



IP PARIS

5SE05 : Conception des Systèmes sur Puces

Introduction

Tarik Graba

`<tarik.graba@telecom-paris.fr>`

Septembre 2026





Plan

Système sur puce

Introduction

Logiciel et matériel

Flot de développement

Objectifs de l'UE



Plan

Système sur puce

Introduction

Logiciel et matériel

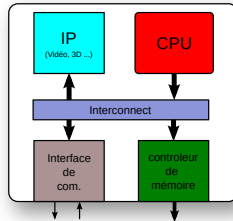
Flot de développement

Objectifs de l'UE

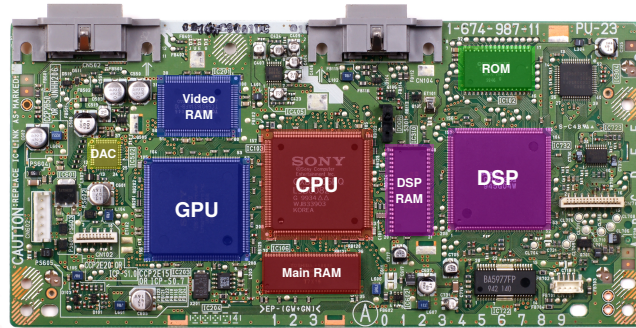
Qu'est-ce qu'un SoC?

SoC

Un SoC (System on Chip/Système sur puce) est un circuit électronique **intégré** qui contient un processeur et des blocs spécifiques pour l'aider à réaliser des tâches particulières.



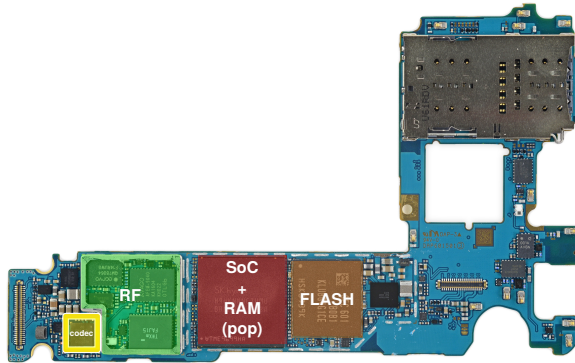
Système embarqué des années 90



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

- Processeur généraliste (32-bits 33MHz)
- Processeur spécialisé (GPU + DSP)
- Mémoires (16Mbits, 1Mbits, 4Mbits), ROM (4Mbits)
- interfaces ...

Système actuel



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

- Processeur (64bits, 2.2GHz) + GPU + DSP + Réseau
- Mémoires (4GBytes),
- Flash (32GBytes)
- interfaces ...



Qu'est-ce qu'un SoC?

Quelle différence avec un PC?

Dans un PC, il y a aussi:

- un processeur;
- de la mémoire;
- des périphériques ...

Aussi les “processeurs” récents sont des SoCs.

Ils intègrent:

- GPU, Interfaces...

Qu'est-ce qu'un SoC?

Quelle différence avec un PC?

La différence principale vient de la spécialisation.

- Un SoC est conçu pour une fonction particulière
 - vidéo, audio, réseau,...
- Un SoC **intègre des modules spécialisés** pour cette fonction
- Un SoC n'intègre que ce qui est nécessaire

Être efficace plutôt qu'universel.



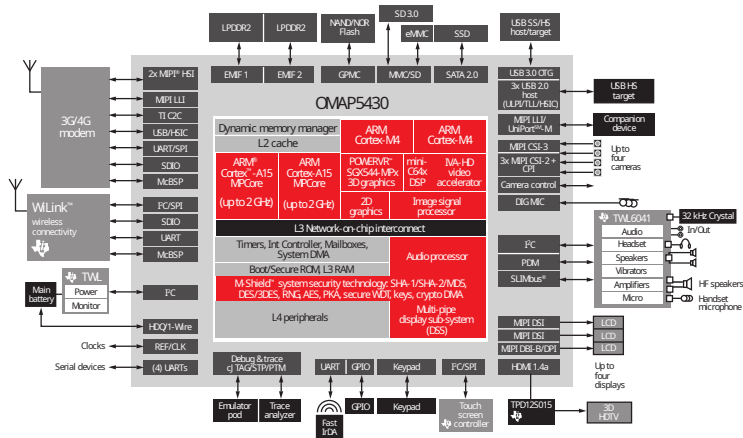
Qu'est-ce qu'un SoC?

Où trouve-t-on des SoC?

- Dans les téléphones/ tablets
- Dans les TV, set-top box
- Dans les appareils photo
- Dans les imprimantes
- Dans les cartes à puce
- ...

Qu'est-ce qu'un SoC?

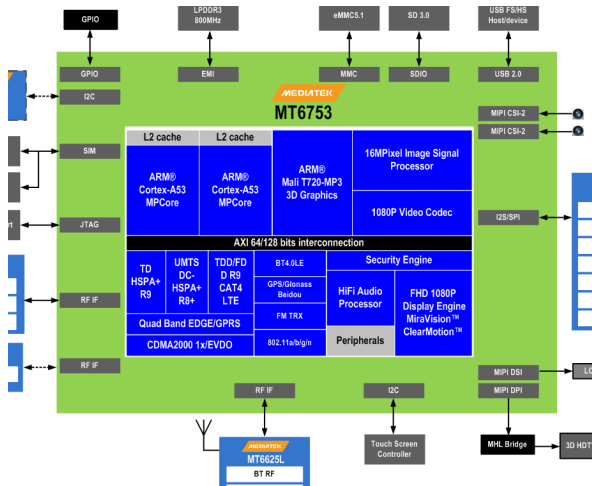
Exemple: OMAP5430 (Texas Instruments 2011)



Source http://focus.ti.com/pdfs/wtbu/OMAP5_2011-7-13.pdf

Qu'est-ce qu'un SoC?

Exemple: MT6753 (Mediatek 2015)



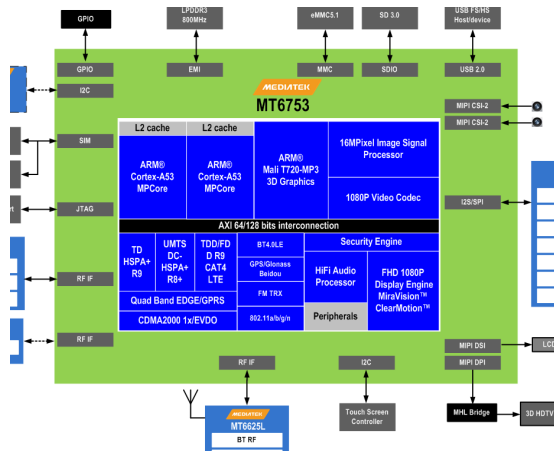
Source <http://www.mediatek.com/en/products/mobile-communications/smartphone1/mt6753/>

Qu'est-ce qu'un SoC?

Exemple: MT6753 (Mediatek 2015)

Application Platform

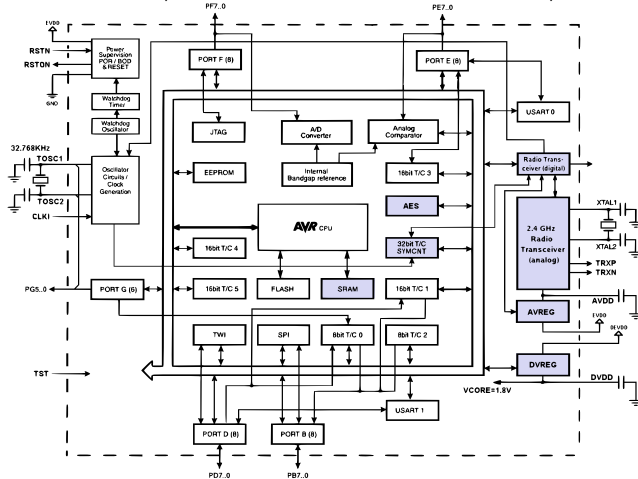
- Processeur **octocœurs** (1.3 GHz)
- 1 DSP, Accélérateur graphique, vidéo
- **Modem GSM/LTE**
- Ctrl DDR, IOs
- ...



Source <http://www.mediatek.com/en/products/mobile-communications/smartphone1/mt6753/>

Exemple: AT32UC3A0128AU (Atmel)

Source <http://www.atmel.com/devices/ATMEGA128RFA1.aspx>

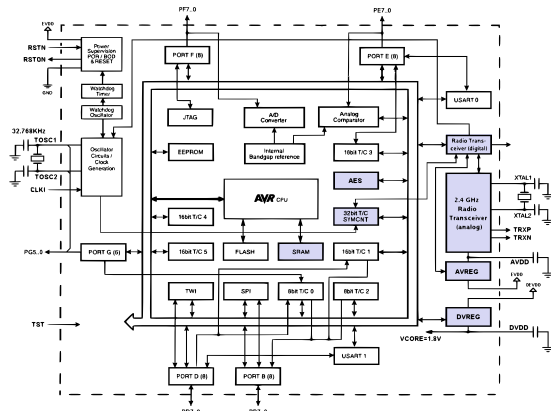


Qu'est-ce qu'un SoC?

Exemple: AT32UC3A0128AU (Atmel)

Interface ZigBee

- 1 microcontrôleur 8bit (16 MHz)
- Transducteur IEEE 802.15.4
- ...

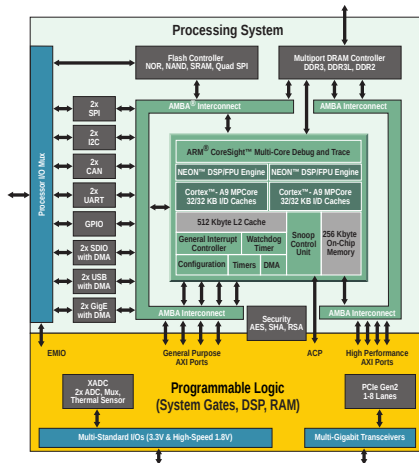


<http://www.atmel.com/devices/ATMEGA128RFA1.aspx>

Qu'est-ce qu'un SoC?

Exemple: Zynq-7000 de Xilinx

Source <http://www.xilinx.com/content/xilinx/en/products/silicon-devices/soc/zynq-7000.html>

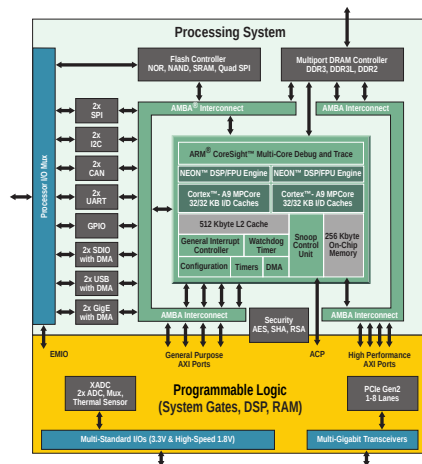


Qu'est-ce qu'un SoC?

Exemple: Zynq-7000 de Xilinx

SoC programmable

- Dual Cortex A9
- Matrice programmable (FPGA)



<http://www.xilinx.com/content/xilinx/en/products/silicon-devices/soc/zynq-7000.html>

Pourquoi intégrer?

- Plus petit
 - Nouvelles applications
- Plus performant
 - Va plus vite
 - Consomme moins
 - Donc de nouvelles applications
- Moins de composants
 - Facture moins élevée
 - Intégration système moins couteuse



Pourquoi intégrer?

Plus petit

La miniaturisation vient principalement de l'intégration de nouvelles fonctionnalités dans une même puce.

Ceci permet de réduire la taille des systèmes et de permettre des nouvelles applications.

- smartphone
- IoT...

Pourquoi intégrer?

Moins cher

Moins cher d'un point de vue composant

La loi de Moore veut dire deux choses:

- à coût constant, intégrer plus de fonctionnalités;
 - réseau, multimédia, M.L.
- baisser le coût pour les mêmes fonctionnalités.
 - smartphone entrée de gamme, IoT

Pourquoi intégrer?

Moins cher

Moins cher d'un point de vue système

Moins de composants:

- systèmes embarqués plus simples à concevoir;
 - PCB plus simple et moins cher
- moins de contraintes sur sa conception.
 - moins de liens de communication



Pourquoi intégrer?

Plus performant

Deux aspects importants entrent en considération:

- Performance brute:
 - vitesse de traitement;
 - fréquence de fonctionnement.
- Consommation énergétique.

Ces deux aspects sont parfois incompatibles.

Pourquoi intégrer?

Plus performant

La puissance dissipée dans un circuit électronique numérique peut s'exprimer de la forme:

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

Où:

f est la fréquence de fonctionnement et

α le taux d'activité.

sont liés à l'application et:

C la capacité équivalente (liée au nombre de portes)

V_{dd} est la tension d'alimentation,

sont liés à la "physique" du circuit.

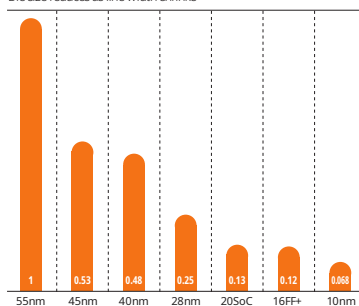
Pourquoi intégrer?

Plus performant

On profite des avancées des technologies de fabrication.

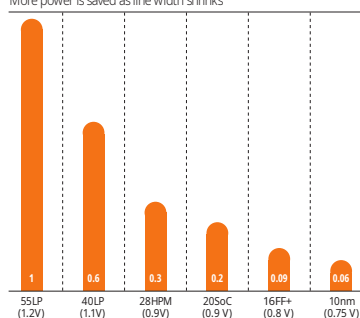
Chip Die Size Cross-Technology Comparison

Die size reduces as line width shrinks



Chip Total Power Consumption Cross-Technology Comparison

More power is saved as line width shrinks



source <http://www.tsmc.com/download/ir/annualReports/2015/english/index.html>

Encore la loi de Moore!

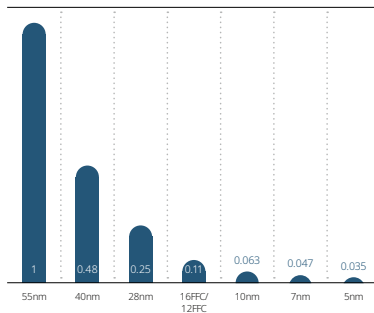
Pourquoi intégrer?

Plus performant

On profite des avancées des technologies de fabrication.

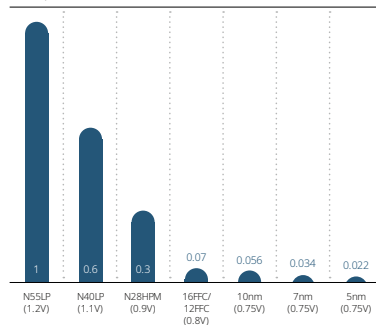
Chip Die Size Cross-Technology Comparison

Die size is shrinking as line width shrinks



Chip Total Power Consumption Cross-Technology Comparison

More power is saved as line width shrinks



source <https://www.tsmc.com/download/ir/annualReports/2018/english/index.html>

Encore la loi de Moore!



Qu'est-ce qu'un SoC?

Spécialiser le SoC

Une chose importante, la spécialisation:

- Faire une chose mais la faire bien/mieux
- Ne mettre que le nécessaire
 - Limiter les couts
 - Le plus efficace pour les besoins de l'application

Être plus performant et plus efficace tout en restant moins chers devant les solutions génériques.



Plan

Système sur puce

Introduction

Logiciel et matériel

Flot de développement

Objectifs de l'UE



Qu'est-ce qu'un SoC?

beaucoup de logiciel

Le maximum en soft

- Si c'est faisable avec un processeur existant on s'arrête là
 - Quelle est la contrainte qui nécessite un matériel spécialisé
- Le développement logiciel est plus simple/moins risqué économiquement
- On utilise des composants sur étagère (*COTS : commercial off-the-shelf*)
- Les clients vous le demanderont
 - de pouvoir modifier, personnaliser, faire évoluer



Qu'est-ce qu'un SoC?

beaucoup de logiciel

- Des solutions logicielles sont parfois suffisantes
- Le logiciel permet une évolution/adaptation plus simple
 - On change le logiciel → on change l'application
 - Corriger/contourner des Bugs!
 - Beaucoup de développeurs
- Ne pas réinventer la roue, beaucoup de choses existent:
 - Des composants tout prêts
 - Systèmes d'exploitation
 - Bibliothèques
 - ...



Qu'est-ce qu'un SoC?

Du matériel

Comment développer du matériel

de façon fiable, en étant sûr de la fonctionnalité et des performances
rapidement, avant d'être rattrapé par la concurrence ou les solutions génériques
sans réinventer la roue, en se concentrant sur la plus-value.

Utiliser des bibliothèques de matériel?

Qu'est-ce qu'un SoC?

Du matériel

On parle de blocs **IPs** (*Intellectual Property*).

On peut:

- Les concevoir soi-même
- Les acheter

Permettant de réduire le temps de développement et partir sur des bases connues et fonctionnelles.

Qu'est-ce qu'un SoC?

Du matériel

De quelles IPs a-t-on besoin?

- Le minimum a avoir
 - Un CPU, des interfaces de communication
 - Utiliser une mémoire externe
- Des accélérateurs pour l'efficacité
 - GPU, DSP, autres
- Des choses impossibles à faire en logiciel
 - Interfaces particulières (radio, optique...)
 - Analogique!

Qu'est-ce qu'un SoC?

Du matériel

Quelles IPs choisir?

- Celles qui entrent dans notre budget
 - On peut les développer
 - On sait les rentabiliser
- Celles qui permettent d'être efficace pour notre fonctionnalité
 - vitesse, consommation, surface, prix
- Celles pour lesquelles on a du support
 - bibliothèques logicielles, OS, drivers
 - garanties?

Qu'est-ce qu'un SoC?

Exemple d'IPs déjà disponibles

Quelles IPs sont disponibles?

- Cœurs de processeurs:
 - ARM, MIPS, RISC-V, 8051, ad-hoc ...
- Accélérateurs:
 - Graphique, vidéo, signal (DSP), IA ...
- Interfaces
 - mémoires, USB, SPI, PCI, CAN...
- Infrastructure
 - Bus, cache

Qu'est-ce qu'un SoC?

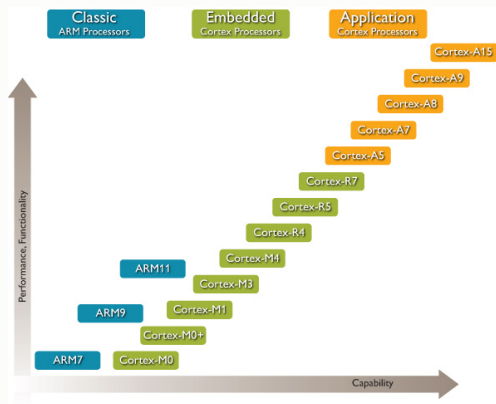
Que vous fournit-on?

- Du RTL
 - Verilog, VHDL...
- Des netlists
 - pré-synthétisées
- Des blocs placés-routés
 - IPs analogiques ou numériques
- des specs, des modèles ...

Qu'est-ce qu'un SoC?

Exemple de fournisseurs d'IPs: ARM

Cœurs de CPU:



Source (2012)

<http://arm.com/products/processors/index.php>

- Des cœurs de CPU allant du microcontrôleur basse consommation au processeur “applicatif haute” performance.
- Infrastructure autour de ces processeurs (bus, cache, smp...)
- Développement de logiciel associé
 - OS, compilateur, bibliothèques spécialisée

Qu'est-ce qu'un SoC?

Exemple de fournisseurs d'IPs: Imagination Technologies

Coprocesseurs graphiques



Source <http://www.imgtec.com>

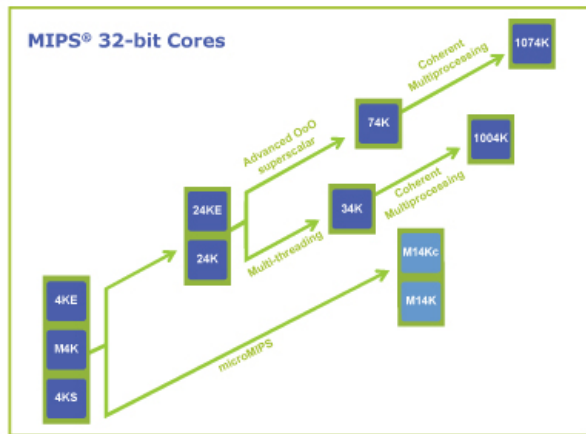
- Coprocesseurs graphiques
 - OpenGL/OpenGL ES
 - Codec vidéo ...
- Logiciel support
 - Drivers
 - Bibliothèques

Qu'est-ce qu'un SoC?

Exemple de fournisseurs d'IPs: MIPS

- Des cœurs de CPU 32 et 64bit
- Développement de logiciel associé
 - OS, compilateur, bibliothèques spécialisée

Cœurs de CPU:



Source <http://mips.com/products/processor-cores>

Qu'est-ce qu'un SoC?

Exemple de fournisseurs d'IPs

Les fondeurs

- STMicroelectronics, Samsung, TSMC ...

Les vendeurs d'outils de CAO

- Cadence, Mentor, Synopsys

Les autres

- Qui ne font que ça ...
- Produits de niche

Qu'est-ce qu'un SoC?

On peut concevoir un SoC en ne faisant qu'assembler des IPs existantes!

Si on veut se différencier il faut apporter une plus-value:

- Premier sur une fonctionnalité,
- Efficacité, coût
- Environnement logiciel, matériel, support
- Niche? (personne ne trouve ça intéressant)

Qu'est-ce qu'un SoC?

Résumons

Donc, pour concevoir un SoC il faut conjointement:

- Concevoir du matériel

- Réutiliser des éléments existants (des IPs)
- Développer des éléments spécifiques nouveaux

- Concevoir du logiciel

- Réutiliser des bibliothèques existantes
- Développer du code spécifique nouveau
- Interagir avec le matériel

- Que l'ensemble respecte des contraintes de:

- temps réel, taille du circuit, consommation, délais, prix ...



Plan

Système sur puce

Introduction

Logiciel et matériel

Flot de développement

Objectifs de l'UE



Flot de développement

Acteurs extérieurs

Un développement logiciel peut être fait entièrement en interne.

Pour fabriquer un circuit on passe par plusieurs acteurs extérieurs:

- Une entreprise qui fabrique le circuit (**fonderie, fondeur, fab**)
- Une entreprise s'occupe de la mise en boîtier (**packaging**)
- Une entreprise qui fabrique le circuit imprimé, assemble...

Le coût de fonctionnement d'une fonderie est tel qu'il y en a très peu dans le monde (sur les doigts d'une main)



Flot de développement

Échelle de temps

Les développements logiciels et matériels se font sur des échelles de temps différentes:

- Les accès à la fonderie sont programmés à l'avance et on ne doit pas rater son créneau
- **Fondre** et faire le packaging prennent plusieurs mois
- Durant ce temps, on attend (ou on fait autre chose)



Flot de développement

Irréversibilité

Un logiciel qui ne fonctionne pas peut être corrigé ou même réécrit entièrement.
Un circuit qui ne fonctionne pas comme prévu ...

- Pas la bonne fonctionnalité, bug,
- Pas les bonnes performances (fréquence, consommation...)

Le coût de fabrication d'un circuit se chiffre en millions de \$.

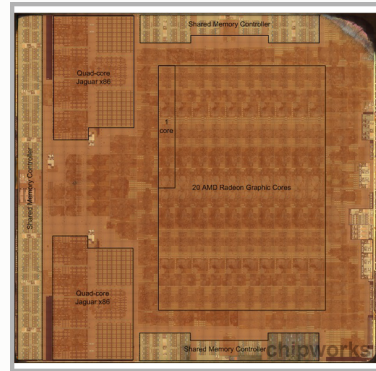
Passer de ça...

```
// 60 frames/s display
Thread1{
    every(16.66 ms)
        display(image)
}

// scene update
Thread2{
    if(ViewChanged)
        redraw(image)
}

// player interaction
Thread3{
    wait(player1)
    move(rabbit0)
}
```

...à ça!



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

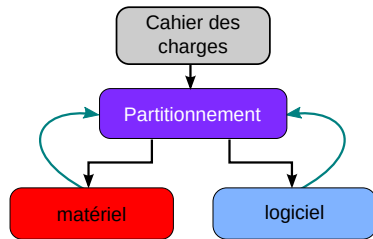
Flot de développement

Le partitionnement

À partir du cahier des charges vous devrez définir:

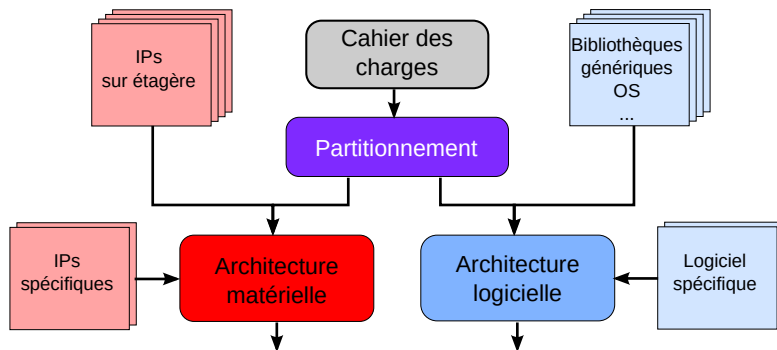
■ Ce Qu'il faut faire:

- En logiciel
- En Matériel



Flot de développement

Le partitionnement

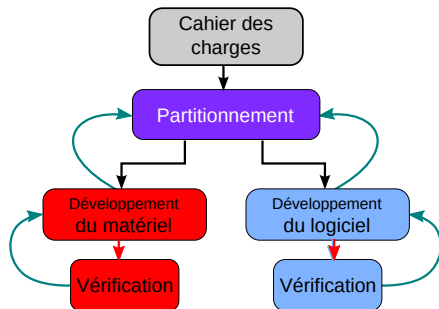


Flot de développement

Le développement

■ Développement de logiciel/matériel

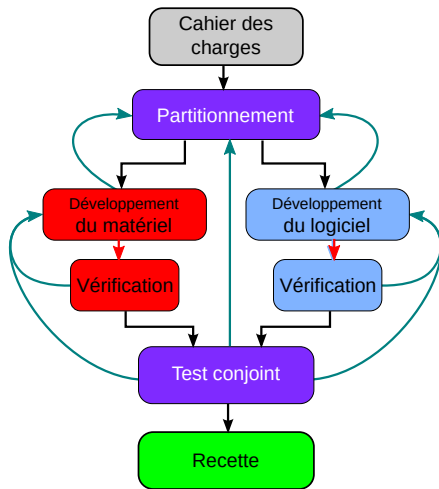
- Équipes différentes
- Rythmes différents
- Vérifications différentes



Flot de développement

La validation

- Valider le résultat de chaque flot par rapport à l'autre

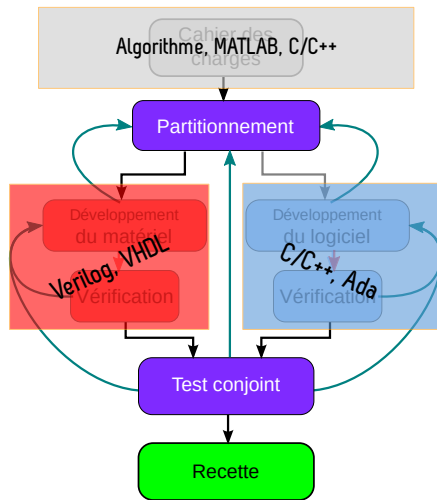


Flot de développement

La complexité

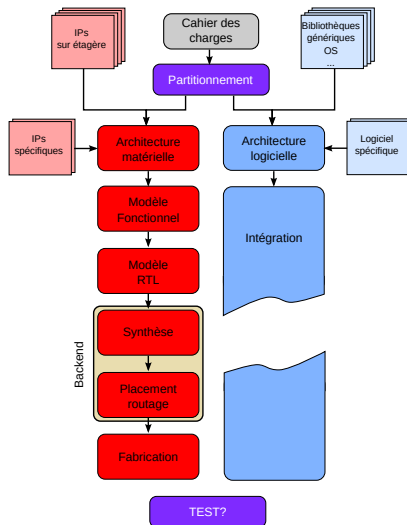
■ Dans la vie réelle

- Équipes différentes
- Langages informatiques différents
- Philosophies différentes



Flot de développement

Les enjeux





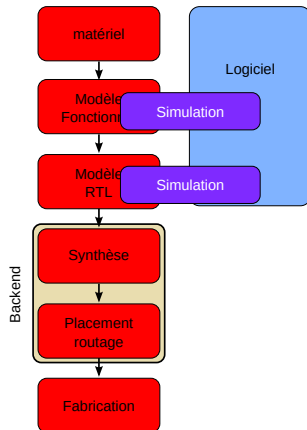
Flot de développement

Les enjeux

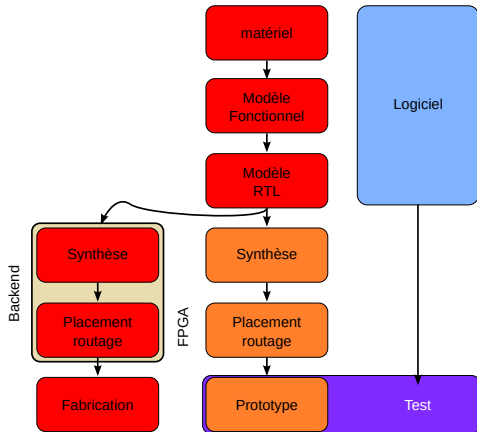
- Commencer le développement du logiciel le plus tôt possible
 - Comment faire avant que le matériel ne soit disponible?
 - Plus on avance dans le flot HW plus c'est compliqué de faire des modifications
- Garder un référentiel unique tout au long des phases de conception/vérification
 - Le plus proche du résultat final
 - Le plus pratique pour les différentes équipes

La simulation

- Utiliser un modèle du matériel
 - Quelle est la précision du modèle (cycle → transactionnel).
 - Quelle est la vitesse de simulation.
- Utiliser le RTL
 - En général très lent.
 - Est-il disponible pour toutes les IPs?
 - Il faut attendre qu'il soit prêt.
- Au delà
 - Encore plus lent
 - Couteux



Flot de développement Prototype FPGA

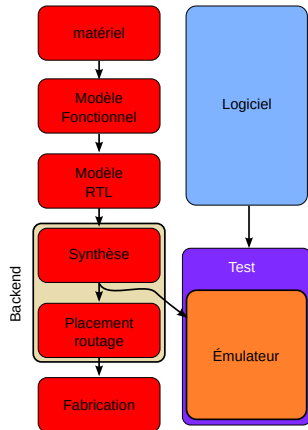


Le prototype FPGA

- Le RTL doit être prêt
 - Permet le test fonctionnel.
 - Permet de valider le logiciel.
 - Moins rapide que le circuit final.
- Le “backend” n’est pas celui de l’ASIC
 - Comportement et bugs introduits par des outils de synthèse différents.

Flot de développement

L'émulation



L'émulation

- Après la synthèse ASIC
- Utilise un émulateur
 - Émule le résultat de la synthèse
 - Utilise une batterie de FPGAs
 - Beaucoup plus lent que le circuit final
- Principalement pour les tests de non-régression
- C'est couteux!



<http://www.mentor.com/products/fv/emulation-systems>

Exemple

- Veloce Strato (Mentor Graphics)
 - une armoire de FPGAs
 - 1.25 Milliard de portes émulées
 - interfaces avec les outils de simulation
 - émulation à quelques de MHz?
- Très couteux
- Réservé à la validation pré-fab

Flot de développement

Les plateformes virtuelles

■ Des plateformes virtuelles

- Simulation rapide (de quoi démarrer un OS)
 - Modèles abstrait, plus haut niveau
- Modèles suffisamment précis (registres, irqs, ...)
 - Pour du développement logiciel
- Interopérabilité (des standards comme SystemC TLM-2)
 - Pour réutiliser des modèles validés

■ Qui permettent

- d'explorer des solutions architecturales
- d'estimer les performances
- de commencer le développement logiciel
- de servir de modèle de référence



Flot de développement

Les plateformes virtuelles

Quelques exemples d'outils commerciaux:

- Fixed Virtual Platforms de ARM [\[web\]](#)
- Open Virtual Platforms (OVP) d'Imperas (ensemble de modèles de processeurs) [\[web\]](#)
- Virtualizer Development Kits (VDKs) de Synopsys [\[web\]](#)
- Virtual System Platform de Cadence [\[web\]](#)

ou opensource:

- **SoCLib** [\[web\]](#) : Ensemble de bibliothèques opensources pour modéliser des SoC au niveau CABA ou TLM
- **SoCRocket** [\[web\]](#) : Plateforme opensource pour modélisant un SoC spatial au niveau TLM

Tout au long du flot de développement matériel:

- Vérifier qu'on respecte les spécifications fonctionnelles
 - Les nôtres et celles imposées par l'extérieur
- Vérifier qu'on respecte des contraintes non fonctionnelles
 - Que se passe-t-il si on fait quelque chose de non prévu!
- Formellement ou en simulant (exhaustivement?)

Il existe des méthodologies et outils de vérification et de preuve.

Dans la partie Backend du flot de développement, d'autres aspects sont vérifiés:

- La taille du circuit (et donc son prix)
- La fréquence de fonctionnement
- La congestion du routage
- La consommation statique et dynamique

On peut être amené à revenir sur des choix d'architecture ou d'implémentation si l'un de ces objectifs n'est pas atteint!



Plan

Système sur puce

Introduction

Logiciel et matériel

Flot de développement

Objectifs de l'UE

Objectifs de l'UE

- Vous transmettre un savoir opérationnel
 - Mise en œuvre d'outils et de principes actuels
 - Conception d'un SoC en utilisant des plateformes virtuelles
 - Raffinement vers du RTL
 - Protocoles industriels et "reuse"
 - La synthèse et le placement routage (notions)
 - SoC sur FPGA
 - Pratique
 - Conception conjointe Logiciel/matériel
 - Prototypage sur FPGA
 - ...
 - Ouverture (5SE06?)
 - Vérification et preuve
 - Synthèse d'architecture
 - ...

Dépôt Git...

Vous aurez des travaux à rendre. Tout passera par votre dépôt Git individuel.
Se connecter à:

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

Demander à être membre du sous-groupe **5se05/2627**

- Groups → Explore Groups → Request Access

Liste de diffusion...

- Une adresse

`https://mattermost.telecom-paris.fr/se/channels/5se05-2627`

Pour poser des questions et y répondre!