



IP PARIS

Introduction à la conception de systèmes sur puce

Introduction

Tarik Graba

tarik.graba@telecom-paris.fr

2024-2025



Systèmes sur puce / Systèmes embarqués

Un système embarqué

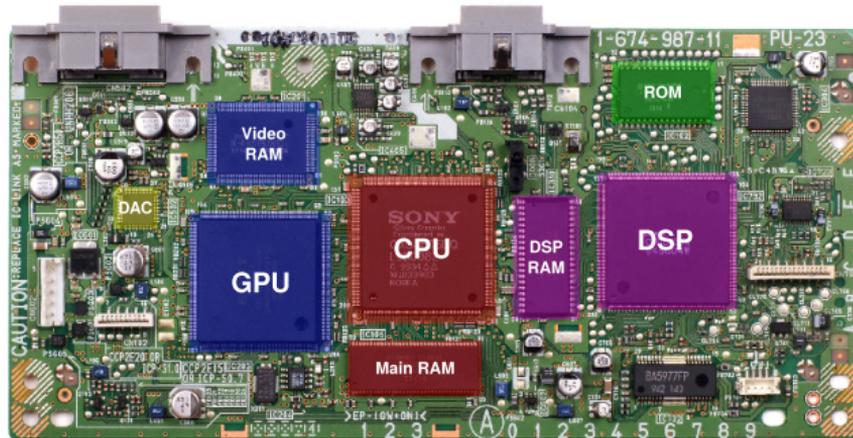
■ Qu'est-ce qu'un système embarqué ?

C'est un système programmable **spécialisé** construit autour d'un **processeur** pour accomplir des tâches spécifiques.

■ Pourquoi spécialiser ?

- Garantir les performances
 - temps réel, vitesse, consommation
- Réduire le coût

Un système embarqué (exemple 1 : Les années 90!)

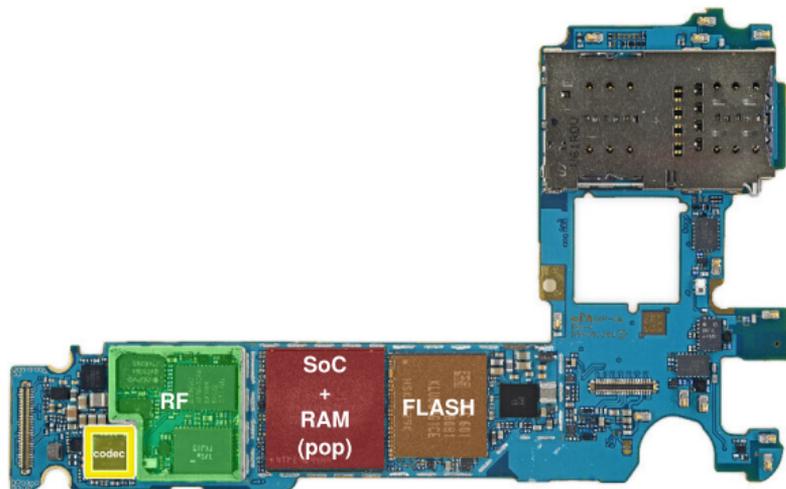


source https://en.wikipedia.org/wiki/PlayStation_technical_specifications

Carte mère PS1

- Un ensemble de composants :
- un processeur généraliste,
 - des processeurs dédiés,
 - des mémoires,
 - des interfaces internes et externes...

Un système embarqué (exemple 2 : Les années 20xx !)



source <https://www.ifixit.com/Teardown/Samsung+Galaxy+S7+Teardown/56686>

Carte mère Galaxy S7

Un ensemble de composants :

- Moins de composants
- Moins cher
- Plus performant

Un système sur puce (SoC)

■ Qu'est-ce qu'un système sur puce ?

C'est un système programmable **intégré et spécialisé** construit autour d'un **processeur** pour accomplir des tâches spécifiques.

■ Pourquoi spécialiser ?

- Garantir les performances
 - temps réel, vitesse, consommation
- Réduire le coût

■ Pourquoi intégrer ?

- Augmenter les performances
- Réduire les coûts au niveau système

Où trouve-t-on des SoC ?

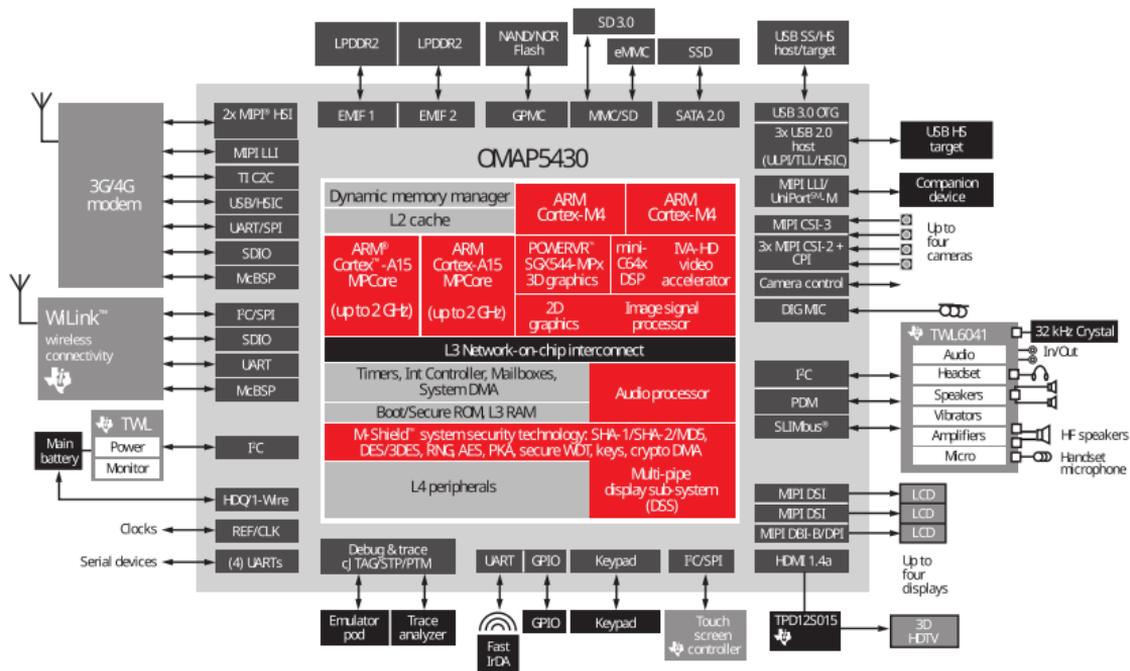
Un peu partout :

- Dans les smartphones/Tablettes
- Dans les TV et Set-top boxes
- Dans les appareils photos et caméras
- Dans les équipements réseau, les imprimantes...
- Dans les cartes à puce...

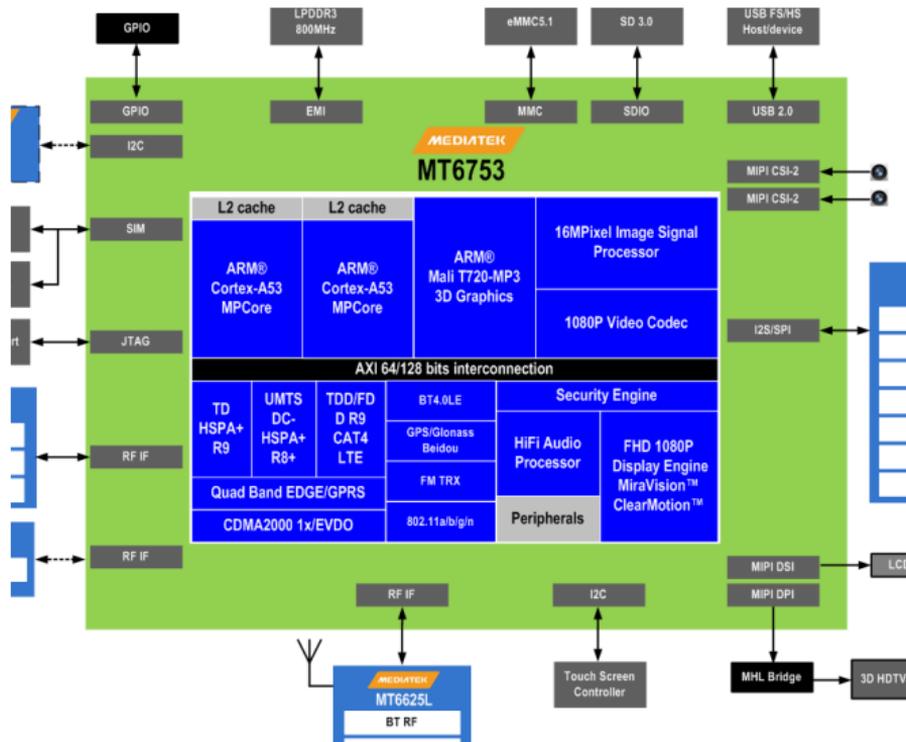
Un système sur puce (importance du marché)

- Identifier son marché et ses besoins.
- Proposer un produit qui intègre les bonnes fonctionnalités.

Exemple, les SoC pour la téléphonie mobile :



OMAP5 2011



MT6753 2015

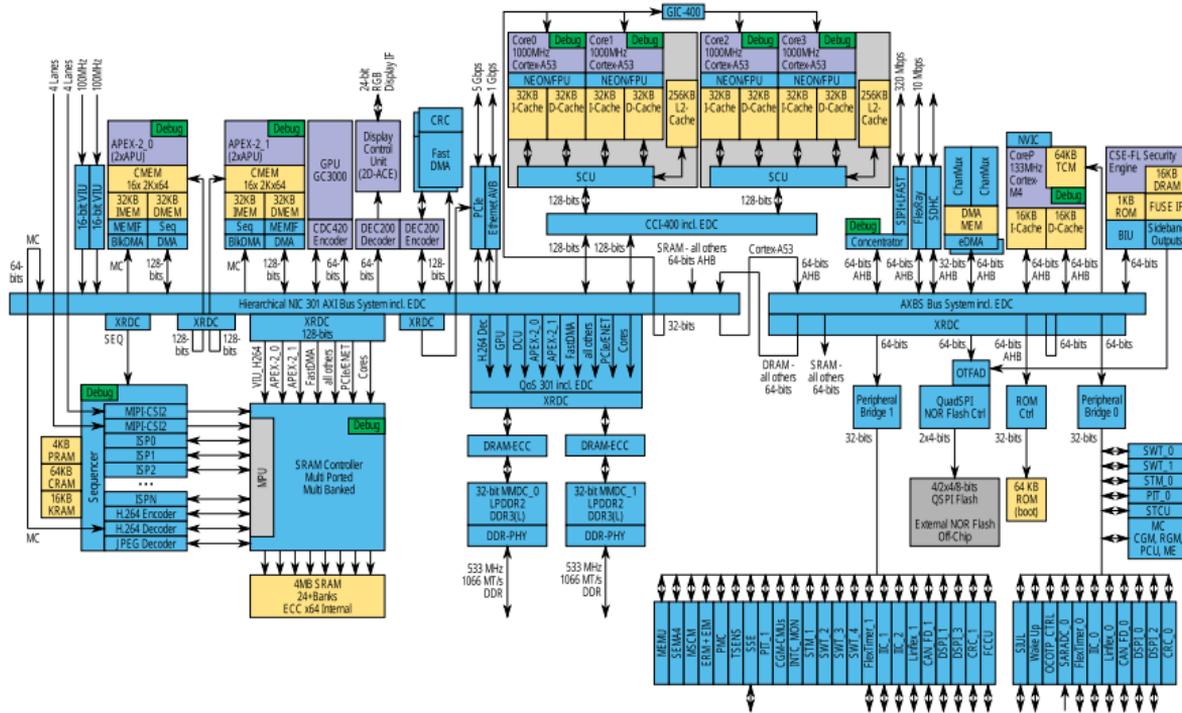
Familles de systèmes sur puce



Familles de systèmes sur puce

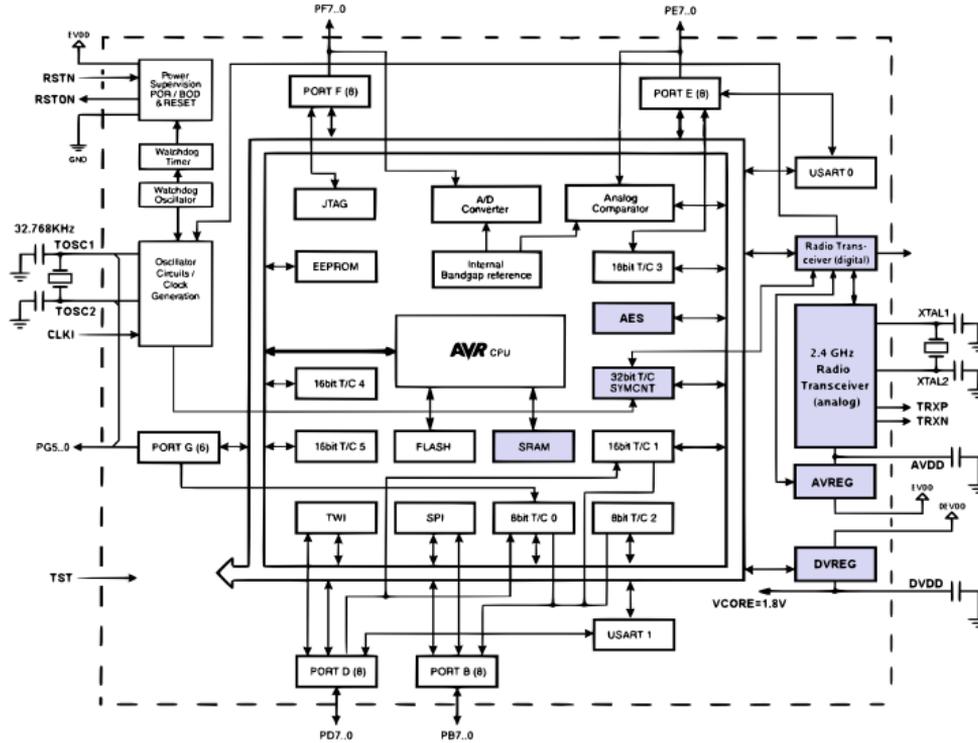
Au-delà des SoC pour les smartphones.

Vision (et IA)



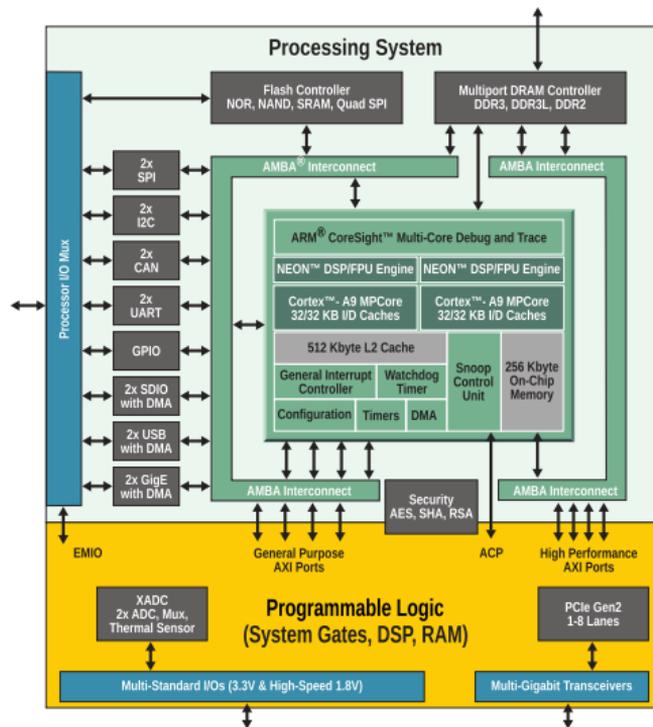
NXP S32V2

Microcontrôleur et radio (composante analogique)



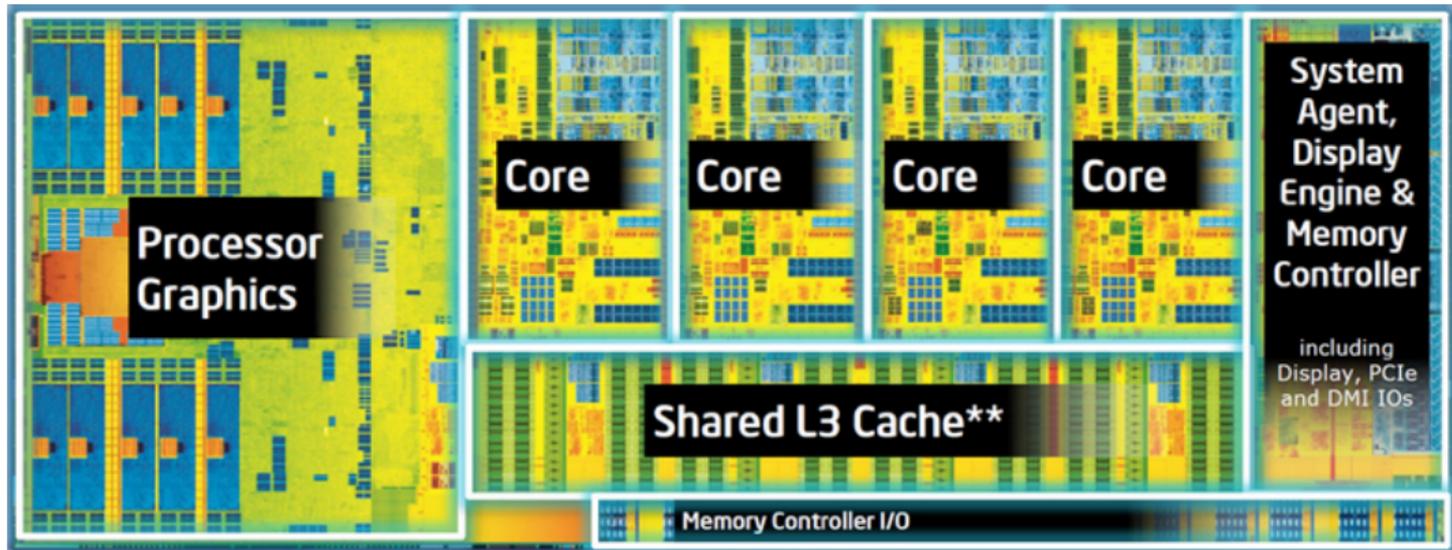
ATmega 128RFA1

Dans certains FPGAs



Zynq 7000 de Xilinx

Les processeurs de PC



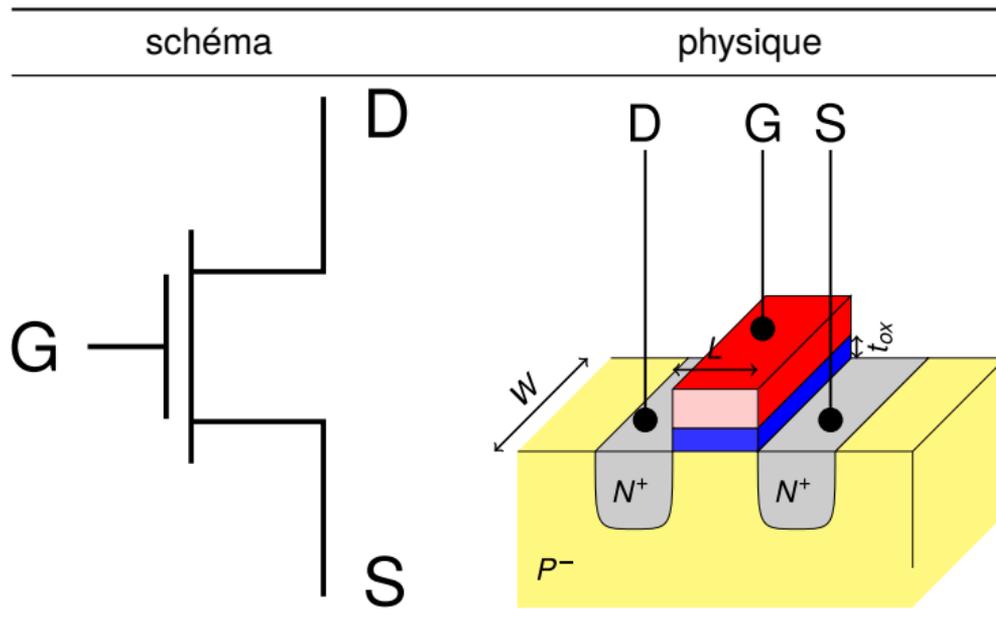
Intel Core i7

Résumons (tous des SoC)

- Processeurs généralistes
 - PC et serveur...
 - Alimentation garantie (?)
 - Performance et coût élevés
 - multi-thread, mémoires externes
 - évolutif
- SoCs embarqués
 - Smartphones, Systèmes automobiles...
 - Sur batteries
 - Performance et coût raisonnables
 - multi-thread contrôlé, mémoire externe
 - figé
- Microcontrôleurs
 - Systèmes embarqué temps réel
 - très faible consommations
 - coût très faible et réactivité
 - temps réel, pas de mémoires externe

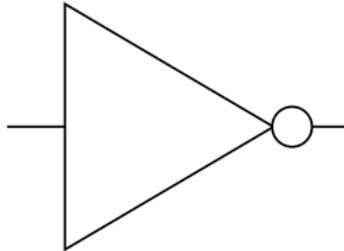
Considérations technologiques

Transistor MOS

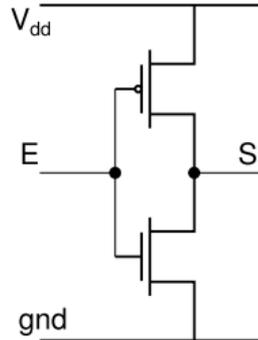


La logique CMOS

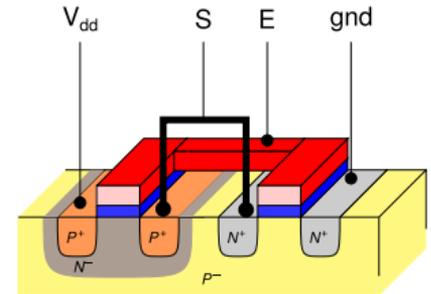
symbole



schéma

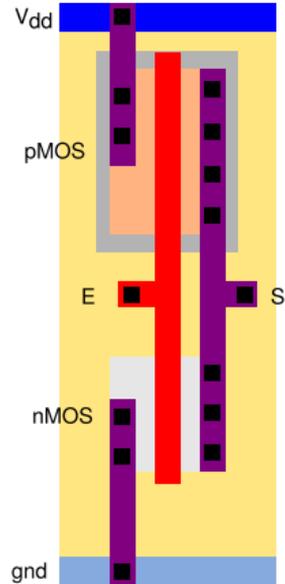


physique



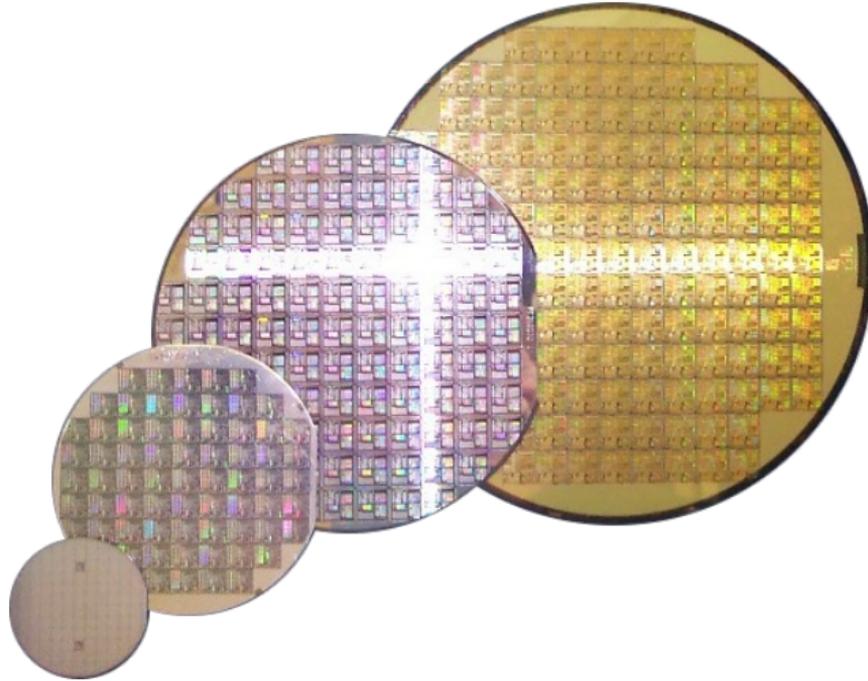
Fabrication

- Dessins de masques (GDSII : *Graphic Data System*)



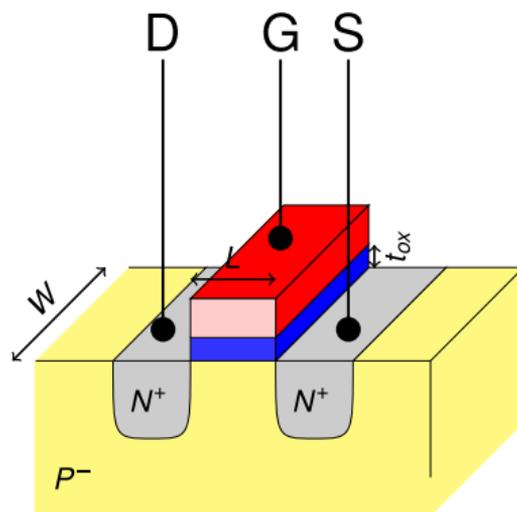
Layout d'un inverseur

■ Les tranches de silicium (*Wafers*)



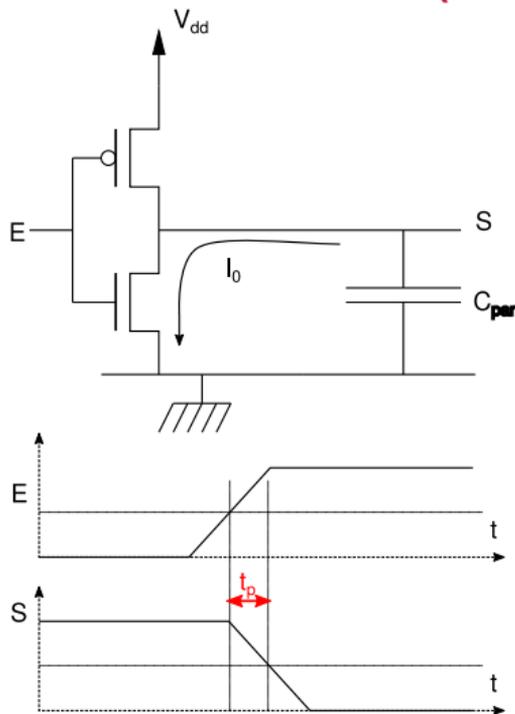
Wafers

Dimensions du transistor de base



nMOS : dimensions

Performances (temps)



Temps de propagation d'une porte CMOS

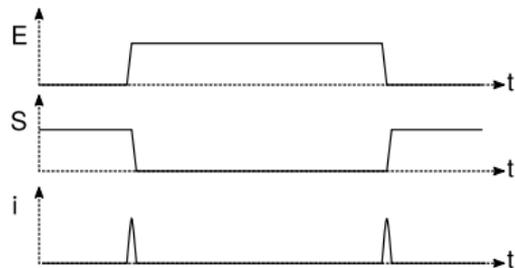
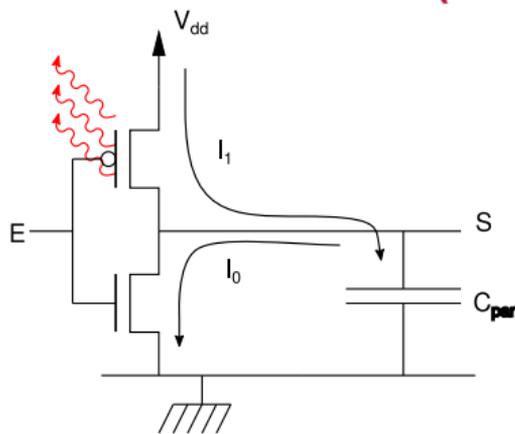
- Temps de propagation dans une porte CMOS
 - Le courant de décharge

$$I_0 \approx K_n \cdot (V_{dd} - V_{tn})^2$$

- Le temps de propagation

$$t_p = C_{par} \frac{\Delta V}{I_0} = C_{par} \frac{V_{dd}}{K_n \cdot (V_{dd} - V_{tn})^2}$$

Performances (consommation)



Consommation d'une porte CMOS

- Consommation dans une porte CMOS liée au changement d'état
 - Energie

$$E = \int_0^{V_{dd}} C_{par} V_S dV_S = \frac{1}{2} C_{par} V_{dd}^2$$

- Puissance

$$P = \alpha \cdot C \cdot f \cdot V_{dd}^2$$

Complexité et coût

Pour un nœud technologique :

- complexité → nombre de transistors → coût
 - circuit plus grand → moins par wafer
 - circuit plus grand → rendement moins important

Évolution des technologies

Loi de Moore

- Doublement de la complexité des circuits intégrés tous les deux ans, à coût constant.
- Doublement du nombre de transistors intégrés dans un circuit tous les deux ans, à coût constant.

Implication de la loi de Moore sur les performances

Pour les technologies planaires, en passant à un nœud β plus petit on essaye de :

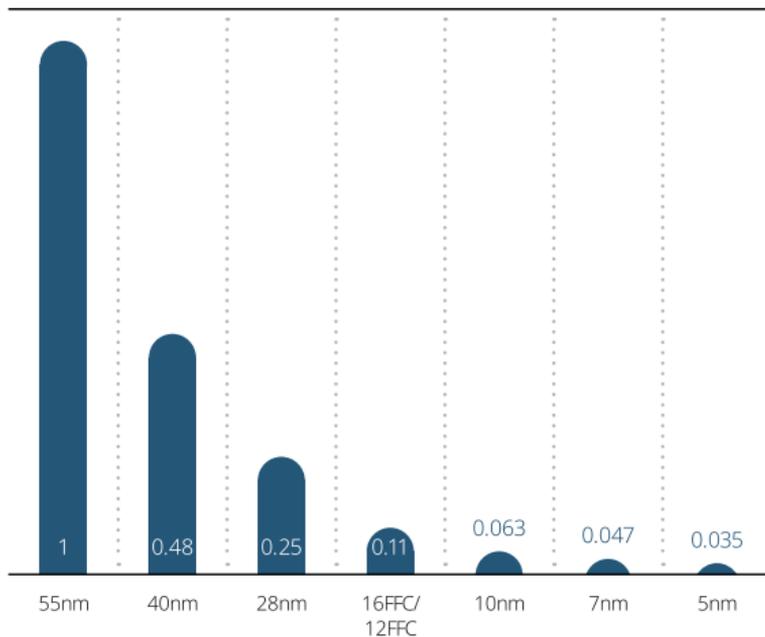
- Réduire les dimensions du transistor
 - $L \rightarrow L/\beta$
 - $W \rightarrow W/\beta$
 - La surface est réduite de β^2
 - $T_{ox} \rightarrow T_{ox}/\beta$
- Réduire la tension d'alimentation
 - $V_{dd} \rightarrow V_{dd}/\beta$
- Réduire les seuils de transistors
 - $V_T \rightarrow V_T/\beta$

On obtient pour les performances :

- Une réduction du temps de propagation
 - $t_p \rightarrow t_p/\beta$
- Une réduction de l'énergie de transition
 - $E \rightarrow E/\beta^3$

Chip Die Size Cross-Technology Comparison

Die size is shrinking as line width shrinks

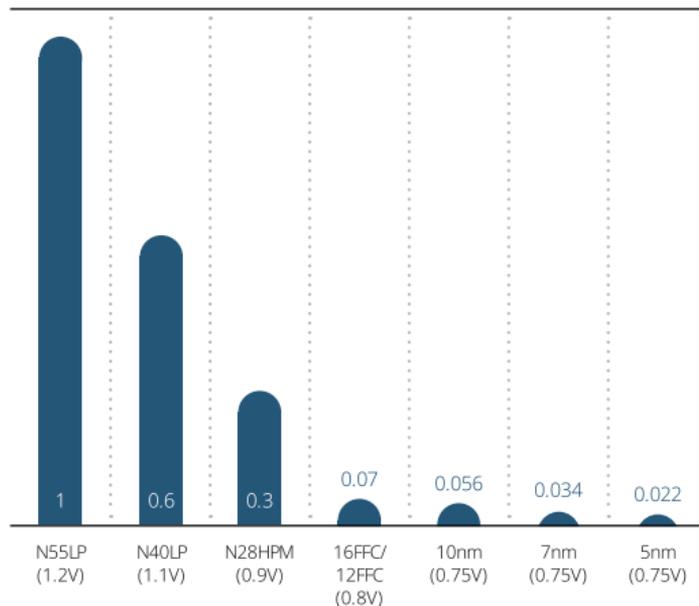


TSMC : Évolution de la surface

Consommation

Chip Total Power Consumption Cross-Technology Comparison

More power is saved as line width shrinks

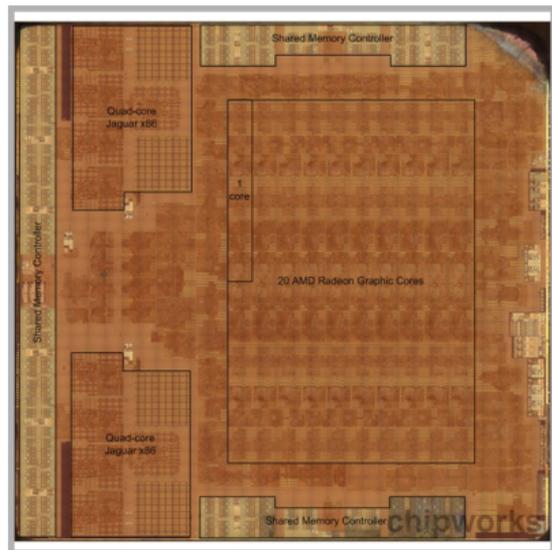


TSMC : Évolution de la consommation

Flot de développement des SoC

Comment passer d'un algorithme à un circuit

```
// 60 frames/s display
Thread1 {
  every(16.66ms)
  display(image)
}
// scene update
Thread2 {
  if(ViewChanged)
    redraw(image)
}
// player interaction
Thread3 {
  wait(player1)
  move(rabbit0)
}
```



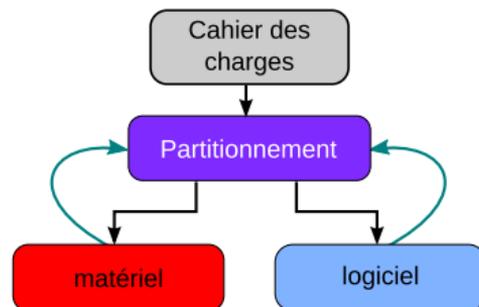
APU PS4 (18 × 19 mm²) TSMC 28 nm,
<http://www.chipworks.com>

Comment passer d'une fonctionnalité (algorithme) à un circuit ?

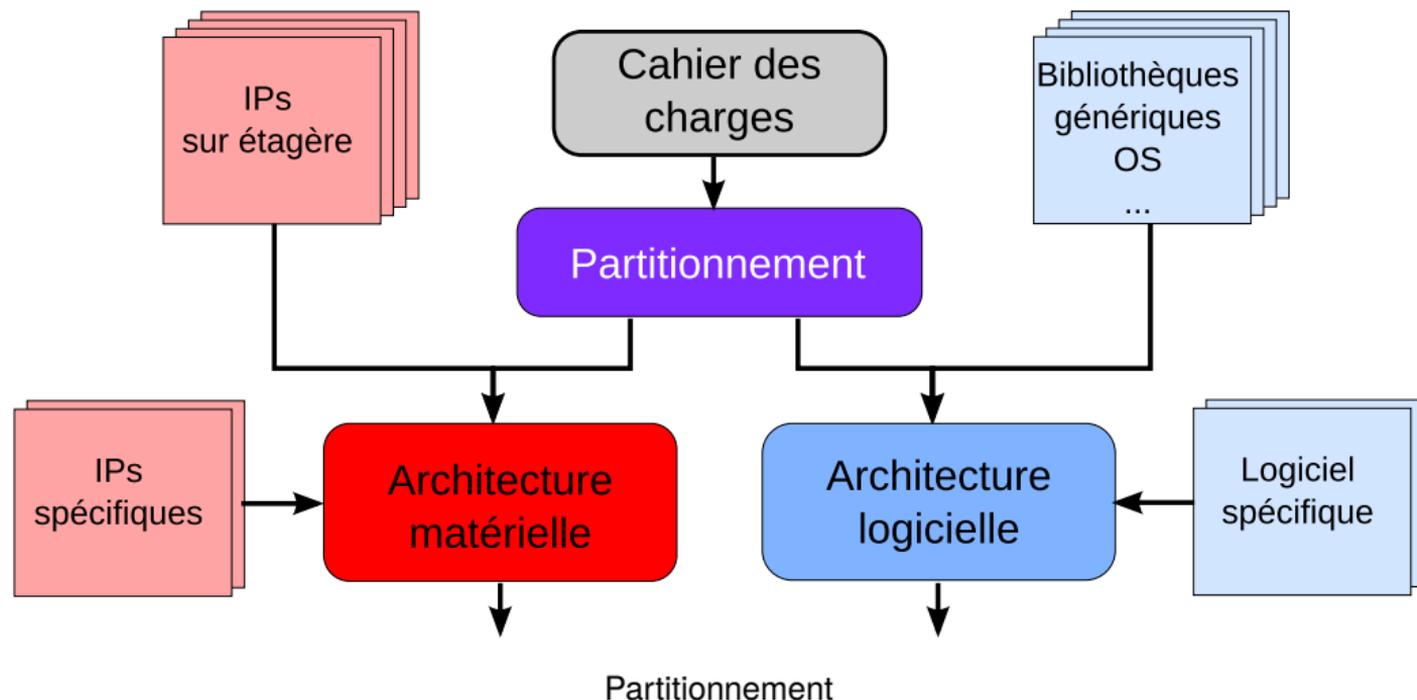
Partitionnement logiciel/matériel

À partir d'un cahier des charges de l'application, définir :

- ce qui doit être fait par le logiciel,
- ce qui doit être fait par le matériel.



Développement conjoint logiciel/matériel



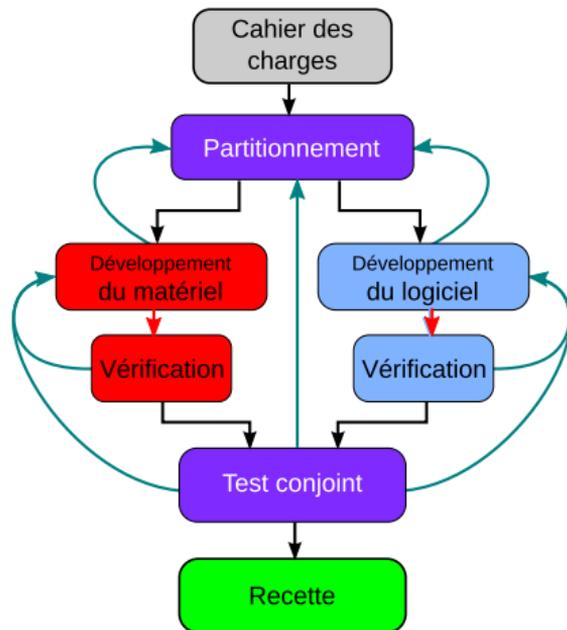
Des blocs matériels réutilisables !

- Réutiliser des blocs matériels existants disponibles sous la forme de :
 - sources RTL (ou sources chiffrées)
 - blocs pré-synthétisés
 - blocs placés-routés
- Ces blocs sont appelés des *IP*
 - IP blocs : *Intellectual Property blocks*

Validation conjointe

Il faudra prévoir des itérations de test et de validation :

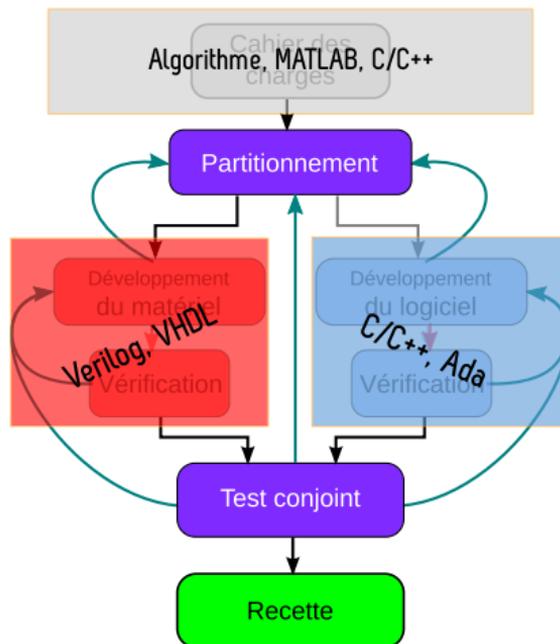
- pour le logiciel
- pour le matériel
- conjoint



Référentiel de validation

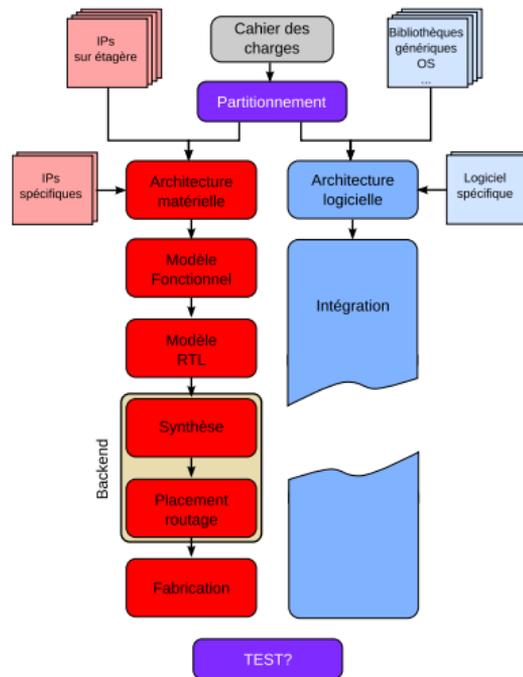
Il est souvent complexe d'avoir un référentiel commun pour la validation :

- Les équipes sont souvent différentes.
- Les philosophies sont différentes.
- Les langages/outils et méthodes sont différents.



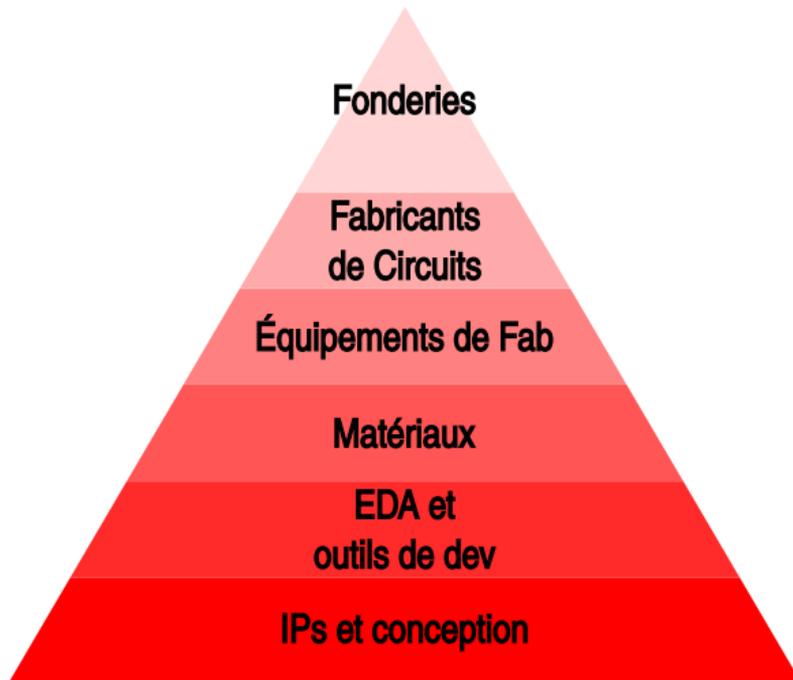
Échelle de temps

- Les échelles de temps ne sont pas les mêmes.
- Attendre la fabrication d'un circuit pour tester le logiciel dessus, n'est pas raisonnable !



La filière industrielle autour des SoCs

Un Ensemble d'intervenants



La filière industrielle

Les fonderies

- TSMC, Samsung, (Intel)
- UMC, SMIC, (ST), GlobalFoundrie, (Micron)...

Les fabricants de circuits

- IDM (Integrated Device Manufacturer)
 - Intel, Samsung, STM,
 - Micron, Kioxia (Toshiba)



■ Fabless

- Qualcomm, Broadcom, Alwinner
- Xilinx, AMD...
- STM, NXP...

- Architecture et conception des systèmes
- Sous-traitent la fabrication des circuits.

Les fabricants d'équipements

- Équipements de lithographie
 - ASML, Nikon
- Réacteurs, dépôts,...
- Chaîne de fabrication...



Nikon



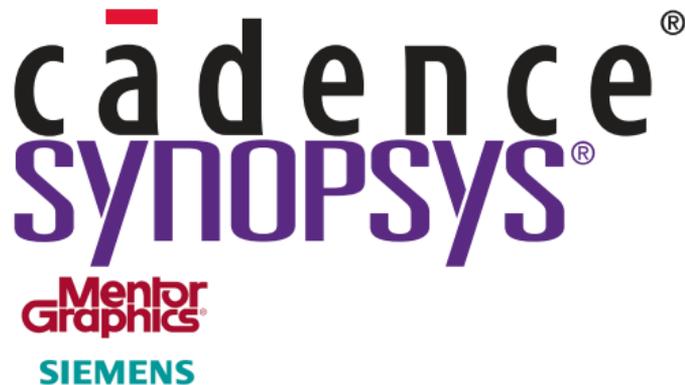
ASML

Les fournisseurs de matériaux

- La chimie...

Les vendeurs d'outils

- 3 grands acteurs
 - Cadence, Synopsys, Siemens EDA (Mentor)



Conception et IPs

- De grand groupes spécialisés
 - ARM, Imagination Tech., Rambus...
- Les vendeurs d'outils
 - Synopsys, Cadence...
- Des start-ups



Organisation de l'UE



Dépôt Git

- <https://gitlab.telecom-paris.fr/se209/2425>

Évaluation individuelle

- TPs notés
 - TPs sur FPGA
 - TPs backend ASIC
- Mini-projet en autonomie