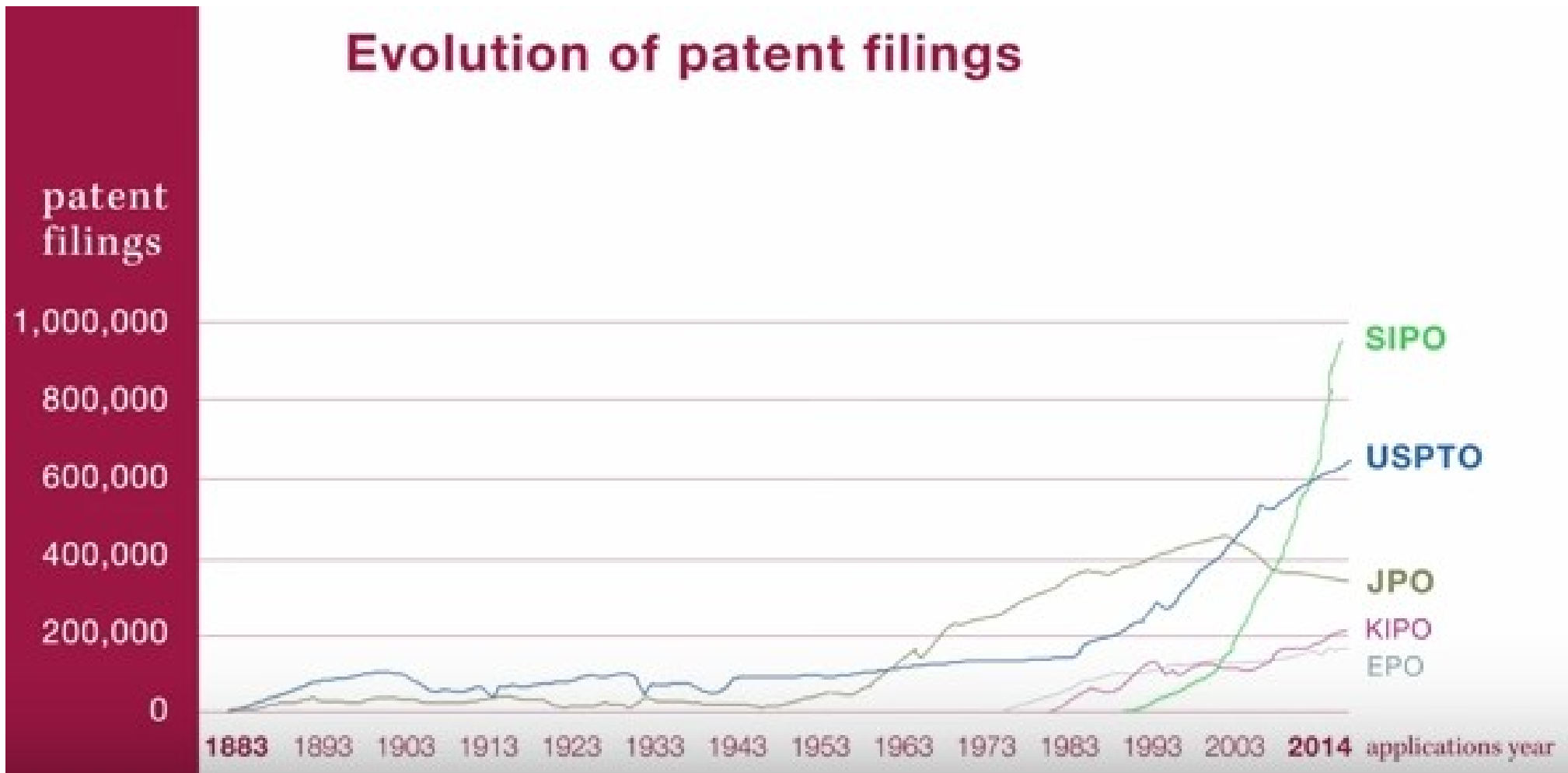
Top patent filers



zaibatsu (財閥)

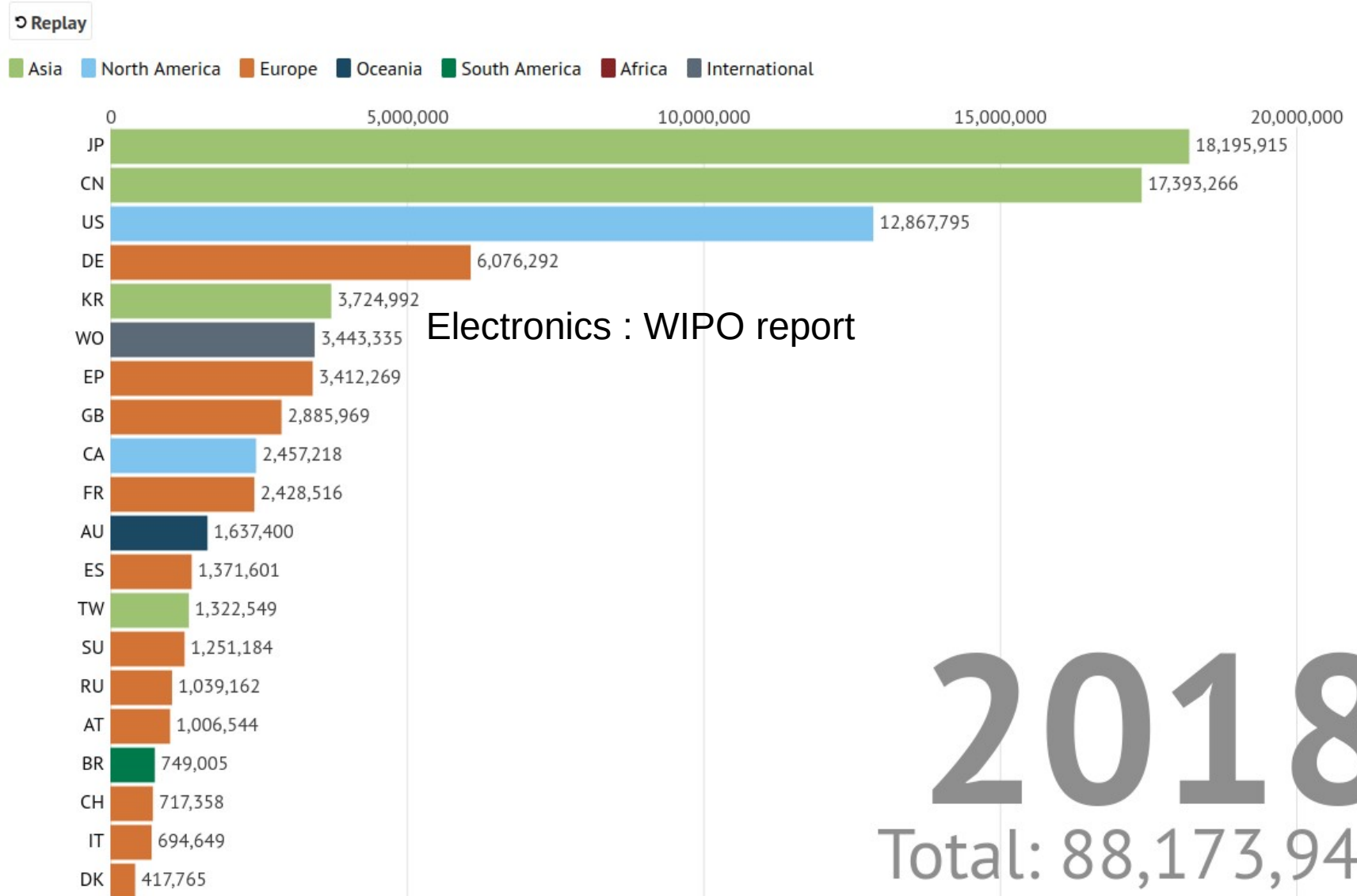
Electronics : WIPO report

Evolution of patent filings



Electronics : WIPO report

Patent Applications per Country/Patent Authority since 1985



Electronics profits

Fab-less industries

Factories



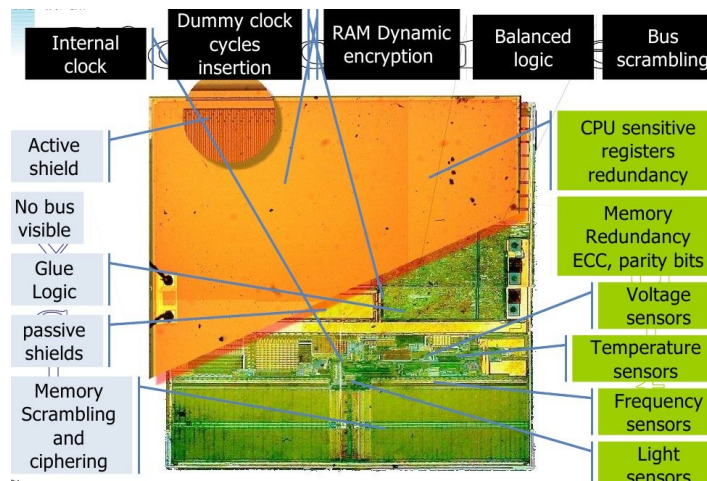
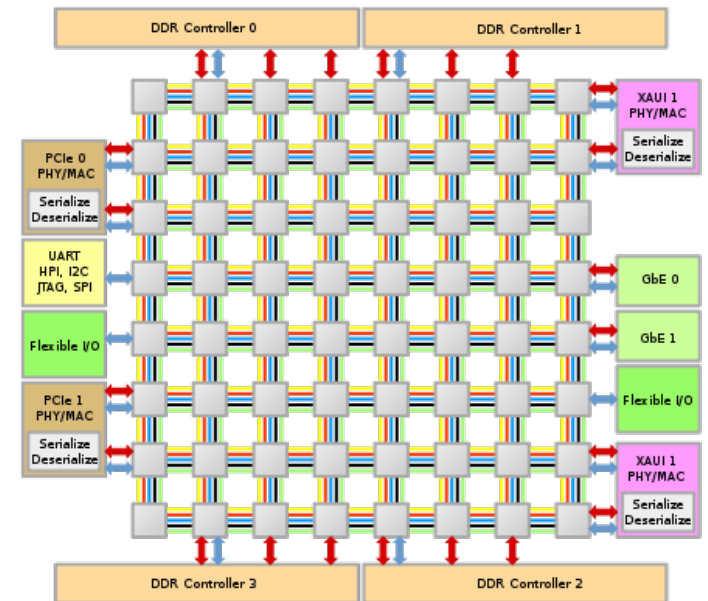
1,700 Meuros

2013 Rank	2012 Rank	Company	Headquarters	2012 (\$M)	2013 (\$M)	% Change
1	1	Qualcomm	U.S.	13,177	17,211	31%
2	2	Broadcom	U.S.	7,793	8,219	5%
3	3	AMD	U.S.	5,422	5,299	-2%
4	5	MediaTek	Taiwan	3,366	4,587	36%
5	4	Nvidia	U.S.	3,965	3,898	-2%
6	6	Marvell	U.S.	3,144	3,352	7%
7	7	LSI	U.S.	2,506	2,370	-5%
8	8	Xilinx	U.S.	2,196	2,297	5%
9	9	Altera	U.S.	1,783	1,732	-3%
10	10	Avago	Singapore	1,479	1,619	9%
11	12	Novatek	Taiwan	1,256	1,398	11%
12	13	HiSilicon	China	1,178	1,355	15%
13	11	MStar	Taiwan	1,271	1,136	-11%
14	18	Spreadtrum	China	725	1,070	48%
15	14	CSR	Europe	1,025	961	-6%
16	15	Realtek	Taiwan	836	951	14%
17	16	Dialog	Europe	774	903	17%
18	19	Cirrus Logic	U.S.	714	772	8%
19	17	Himax	Taiwan	737	771	5%
20	21	Silicon Labs	U.S.	563	580	3%
21	22	MegaChips	Japan	553	577	4%
22	24	Semtech	U.S.	518	555	7%
23	23	PMC-Sierra	U.S.	531	508	-4%
24	25	IDT	U.S.	497	475	-4%
25	26	Microsemi	U.S.	450	433	-4%
Top 25 Total			—	56,459	63,029	12%
Other Total			—	15,650	14,882	-5%
Total Fabless			—	72,109	77,911	8%

Source: Company reports, IC Insights' Strategic Reviews database

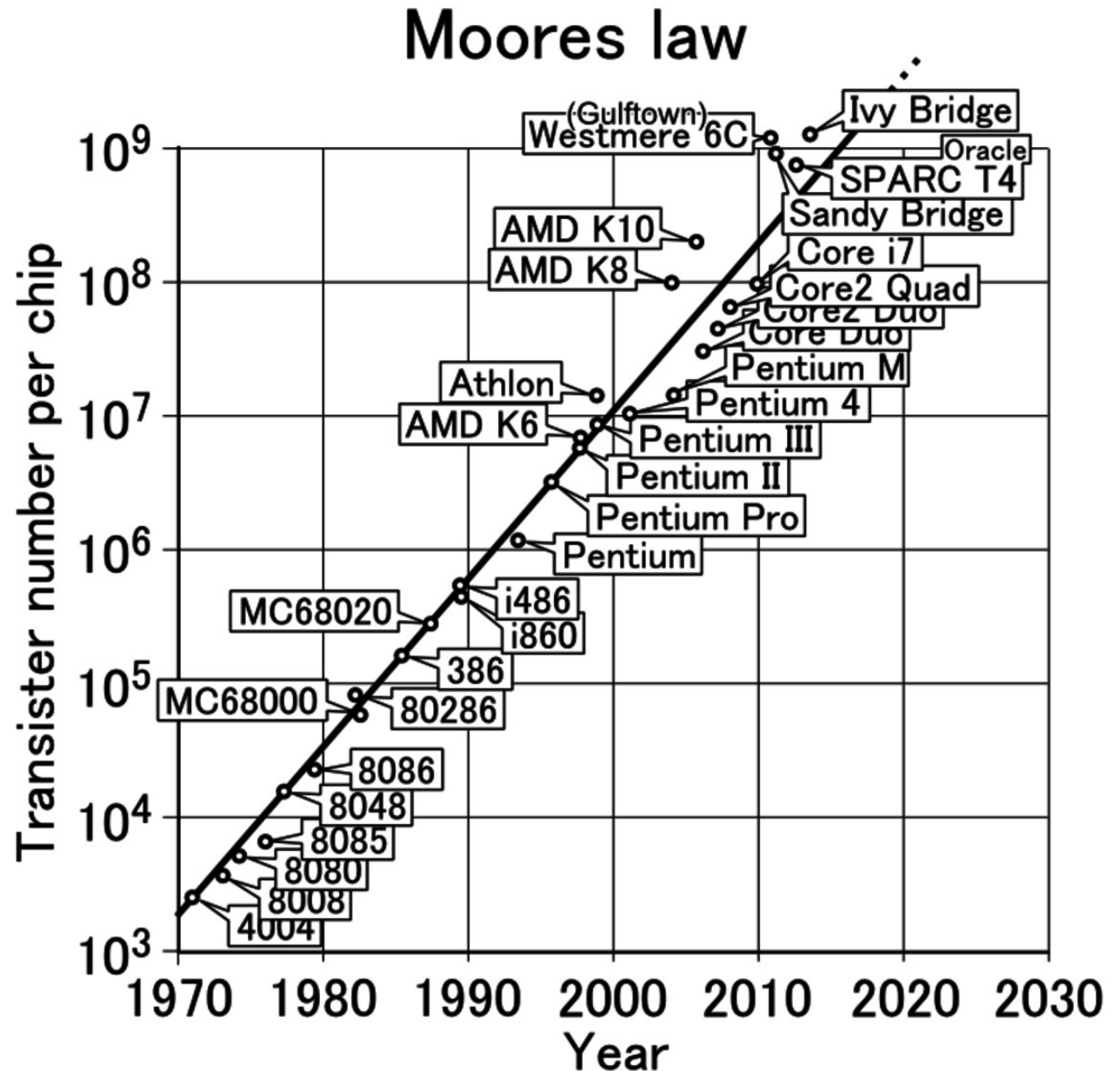
Why electronics ?

- Processors → software
- Performance : e.g., Kalray
- Low-power
- Security
- Safety
- ...

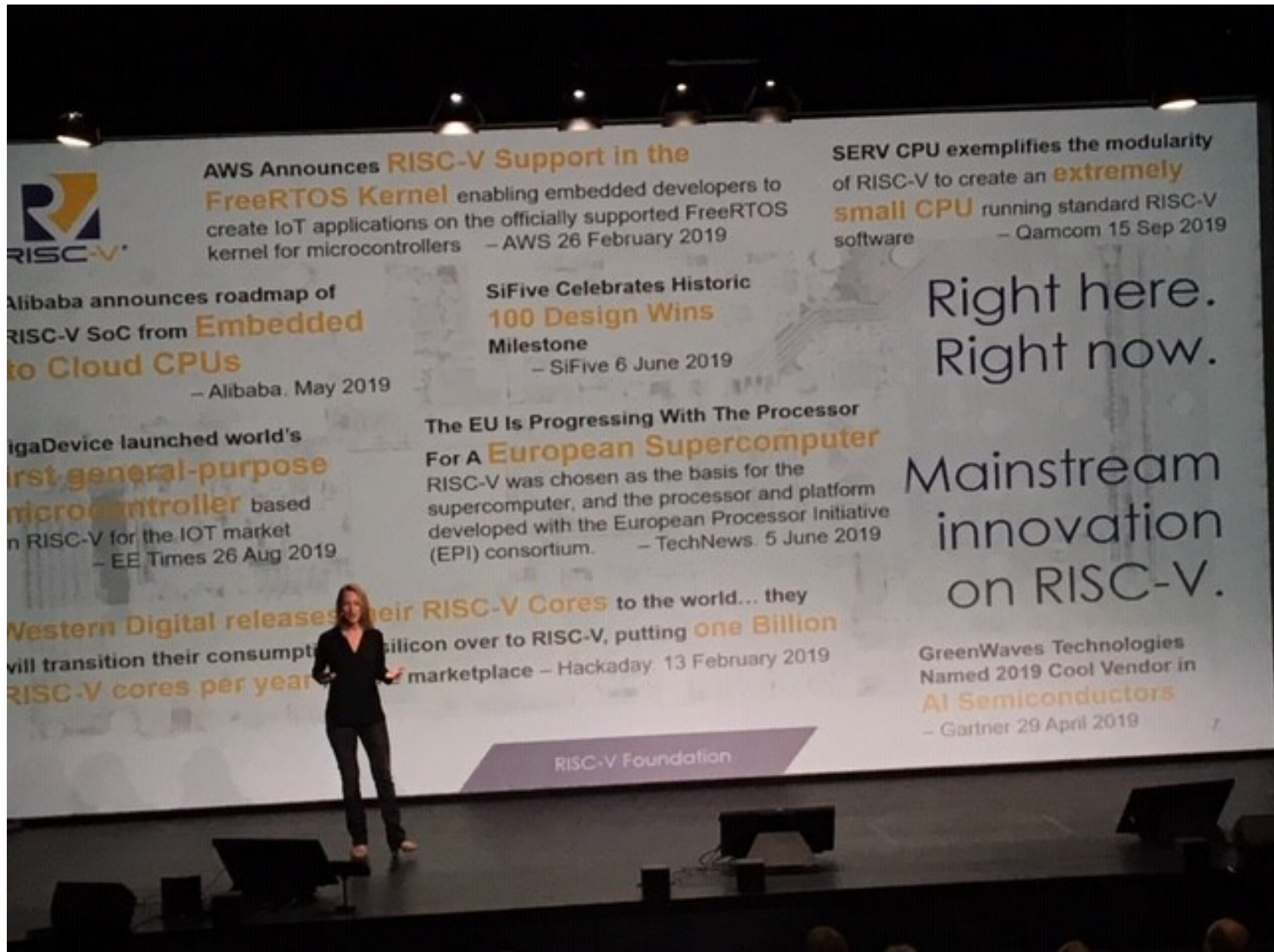


Records

- 10 nm feature size
- #transistors, exponential increase (2x/18month)
- More processors today than human beings

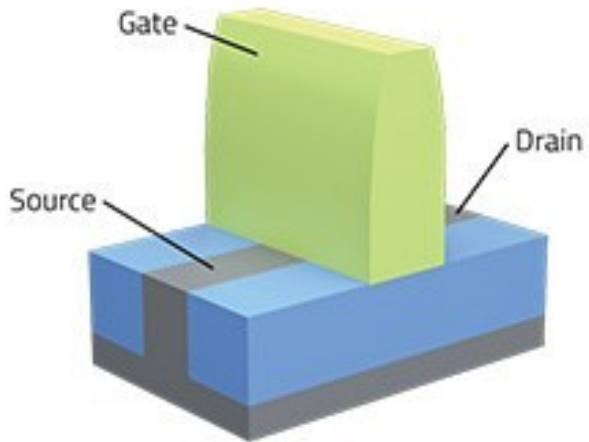


Trends : this morning at station F

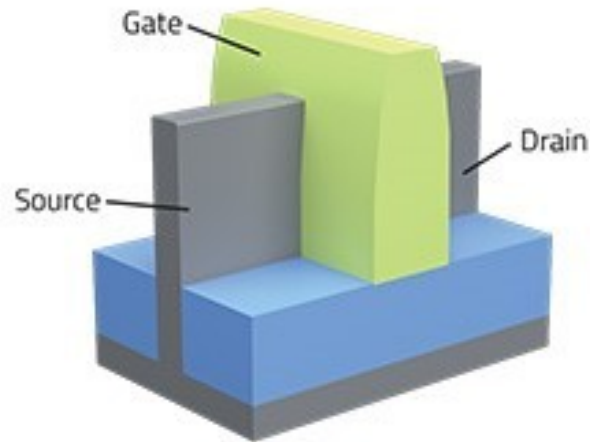


Trends : for 5 and 3 nm

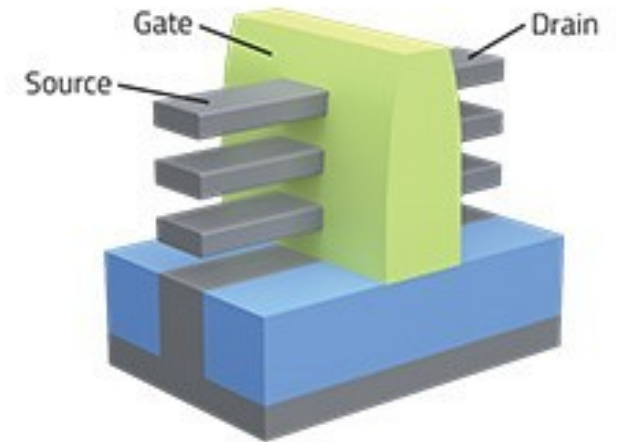
Planar



FinFET



Gate-All-Around

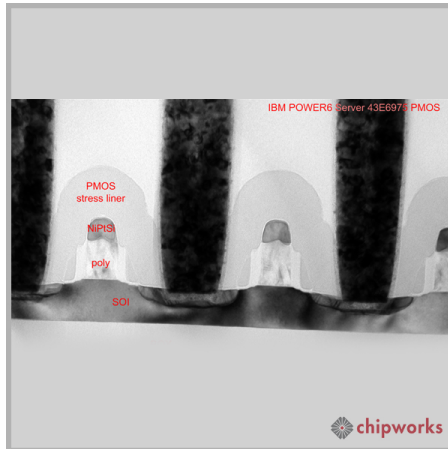


Transistor en 2004



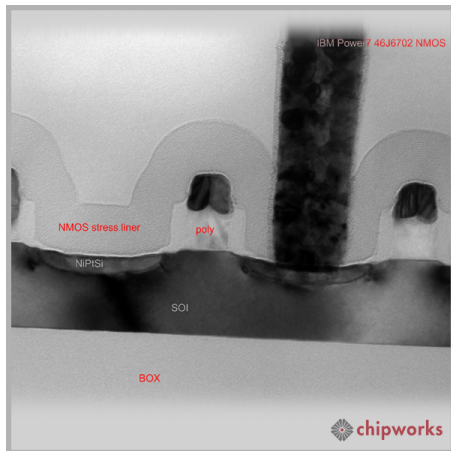
PPC970fx (90nm)

Transistor en 2008



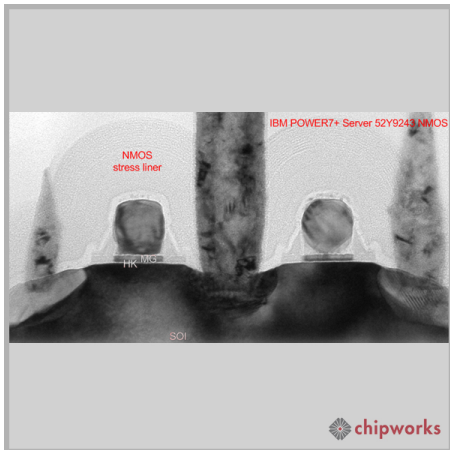
Power6 (65nm)

Transistor en 2011



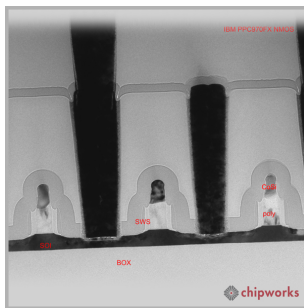
Power7 (45nm)

Transistor en 2013

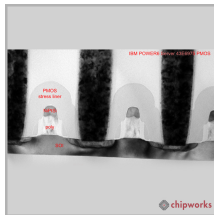


Power7+ (32nm)

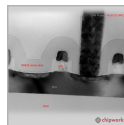
A l'échelle



2004
90nm
PPC970fx



2009
65nm
Power6



2011
45nm
Power7



2013
32 nm
Power7+

Les images sont issues de l'analyse de l'évolution des technologies IBM faite par ChipWorks Inc.
L'analyse ainsi que les images originales étaient accessibles en 2014 ici :

[http://www.chipworks.com/en/
technical-competitive-analysis/resources/blog/
ibm-continues-major-source-chip-innovation/](http://www.chipworks.com/en/technical-competitive-analysis/resources/blog/ibm-continues-major-source-chip-innovation/)

Changement de technologie

"Downsizing théorique"

■ Générations technologiques :

- La longueur de grille minimale est caractéristique d'une génération technologique (90nm, 65nm, 40nm, 28nm, ...)
- À chaque génération, les *fondeurs* visent une réduction de la surface d'un facteur **2**
- Les *fondeurs* investissent les milliards nécessaires pour cela ...

Changement de technologie

"Downsizing théorique"

- Générations technologiques :
 - La longueur de grille minimale est caractéristique d'une génération technologique (90nm, 65nm, 40nm, 28nm, ...)
 - À chaque génération, les *fondeurs* visent une réduction de la surface d'un facteur **2**
 - Les *fondeurs* investissent les milliards nécessaires pour cela ...
- On utilise un facteur de réduction $\beta = \sqrt{2}$
 - division par β de la largeur W et la longueur L des transistors
 - division par β de l'épaisseur d'oxyde de grille T_{OX}
 - division par β de la tension d'alimentation V_{dd} des circuits
 - division par β de la tension de seuil V_T des transistors

Changement de technologie

Conséquences sur les performances

Évolution des capacités parasites

$$C_{par}(\beta) = (W/\beta)(L/\beta)(\beta C'_{ox}) = \frac{C_{par}}{\beta}$$

Évolution de l'énergie consommée par une porte

$$E_{porte}(\beta) = \frac{C_{par}}{\beta} \left(\frac{V_{dd}}{\beta} \right)^2 = \frac{E_{porte}}{\beta^3}$$

Évolution du temps de propagation des fonctions combinatoires

$$t_{calc}(\beta) = \frac{t_{calc}}{\beta}$$

Changement de technologie

Diminuer les coûts et la consommation

- On n'exploite pas le gain en vitesse
 - $F_h(\beta) = F_h$
- Le gain en surface fait diminuer les prix
 - $Surf(\beta) = \frac{Surf}{\beta^2}$
- La consommation diminue.
 - $P_{circuit}(\beta) = T_{act}(F_h) \frac{E_{circuit}}{\beta^3} = \frac{P_{circuit}}{\beta^3}$

Changement de technologie

Diminuer les coûts et la consommation

- On n'exploite pas le gain en vitesse
 - $F_h(\beta) = F_h$
- Le gain en surface fait diminuer les prix
 - $Surf(\beta) = \frac{Surf}{\beta^2}$
- La consommation diminue.
 - $P_{circuit}(\beta) = T_{act}(F_h) \frac{E_{circuit}}{\beta^3} = \frac{P_{circuit}}{\beta^3}$
- Cette stratégie est particulièrement intéressante dans l'embarqué :
 - transition du haut de gamme vers le milieu, puis bas de gamme (smartphones),
 - ouverture à de nouvelles utilisations (objets connectés).

Changement de technologie

Augmenter les performances

- On exploite le gain en vitesse
 - $F_h(\beta) = \beta F_h$
- On profite du gain en taille des transistors pour accroître la complexité du circuit
 - $Surf(\beta) = Surf$
- La consommation ne change pas
 - $P_{circuit}(\beta) = P_{circuit}$

Changement de technologie

Augmenter les performances

- On exploite le gain en vitesse
 - $F_h(\beta) = \beta F_h$
- On profite du gain en taille des transistors pour accroître la complexité du circuit
 - $Surf(\beta) = Surf$
- La consommation ne change pas
 - $P_{circuit}(\beta) = P_{circuit}$
- Cette stratégie est particulièrement pour les processeurs de serveurs :
 - la puissance de calcul profite de l'augmentation de fréquence,
 - la puissance de calcul profite de l'augmentation du parallélisme.

Changement de technologie

Dans la pratique

Dans la pratique ça ne fonctionne plus si bien :

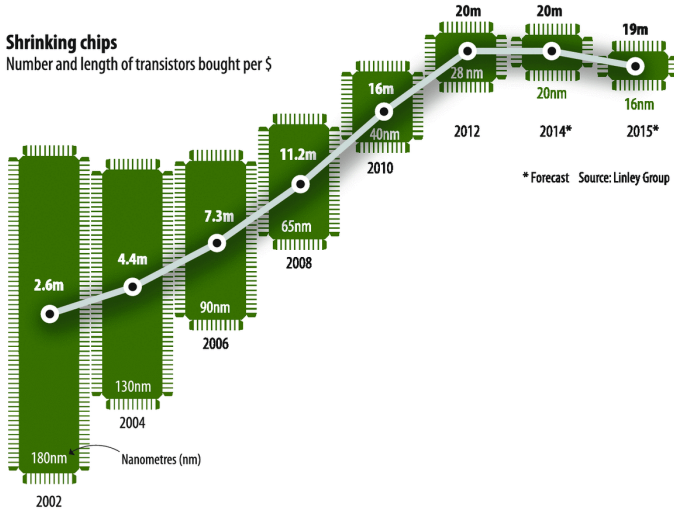
- Les vitesses maximum stagnent depuis le début des années 2000 (3 à 4 Ghz) à cause des problèmes de dissipation thermique.
- On ne peut diminuer sans cesse la tension d'alimentation sans s'éloigner du modèle d'interrupteur idéal : les circuits ont des courants de fuite de moins en moins négligeables
- Les technologues doivent jongler avec des procédés de fabrication de plus en plus complexes (et couteux) pour continuer à suivre la "loi de Moore".
- On a plusieurs fois prédit la fin de la loi de Moore pour des raisons "scientifiques" (physique du transistor) mais il semble depuis 2014 que le plus grave problème soit économique !

Evolution technologique

La fin de la loi de Moore ?

Shrinking chips

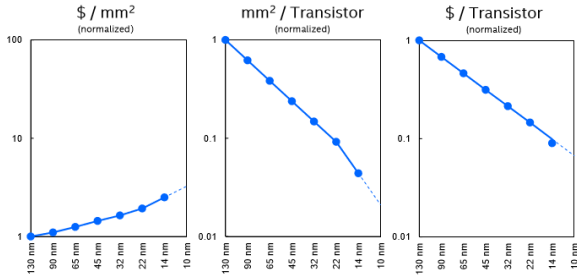
Number and length of transistors bought per \$



Evolution technologique

Pas pour Intel ?

(EP1) Moore's Law Challenges Below 10nm: Technology, Design and Economic Implications



Scaling continues to provide lower cost per transistor
Cost reduction is needed to justify new technology generations

Gordon Moore Fishing



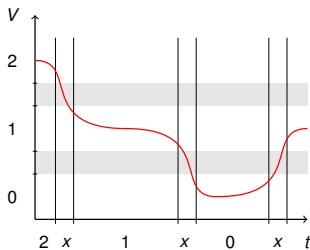
source https://commons.wikimedia.org/wiki/File:Gordon_moore_fishing.jpg

Le signal électrique

Support de l'information

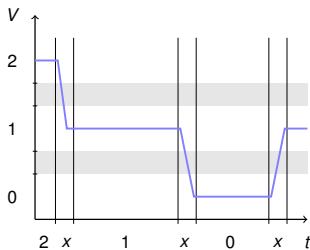
- Passer de grandeurs physiques à des signaux électriques grâce à :
 - Capteurs
 - Transducteurs
- Possibilité de mesurer/manipuler :
 - la tension
 - le courant
 - la charge ...
- C'est pour cela qu'on fait de l'électronique
 - analogique : traitement de valeurs continues
 - numérique : traitement de valeurs discrètes

Codage numérique de l'information



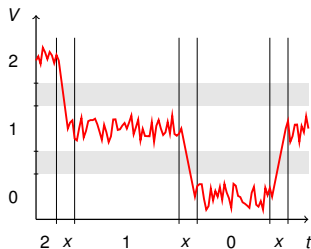
- Discrétiser le signal dans le temps
 - échantillonnage
- Représenter le signal par nombre fini de valeurs (de nombres)
 - quantification

Codage numérique de l'information



- Discrétiser le signal dans le temps
 - échantillonnage
- Représenter le signal par nombre fini de valeurs (de nombres)
 - quantification

Codage numérique de l'information



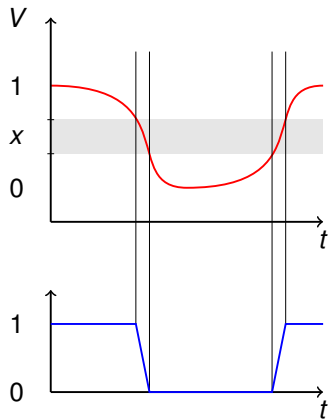
- Discrétiser le signal dans le temps
 - échantillonnage
- Représenter le signal par nombre fini de valeurs (de nombres)
 - quantification



Codage numérique de l'information : intérêts

- Possibilité de reproduire sans perte et de façon illimitée l'information
- Indépendance du support utilisé
 - Câble électrique
 - Fibre optique
 - CDROM, disque dur...
- Possibilité d'organiser le traitement de l'information dans le temps

Codage binaire



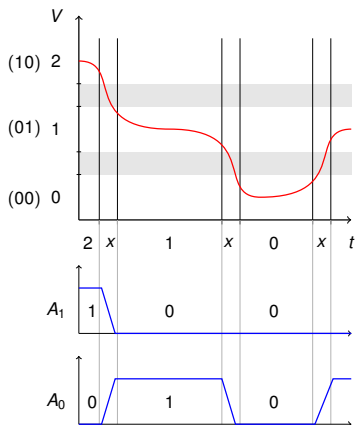
■ Interprétations logiques multiples

- 0/1
- vrai/faux

■ Support électrique très simple

- Codage utilisant deux valeurs

Codage binaire : travailler en base 2



Comment représenter plus de deux niveaux en binaire ?

- Utiliser plusieurs fils
- Les transmettre à tour de rôle

Chaque élément est un **bit**

- **binary digit**

On peut représenter tous les nombres en base 2

■ Convention

- 2 niveaux de tension (0/5V ou -12/12V ou ...)
- 2 niveaux de courant électrique
- Absence/présence de lumière sur une fibre
- ...

■ Interprétations

- Vrai/Faux pour des variables de commande ou status
 - Contrôle
- valeur des nombres
 - Calcul

Plan

Traitement numérique de l'information

Signal électrique binaire

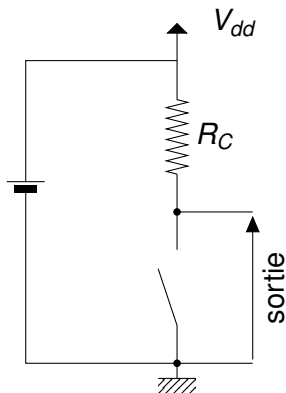
Logique Booléenne

Représentation des nombres

Opérateurs Arithmétiques

Notion de temps de propagation

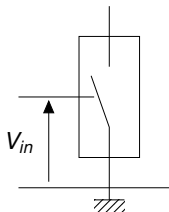
Génération d'un signal électrique binaire



- Interrupteur fermé
 - $\rightarrow 0V$ en sortie
- Interrupteur ouvert
 - $\rightarrow V_{dd}$ en sortie

tension	niveau logique
$0V$	0
V_{dd}	1

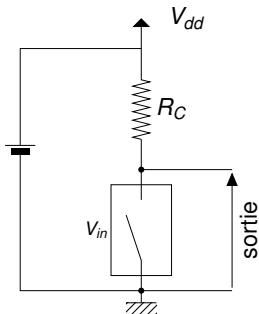
Génération d'un signal électrique binaire



- Interrupteur commandé :
 - Si $V_{in} < V_{ref}$ alors l'interrupteur est ouvert
 - Si $V_{in} > V_{ref}$ alors l'interrupteur est fermé
- Cet interrupteur peut être :
 - Un relais électromagnétique
 - Un tube à vide
 - Un transistor

Opérateur de traitement binaire

Fonction Non

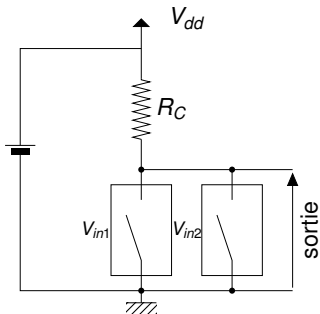


V_{in}	V_s	I_n	Sortie
$< V_{ref}$	V_{dd}	0	1
$> V_{ref}$	0V	1	0

- La fonction Non
- La sortie vaut 0 ssi l'entrée vaut 1

Opérateur de traitement binaire :

Fonction Non-Ou

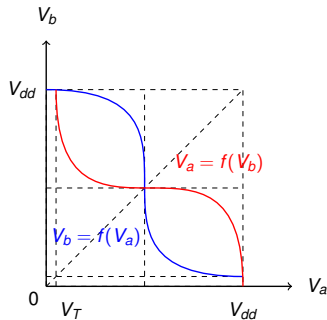
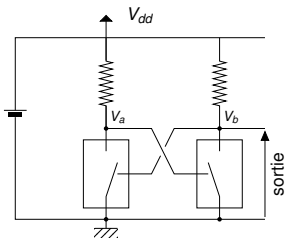


V_{in1}	V_{in2}	V_s	I_{n1}	I_{n1}	Sortie
$< V_{ref}$	$< V_{ref}$	V_{dd}	0	0	1
$< V_{ref}$	$> V_{ref}$	$0V$	0	1	0
$> V_{ref}$	$< V_{ref}$	$0V$	1	0	0
$> V_{ref}$	$> V_{ref}$	$0V$	1	1	0

- Fonction Non-Ou
- la sortie vaut 0 si l'une des entrées vaut 1

Opérateur de traitement binaire :

Fonction de mémorisation



- Le couple (V_a, V_b) possède deux états stables :
 - (V_{dd}, V_{min}) ou (V_{min}, V_{dd})

De l'opérateur de traitement binaire au microprocesseur

- Assemblage simples :
 - Portes logiques
- Assemblage en opérateurs
 - Arithmétique, contrôle ...
- Circuits électroniques exécutant des fonctions complexes
 - Microprocesseur
 - ASIC¹ (Circuits spécifiques à une application)
 - Circuits logiques programmables

1. *Application Specific Integrated Circuit*

Plan

Traitement numérique de l'information

Signal électrique binaire

Logique Booléenne

Représentation des nombres

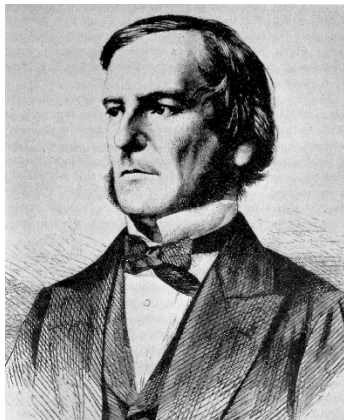
Opérateurs Arithmétiques

Notion de temps de propagation

Algèbre de Boole

Formalisme de la logique

On le doit à George Boole



Crédits image : wikipedia (http://fr.wikipedia.org/wiki/George_Boole)

Variables et fonctions logiques

Variables logiques

- Une variable logique est un élément qui appartient à l'ensemble $E = \{0, 1\}$
- Ne possède que deux états possibles : 0 ou 1

Fonctions logiques

- Fonction d'une ou plusieurs variables logiques.

$$\begin{cases} E \times E \dots \times E \rightarrow E \\ e_0, e_1, \dots, e_n \rightarrow s = F(e_0, e_1, \dots, e_n) \end{cases}$$

Fonctions logiques

Deux catégories

Fonctions combinatoires

La sortie ne dépend que de l'état actuel des entrées

$$\forall t, s(t) = F(e_0(t), e_1(t), \dots, e_n(t))$$

Fonctions séquentielles

La sortie dépend de l'état actuel des entrées et de leur passé

$$s(t) = F(e_0(t), e_1(t), \dots, e_n(t), e_0(t - t_1), e_1(t - t_1) \dots)$$

Fonctions logiques

Représentations

Plusieurs représentations possibles :

Table de vérité : En donnant toutes les valeurs possibles pour toutes les entrées possibles.

Analytique : En donnant l'équation analytique

Graphique : En utilisant les symboles de fonctions de base

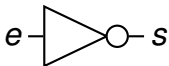
HDL : Langage "informatique" de description du matériel
(Hardware Description Language)

Fonctions élémentaires

L'inverseur (Not)

- La sortie est le complément de l'entrée
- La sortie vaut 1 si et seulement si l'entrée vaut 0

Symbole



Équation

$$s = \bar{e}$$

Table de vérité

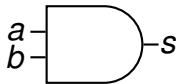
e	s
0	1
1	0

Fonctions élémentaires

Le "et" (And)

- La sortie vaut 1 si et seulement si les deux entrées valent 1
- Si l'une des entrées vaut 0 alors la sortie vaut 0

Symbole



Équation

$$s = a \cdot b$$

Table de vérité

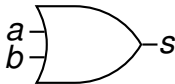
<i>a</i>	<i>b</i>	<i>s</i>
0	0	0
0	1	0
1	0	0
1	1	1

Fonctions élémentaires

Le "ou" (Or)

- Si l'une des entrées vaut 1 alors la sortie vaut 1
- La sortie vaut 0 si et seulement si les deux entrées valent 0

Symbole



Équation

$$s = a + b$$

Table de vérité

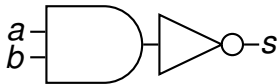
<i>a</i>	<i>b</i>	<i>s</i>
0	0	0
0	1	1
1	0	1
1	1	1

Fonctions de base

Le “non et” (Nand)

- La fonction complémentaire du And
- La sortie vaut 1 si l'une des entrées est à 0

Symbole



Équation

$$s = \overline{a \cdot b}$$

Table de vérité

<i>a</i>	<i>b</i>	<i>s</i>
0	0	1
0	1	1
1	0	1
1	1	0

Fonctions de base

Le “non et” (Nand)

- La fonction complémentaire du And
- La sortie vaut 1 si l'une des entrées est à 0

Symbole



Équation

$$s = \overline{a \cdot b}$$

Table de vérité

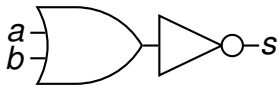
<i>a</i>	<i>b</i>	<i>s</i>
0	0	1
0	1	1
1	0	1
1	1	0

Fonctions de base

Le “non ou” (Nor)

- La fonction complémentaire du Or
- La sortie vaut 0 si l'une des entrées est à 1

Symbole



Équation

$$s = \overline{a + b}$$

Table de vérité

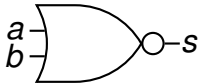
<i>a</i>	<i>b</i>	<i>s</i>
0	0	1
0	1	0
1	0	0
1	1	0

Fonctions de base

Le “non ou” (Nor)

- La fonction complémentaire du Or
- La sortie vaut 0 si l'une des entrées est à 1

Symbole



Équation

$$s = \overline{a + b}$$

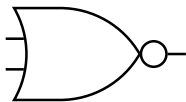
Table de vérité

<i>a</i>	<i>b</i>	<i>s</i>
0	0	1
0	1	0
1	0	0
1	1	0

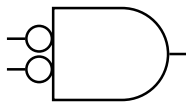
Équivalence And/Or

Théorème de De Morgan

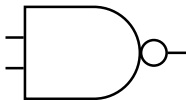
$$\overline{a + b} = \bar{a} \cdot \bar{b}$$



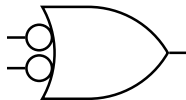
≡



$$\overline{a \cdot b} = \bar{a} + \bar{b}$$



≡



Exercice

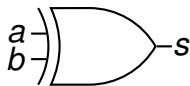
- Comment réaliser une porte à deux entrées, dont la sortie vaut '1' si et seulement si les deux entrées sont différentes ?
- Comment réaliser une porte à deux entrées, dont la sortie vaut '1' si et seulement si les deux entrées sont identiques ?

Fonctions de base

Le "Ou exclusif" (Xor)

- La sortie vaut 1 si une seule entrée est à 1
- La sortie vaut 1 si les deux entrées sont différentes

Symbole



Équation

$$s = a \oplus b$$
$$s = a \cdot \bar{b} + \bar{a} \cdot b$$

Table de vérité

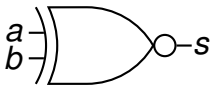
a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

Fonctions de base

Le "Non Ou exclusif" (Xnor)

- La sortie vaut 1 si les deux entrées sont identiques
- C'est la fonction complémentaire du xor
- C'est la porte égalité

Symbole



Équation

$$s = \overline{a \oplus b}$$
$$s = a \cdot b + \bar{a} \cdot \bar{b}$$

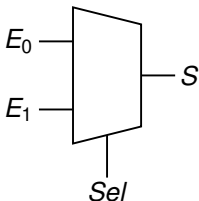
Table de vérité

a	b	s
0	0	1
0	1	0
1	0	0
1	1	1

Exercices ?

Multiplexeur

- On veut réaliser une fonction d'aiguillage $2 \rightarrow 1$.
- Cette porte permet de sélectionner l'une des deux entrées en fonction d'une troisième entrée de sélection



Disjunctive Normal Form

Elle correspond à une somme de produits logiques : $F = \Sigma\Pi(e_i)$, où e_i représente une variable ou son complément. Exemple :

$$F_1(X, Y, Z) = X \cdot Y + X \cdot \bar{Z} + \bar{X} \cdot \bar{Y} \cdot Z$$

Si chacun des produits contient toutes les variables d'entrée sous une forme directe ou complémentée, alors la forme est appelée « **première forme canonique** » ou « **forme canonique disjonctive** ». Chacun des produits est alors appelé **minterme**. Exemple de forme canonique disjonctive :

$$F_2(X, Y, Z) = \bar{X} \cdot \bar{Y} \cdot Z + \bar{X} \cdot Y \cdot Z + X \cdot \bar{Y} \cdot \bar{Z}$$

Conjunctive Normal Form

Elle fait référence à un produit de sommes logiques : $F = \Pi\Sigma(e_i)$. Voici un exemple :

$$F_3(X, Y, Z) = (X + Y) \cdot (\bar{X} + Z) \cdot (\bar{X} + Y + \bar{Z})$$

Si chacune des sommes contient toutes les variables d'entrée sous une forme directe ou complémentée, alors la forme est appelée « **deuxième forme canonique** » ou « **forme canonique conjonctive** ». Chacune des sommes est alors appelée **maxterme**. Exemple de forme canonique conjonctive :

$$F_4(X, Y, Z) = (X + Y + Z) \cdot (\bar{X} + \bar{Y} + Z) \cdot (\bar{X} + Y + \bar{Z})$$

Example of DNF

<i>A</i>	<i>B</i>	<i>C</i>	$H(A, B, C)$	État	Minterme
0	0	0	1	0	$\overline{A} \cdot \overline{B} \cdot \overline{C}$
0	0	1	1	1	$\overline{A} \cdot \overline{B} \cdot C$
0	1	0	0	2	$\overline{A} \cdot B \cdot \overline{C}$
0	1	1	1	3	$\overline{A} \cdot B \cdot C$
1	0	0	0	4	$A \cdot \overline{B} \cdot \overline{C}$
1	0	1	1	5	$A \cdot \overline{B} \cdot C$
1	1	0	0	6	$A \cdot B \cdot \overline{C}$
1	1	1	0	7	$A \cdot B \cdot C$

$$H(A, B, C) = \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C$$

Example of CNF

A	B	C	$H(A, B, C)$	Etat	Maxterme
0	0	0	1	0	$A + B + C$
0	0	1	1	1	$A + B + \bar{C}$
0	1	0	0	2	$A + \bar{B} + C$
0	1	1	1	3	$A + \bar{B} + \bar{C}$
1	0	0	0	4	$\bar{A} + B + C$
1	0	1	1	5	$\bar{A} + B + \bar{C}$
1	1	0	0	6	$\bar{A} + \bar{B} + C$
1	1	1	0	7	$\bar{A} + \bar{B} + \bar{C}$

$$H(A, B, C) = (A + \bar{B} + C) \cdot (\bar{A} + B + C) \cdot (\bar{A} + \bar{B} + C) \cdot (\bar{A} + \bar{B} + \bar{C})$$

Boolean algebra properties

Complémentarité :	$a + \bar{a} = 1,$	$a \cdot \bar{a} = 0,$	$\bar{\bar{a}} = a$
Idempotence :	$a + a + a + \dots = a,$		$a \cdot a \cdot a \dots = a$
Éléments neutres :	$a + 0 = a,$		$a \cdot 1 = a$
Éléments absorbants :	$a + 1 = 1,$		$a \cdot 0 = 0$
Commutativité :	$a + b = b + a,$		$a \cdot b = b \cdot a$
Associativité :	$(a + b) + c = a + (b + c) = a + b + c,$	$(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c$	
Distributivité :	$(a + b) \cdot c = (a \cdot c) + (b \cdot c),$	$(a \cdot b) + c = (a + c) \cdot (b + c)$	
Théorème d'absorption (1) :	$a + (a \cdot b) = a,$		$a \cdot (a + b) = a$
Théorème d'absorption (2) :	$a \cdot \bar{b} + b = a + b,$		$(a + \bar{b}) \cdot b = a \cdot b$
Théorème d'adjacence :	$(a + \bar{b}) \cdot (a + b) = a,$		$a \cdot \bar{b} + a \cdot b = a$

Remarque : Deux termes sont dits **adjacents** logiquement s'ils ne diffèrent que par une variable.

Secondary results

Théorème de De Morgan :

$$\overline{a + b} = \bar{a} \cdot \bar{b}, \quad \overline{a \cdot b} = \bar{a} + \bar{b}$$

Premier théorème d'expansion :

$$F(e_0, e_1, \dots, e_i, \dots, e_{n-1}) = e_i \cdot F(e_0, e_1, \dots, 1, \dots, e_{n-1}) + \bar{e}_i \cdot F(e_0, e_1, \dots, 1, \dots, e_{n-1})$$

Second théorème d'expansion :

$$F(e_0, e_1, \dots, e_i, \dots, e_{n-1}) = [e_i + F(e_0, e_1, \dots, 0, \dots, e_{n-1})] \cdot [\bar{e}_i + F(e_0, e_1, \dots, 1, \dots, e_{n-1})]$$

Simplifications 1/3

Regroupement des termes et mises en facteur

$$\begin{aligned} Z &= \bar{a} \cdot \bar{c} \cdot d + \bar{a} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot b \cdot c \cdot \bar{d} = \bar{a} \cdot \bar{c} \cdot (d + \bar{d}) + \bar{a} \cdot b \cdot c \cdot \bar{d} \\ &= \bar{a} \cdot \bar{c} + \bar{a} \cdot b \cdot c \cdot \bar{d} = \bar{a} \cdot (\bar{c} + c \cdot b \cdot \bar{d}) = \bar{a} \cdot (\bar{c} + b \cdot \bar{d}) \end{aligned}$$

Nous avons successivement utilisé une mise en facteur, la complémentarité, une deuxième mise en facteur et enfin le théorème d'absorption.

Simplifications 2/3

Réplication de termes existants

$$\begin{aligned} Z &= \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c} + a \cdot b \cdot c \\ &= \bar{a} \cdot b \cdot c + a \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot c + a \cdot b \cdot \bar{c} + a \cdot b \cdot c \\ &= (\bar{a} + a) \cdot b \cdot c + (\bar{b} + b) \cdot a \cdot c + (\bar{c} + c) \cdot a \cdot b \\ &= b \cdot c + a \cdot c + a \cdot b \end{aligned}$$

La réplication du terme $a \cdot b \cdot c$ permet de simplifier chacun des trois premiers termes en utilisant une mise en facteur et la complémentarité.

Simplifications 3/3

Suppression de termes superflus

$$\begin{aligned} Z &= \bar{a} \cdot \bar{b} + b \cdot \bar{c} + \bar{a} \cdot \bar{c} = \bar{a} \cdot \bar{b} + b \cdot \bar{c} + \bar{a} \cdot \bar{c} \cdot (b + \bar{b}) \\ &= \bar{a} \cdot \bar{b} + \bar{a} \cdot \bar{b} \cdot \bar{c} + b \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} = \bar{a} \cdot \bar{b} \cdot (1 + \bar{c}) + b \cdot \bar{c} \cdot (1 + \bar{a}) \\ &= \bar{a} \cdot \bar{b} + b \cdot \bar{c} \end{aligned}$$

Nous avons ici réintroduit la variable b dans le troisième terme par l'intermédiaire de la propriété de complémentarité, nous avons ensuite utilisé la propriété d'absorption pour simplifier les produits.

Karnaugh maps

<i>A</i> \ <i>BC</i>	00	01	11	10
0	1	1	1	0
1	1	0	1	1

<i>A</i> \ <i>BC</i>	00	01	11	10
0	1	1	1	0
1	1	0	1	1

$$\bar{b} \cdot \bar{c} + \bar{a} \cdot c + a \cdot b$$

<i>A</i> \ <i>BC</i>	00	01	11	10
0	1	1	1	0
1	1	0	1	1

$$\bar{a} \cdot \bar{b} + b \cdot c + a \cdot \bar{c}$$