La Conception de Systèmes Complexes et Critiques

Un nouveau défi pour l'industrie du semi-conducteur

Nathalie Messina

Advanced System Technology
Wireless Terminal Business Unit
Texas Instruments Europe

n-messina@ti.com

Systèmes Complexes et Critiques ?

"SYSTEME CRITIQUE"

Un système vital pour le fournisseur et/ou son client.

"SYSTEME COMPLEXE "

Un système tel que son organisation est d'une complexité supérieure à celle de ses modules.

Deux Concepts Importants

"**VALIDATION**"

Construisons nous le bon produit ?

"**VERIFICATION**"

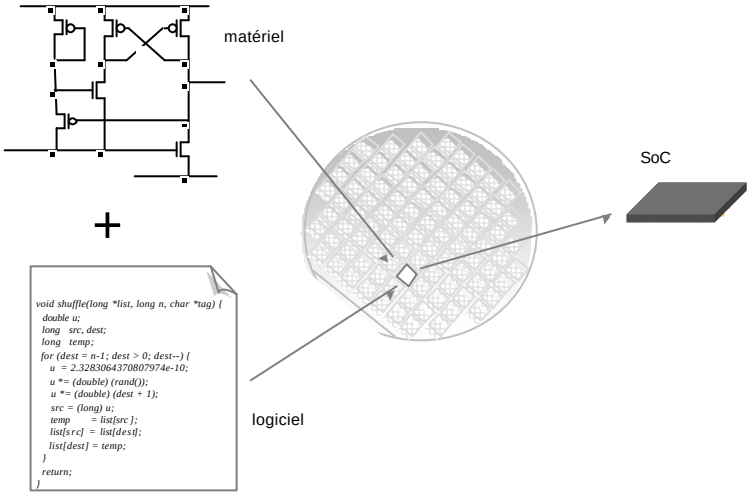
Construisons nous le produit correctement ?

Plan de la présentation

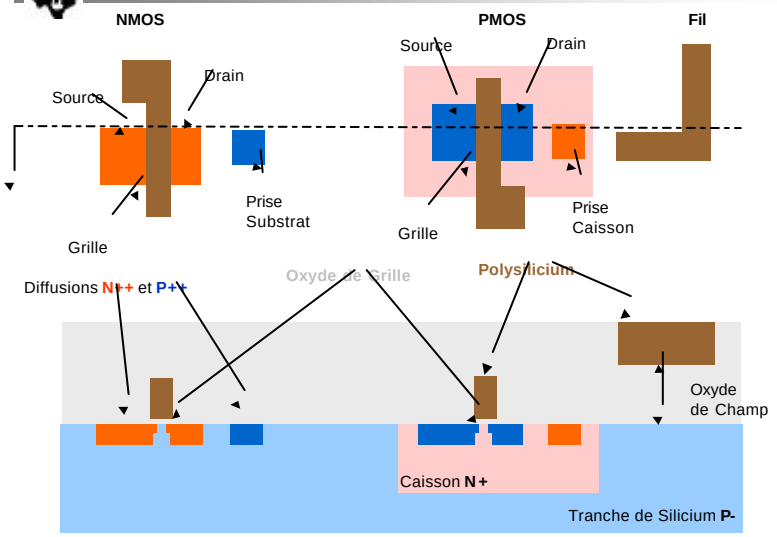
SoC 

- Notre domaine de compétence
- Notre problème
- Spécification de la solution
- Quelques pistes vers la solution
- Conclusion

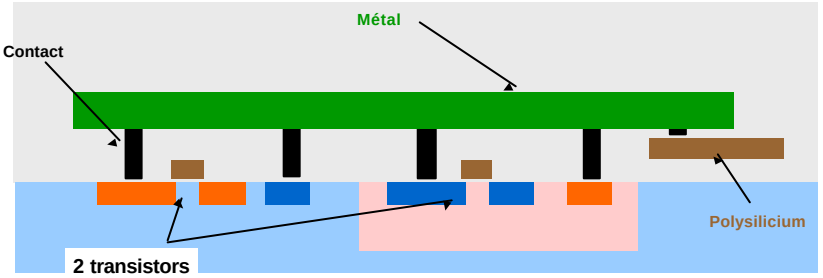
SoC: “System on Chip”



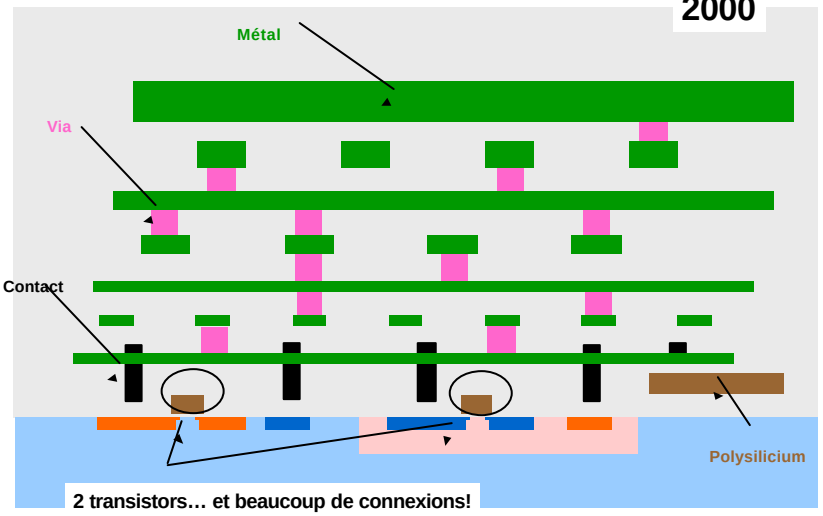
Un procédé de fabrication typique



Un circuit intégré: des transistors connectés

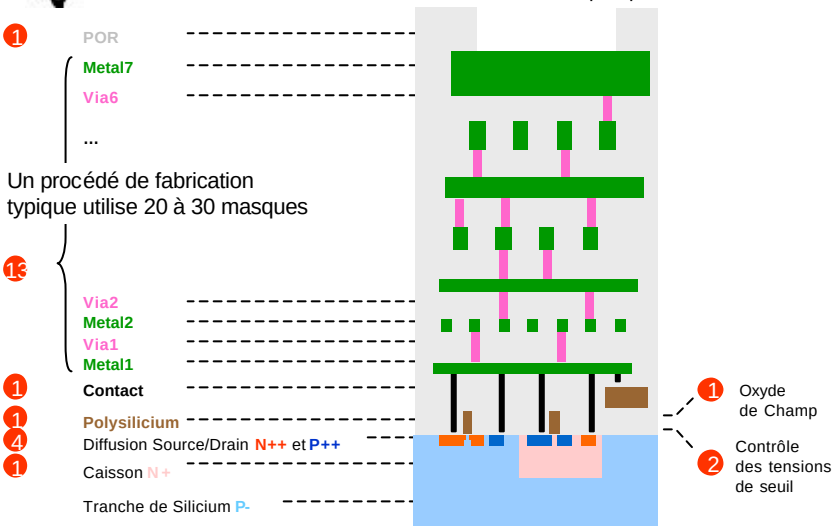


Un circuit intégré moderne



Étapes de fabrication

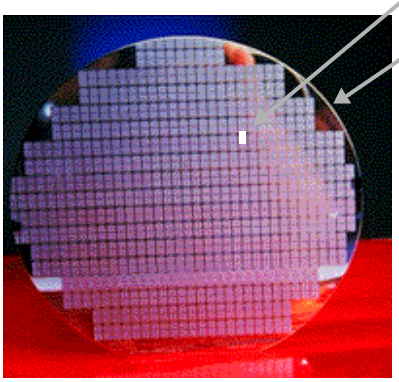
9 masques pour les transistors
15 masques pour les connexions



Quelques chiffres

$

Une tranche de silicium



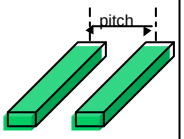
Diamètre d'une Tranche	300 mm
Surface Utile	62 500 mm^2
Puce	25 mm^2
# Puces / Tranche	2 500
1 lot	40 Tranches
# Puces / Lot	**100 000**

Une unité de fabrication

Coût d'une Unité: **3 000 000 000 de $** *(2005)*

Un Procédé de Fabrication Avancé

Exemple:	*Diamètre de la tranche*	*300 mm*
	"Pitch" Métal	*200 nm*
	Longueur de canal d'un transistor	*65 nm*



Coût d'une unité de fabrication: **3 000 000 000 $**
Coût d'un jeu de masques: **3 000 000 $** (2005, *Sematech*)
Coût d'un lot (40 tranches): **($$$)**

Durée de la fabrication d'un lot: **2 à 6 semaines**

Nombre de puces par lot: 25 000 (100 mm^2)
...
500 000 (5 mm^2)

Coût d'une puce: **($)** elle ne peut pas être chère!

L'impact d'un bug dans un SoC

Exemple:

1 bug dans un microprocesseur récent
=
500 000 000 $

Définition

“SYSTEME CRITIQUE”

*Un système **vital** pour le fournisseur et/ou son client.*

Ne pas livrer un tel système à temps pousse une entreprise hors du marché.

D’autres chiffres

1 2 5
3 4
6

Discontinuité d'échelle (Matériel)

En Nombre

1 puce complexe contient typiquement 100 000 000 transistors.

Il faudrait plus de **3 ans** pour vérifier ces transistors, un par un, au rythme d'un transistor par seconde, sans s'arrêter …

En Taille

Il faut 25 couleurs différentes pour colorer le dessin des 25 masques.
Il faut un grandissement de 10 000 pour voir à l'œil nu une ligne de 100 nm.

Le dessin d'un circuit de 1 cm x 1 cm demanderait donc une surface de **100 m x 100 m**, avec 25 niveaux colorés, et des détails de 1 mm …

Discontinuité d'échelle (Logiciel)

En Taille

1 000 000 de lignes de code en langage de haut niveau

En Complexité

100 modules

Le Développement d'un SOC

Développement Matériel: 100 Années Hommes

10 transistors par personne et par jour
10 lignes de VHDL par personne et par jour

Développement Logiciel: 100 Années Hommes

10 lignes de code C par personne et par jour

Taille de la Base de Données: 1000 G Bytes par SoC

Définition

"SYSTEME COMPLEXE "

*Un système tel que son **organisation** est d'une complexité supérieure à celle de ses modules.*

Personne n'espère découvrir un problème d'architecture d'un tel système pendant son développement.

Un SoC aujourd'hui

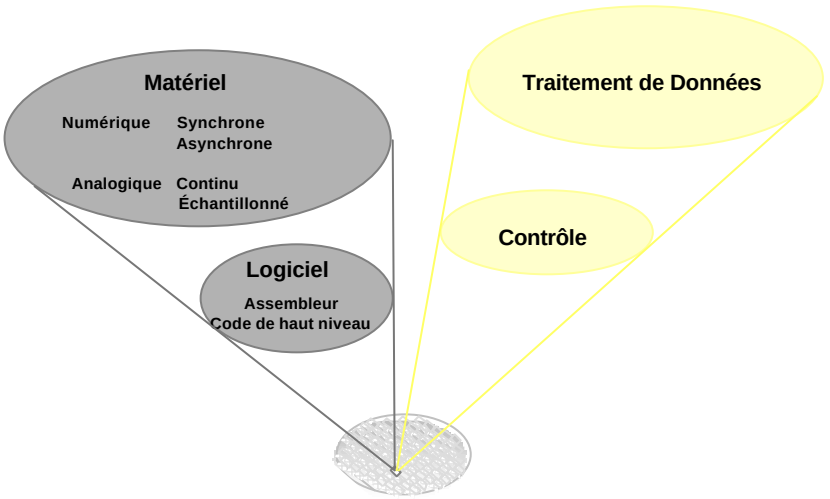
Les SoC's sont

COMPLEXES ET CRITIQUES

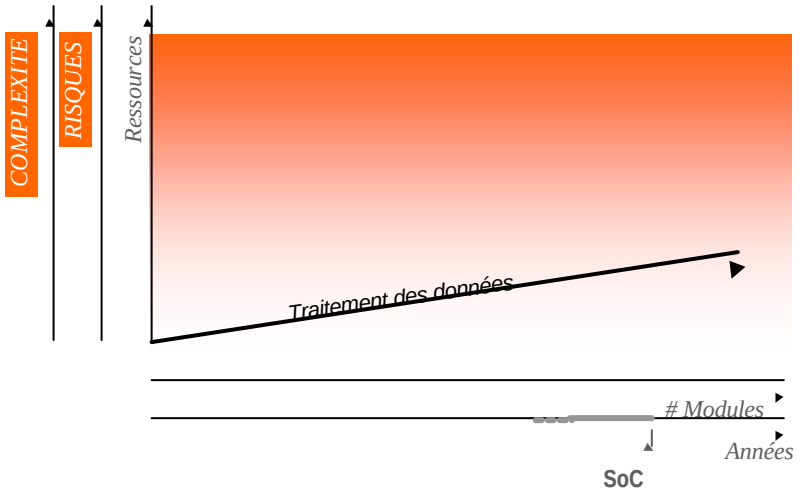
Plan de la présentation

- Notre domaine de compétence
- Notre problème
- Spécification de la solution
- Quelques pistes vers la solution
- Conclusion

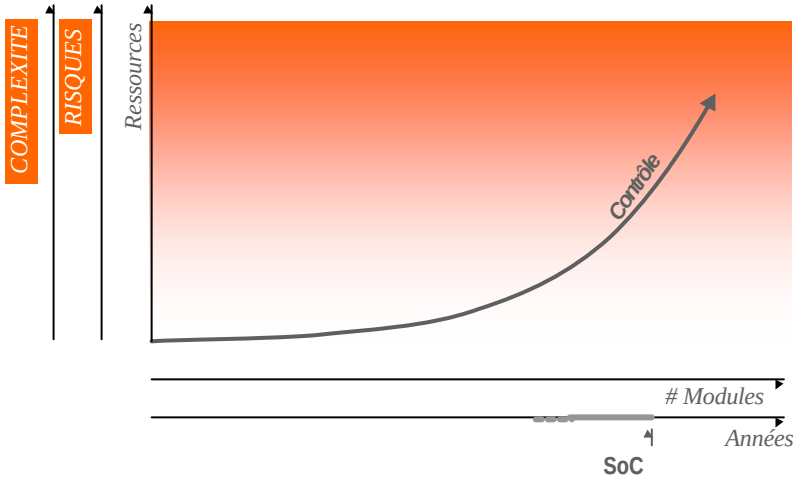
A l'intérieur d'un SoC



Le développement des SoC's: le Traitement des données

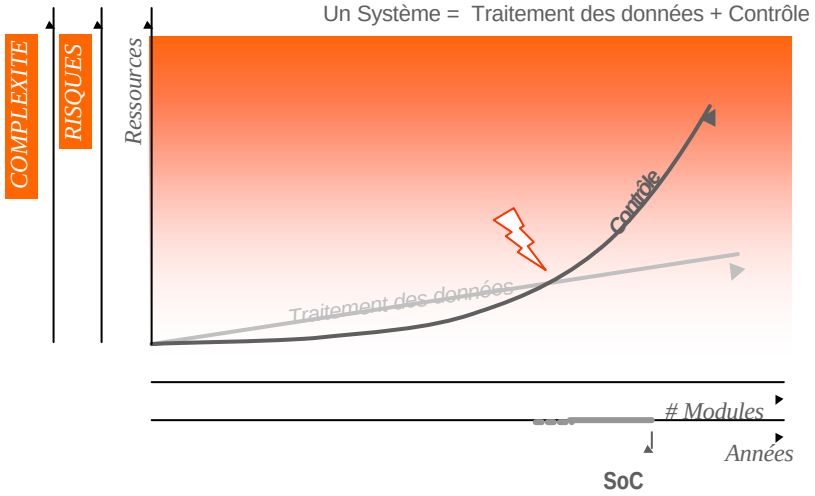


Le développement des SoC's: le Contrôle

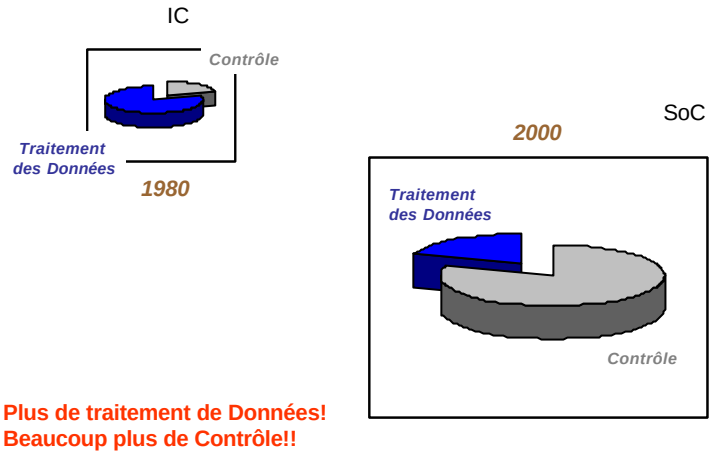


Discontinuité dans le développement des Systèmes

Un Système = Traitement des données + Contrôle



Une Discontinuité = Un changement de métier



Plus de traitement de Données!
Beaucoup plus de Contrôle!!

Une Discontinuité = Un changement de point de vue

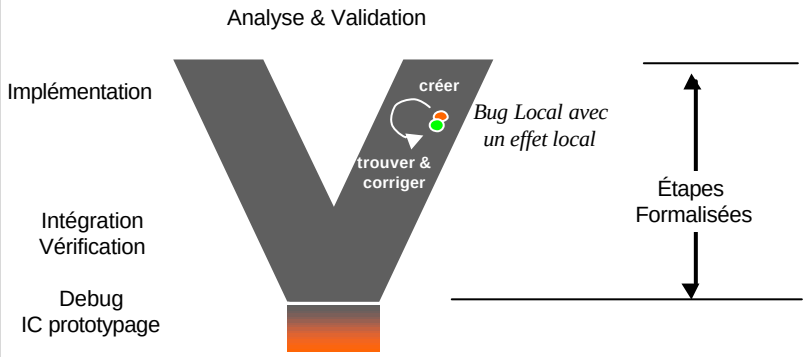
La complexité des systèmes dominés par du **traitement des données** réside dans les algorithmes sur les données.

Il s'agit d'un **problème LOCAL**.

La complexité des systèmes dominés par le **contrôle** et qui réagissent avec leur environnement réside dans l'organisation du système.

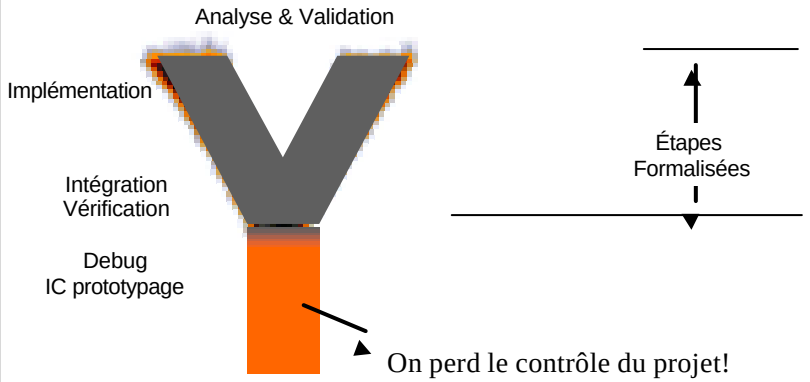
Il s'agit d'un **problème GLOBAL**.

Le Traditionnel Flot de Développement…



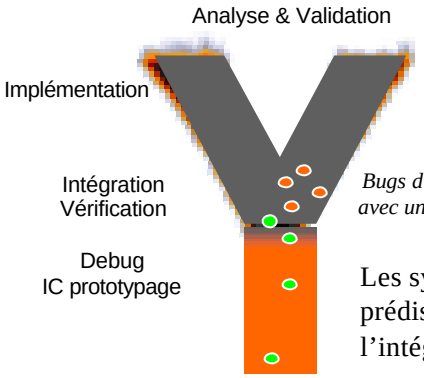
Ce flot a montré une bonne efficacité et réactivité pour les systèmes à dominance traitement du signal de ces 10 dernières années.

… Pour un Système Complexe ?



On perd le contrôle du projet!

… Pourquoi ?



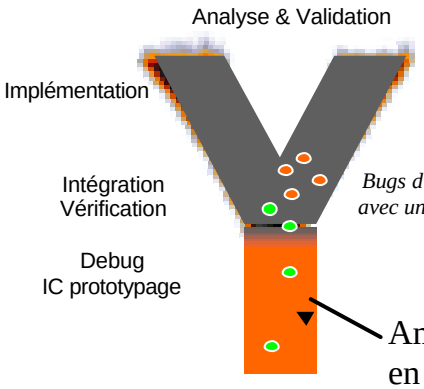
Bugs d'intégration avec un effet global

Étapes Formalisées

Les systèmes à dominance contrôle sont prédisposés aux bugs introduits pendant l'intégration.

Découverts tard, et impliquant le système complet, les bugs sont difficiles à trouver, difficiles à simuler et longs à corriger.

Action Corrective!



Bugs d'intégration avec un effet global

Étapes Formalisées

Améliorons l'exécution en ajoutant plus de ressources!

La bonne question ?

Sommes nous certains que tous nos problèmes viennent uniquement de l'exécution ?

Croyons nous encore que les designers ("software" ou "hardware") sont capables de percevoir tous les bugs système pendant la phase d'implémentation ?

En d'autres termes, est-ce une bonne chose de continuer à injecter de plus en plus de ressources à la fin d'un projet ?

Plan de la présentation

- Notre domaine de compétence
- Notre problème
- Spécification de la solution
- Quelques pistes vers la solution
- Conclusion

Une Discontinuité = Un Changement de métier

Une Discontinuité = Un changement de point de vue

Nous avons besoin d'une
nouvelle méthode de travail

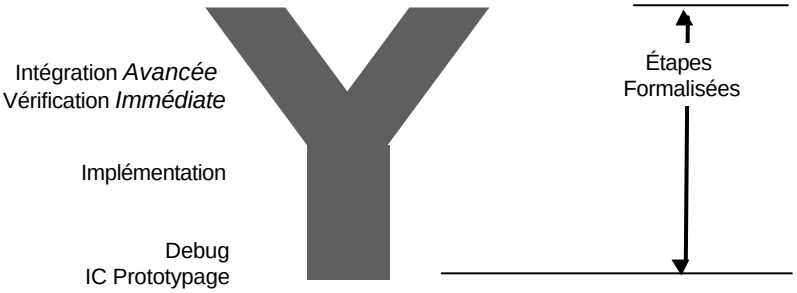
Que font nos voisins ?

D'autres ont adressé la **complexité** des systèmes **critiques** avant nous:

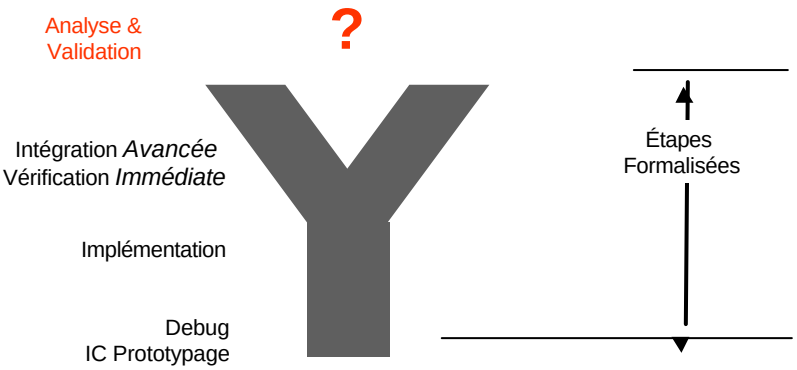
- l'industrie aérospatiale
- l'industrie nucléaire
- l'industrie du transport
- l'industrie automobile
- ...

Pour leurs systèmes logiciels, ils utilisent le flot de développement en "**Y**" et la **vérification formelle**.

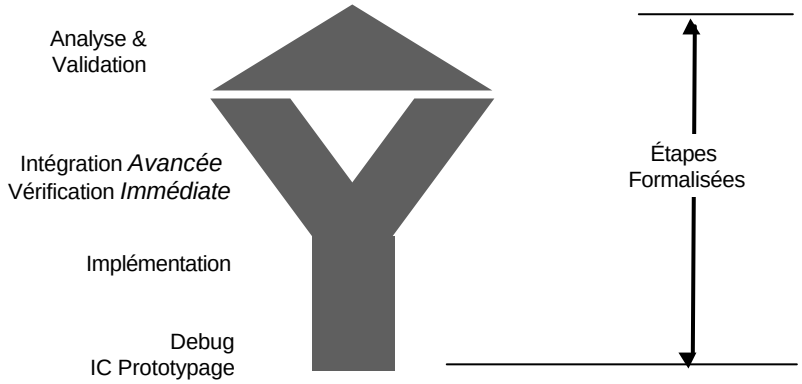
Le Flot de Développement pour un Système Complexe



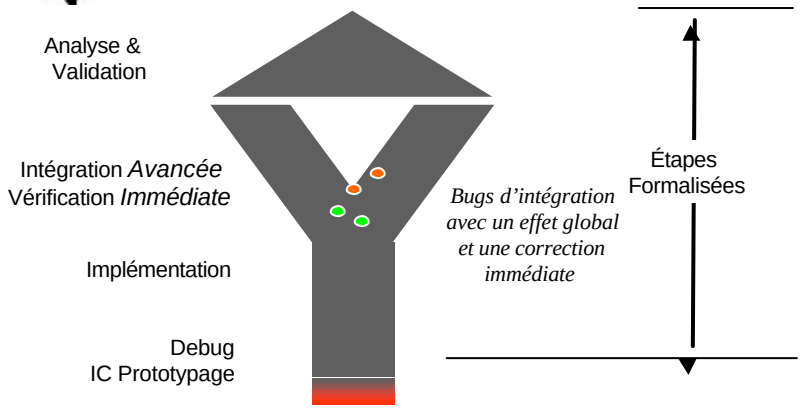
Le Flot de Développement pour un Système Complexe



Le Flot de Développement pour un Système Complexe



Le Flot de Développement pour un Système Complexe



Découverts tôt grâce à une description de haut niveau,
les erreurs sont plus faciles à trouver et plus simples à corriger.

Une Nouvelle Méthodologie !

Nous avons besoin d'une **méthodologie commune** pour les différentes compétences nécessaires au développement d'un SoC complexe.

Notre but est de fournir un **environnement de travail** commun a l'intégration de systèmes avancés sur une puce.

Nous avons besoin d'un nouveau flot de développement qui nous permettra d'avoir une **spécification exécutable** et une plateforme pour la **vérification formelle**.

Une Nouvelle Méthodologie !

Nous devons renforcer et formaliser la **capture des besoins** du client.

Nous avons besoin d'une méthode pour **créer** et non pas pour décrire ou vérifier un solution existante.

Le flot de développement doit être **continu** depuis la capture des besoins jusqu'à la réception par le client.

La méthodologie doit définir clairement la **responsabilité** de chacun et renforcer le **dialogue** entre les différents acteurs.

Elle doit produire naturellement une **documentation** adéquate et complète.

Plan de la présentation

- Notre domaine de compétence
- Notre problème
- Spécification de la solution
- Quelques pistes vers la solution
- Conclusion

Les pistes que nous suivons

Formalisation de la **collection des besoins** du client, ce que le système doit faire :

=> **Modèle UML**

Définition des interfaces avec l'environnement

Identification des interactions entre les acteurs et le système

Identification des cas d'utilisation et des scénarios

Écriture des diagrammes de séquences

Les pistes que nous suivons

Analyse objet pour obtenir l'**architecture interne** du système et une bonne définition des modules élémentaires.

=> **Modèle UML**

Identification des objets et de leurs relations

Définition d'une partition fonctionnelle

Analyse et formalisation des messages échangés

Écriture des diagrammes de séquences

Séparation des domaines contrôle et données

Définition des propriétés du système à vérifier

Les pistes que nous suivons

Modèle exécutable du système complet pouvant être prouvé.

=> **Modèle Esterel**

Écriture des modules en accordance avec l'analyse objet incluant le contrôle et les données, le matériel et le logiciel, le numérique et l'analogique.

Simulation des scénarios.

Vérification des propriétés du système par simulation et preuve formelle et définition des postulats, contraintes et spécifications.

Plan de la présentation

- Notre domaine de compétence
- Notre problème
- Spécification de la solution
- Quelques pistes vers la solution
- Conclusion

Mise en perspective

Puces

1960, pas de VERIFICATION PHYSIQUE, pas de FUTUR

Circuit intégré

1980, pas de TESTABILITE, pas de FUTUR

SoCs

2000, pas de VALIDATION et VERIFICATION SYSTEME, pas de FUTUR

Conclusion

Le principal facteur qui limite le développement de systèmes complexes n'est pas la fabrication du silicium mais l'inadéquation de méthodologie de design pour des systèmes complexes comme les SoC's.

Il existe des pistes intéressantes qui devraient nous permettre de répondre au nouveau défi posé par le développement de nos SoC's complexes et critiques.

Nous avons besoin de méthodes adaptées, de moteurs de preuve toujours plus performants, …