

# Chapitre 1

# Introduction aux Systèmes Embarqués

Abdelkader Gouaïch

<http://www.lirmm.fr/~gouaich>

[gouaich@lirmm.fr](mailto:gouaich@lirmm.fr)

# Objectifs du chapitre

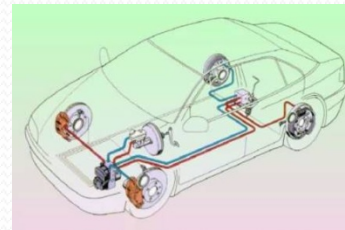
- Définition d'un système embarqué
  - C'est quoi un système embarqué?
  - C'est quoi un logiciel embarqué?
  - Les évolutions des systèmes embarqués?
- Défis de conception – Métrologie
- Technologies des systèmes embarqués
  - Processeurs
  - Méthodes de conception

# Introduction

- Un fait => Ubiquité et omniprésence des « systèmes informatiques »
- Que veut-on dire par « systèmes informatiques » ?
  - Les ordinateurs personnels (PC)
  - Les ordinateurs portables
  - Les serveurs d'application
- Mais, le sens commun nous fait oublier...
  - Une famille plus grande/riche de systèmes informatiques existe sur des architectures particulières !

# Par exemple....

- Exemples de ces systèmes oubliés:
  - Les appareils électroniques
  - Systèmes de contrôle de véhicules
  - Systèmes de navigation



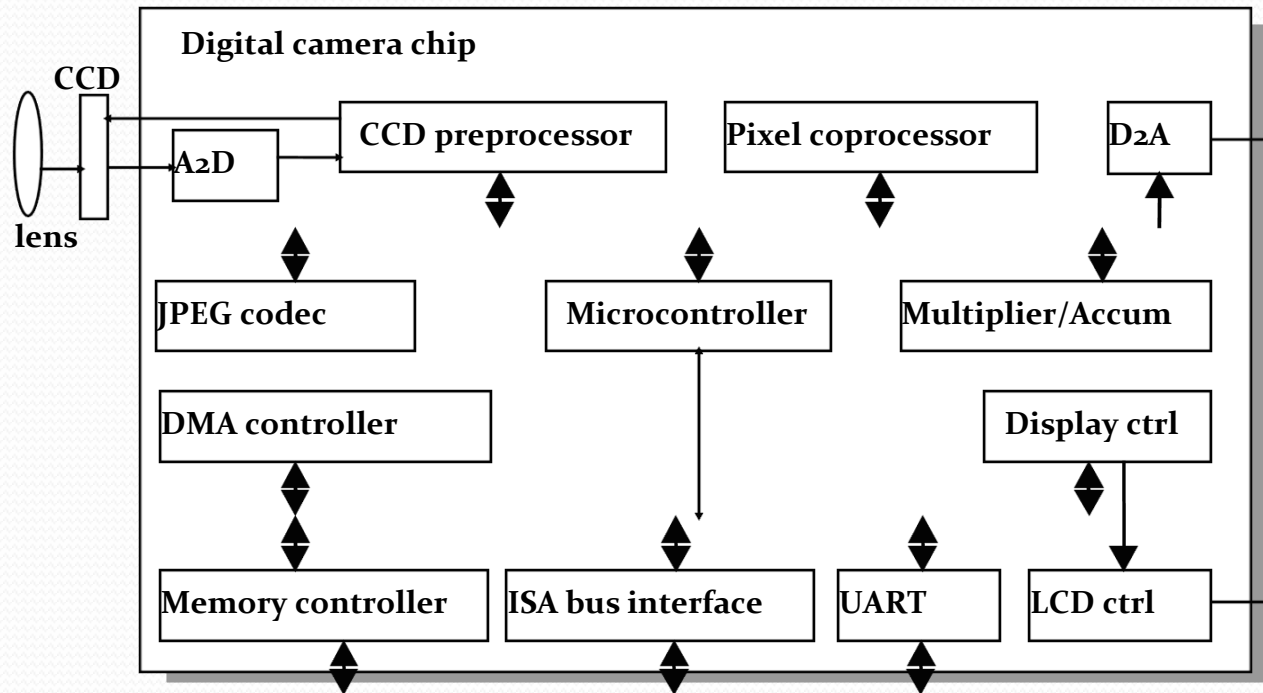
# Définition

- Comment définir ces systèmes?
  - Difficile de trouver une définition ...
- Intuitivement:
  - Tout système informatique différent de l'ordinateur personnel (desktop PC)
- Avec cette définition:
  - On parle de milliards d'unités ! (PC l'ordre est de millions)

# Les caractéristiques d'un système embarqué

- Monofonctionnel (en général)
  - Exécution d'un programme en boucle
- Fortement contraint
  - Coût, taille, rapidité, consommation énergie
- Réactif
  - sensible aux changements de son environnement
- Temps réel:
  - Des contraintes temporelles pour donner le résultat
- Communicants (récemment)
  - Interaction et communication entre composants

# Example:



# Défis pour la conception

- L'objectif
  - Construire un système (dispositif) qui réalise une fonctionnalité désirée
- Le défi majeur:
  - Optimiser simultanément plusieurs métriques de conception
- Métrique de conception:
  - Une propriété mesurable d'une réalisation d'un système
  - L'objectif est d'optimiser l'ensemble des métriques pour juger de la qualité/viabilité d'une réalisation



# Métriques

## Métriques communes

Métriques communes	
Coût unitaire	le coût monétaire de la fabrication de chaque copie du système (NRE exclu)
Coût NRE	Le coût non récurrent monétaire de conception (Non-Recurring Engineering)
Taille	Indicateur de la taille occupée (nbre de transistor, nbre de portes etc)
Performance	temps d'exécution, temps de réponse, etc.
Energie	consommation en énergie du système

# Métriques

## Métriques communes

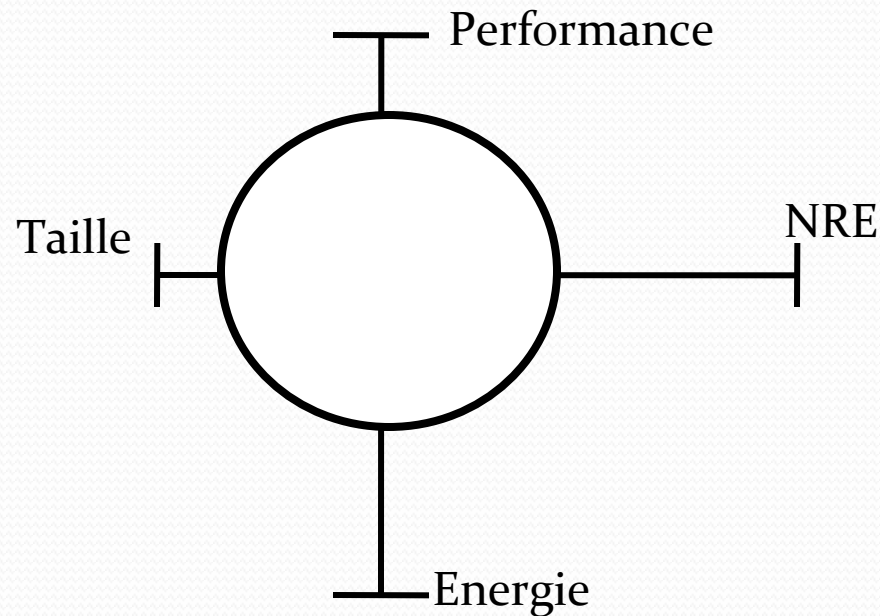
Temps pour le marché (TTM)	Le temps nécessaire pour développer une version du système suffisamment fiable pour être commercialisée.
« Maintenabilité »	Capacité de modifier le système après sa sortie
Consistance	Le système répond correctement aux fonctions attendues.
Sécurité	Le système répond aux normes de sécurité exprimées.
Flexibilité	capacité de modifier les fonctionnalités du système avec un faible NRE
Temps du prototype (TTP)	Temps pour avoir la première version (qui fonctionne) du système

# Optimisation des métriques

- Il faut avoir une expertise du matériel ET du logiciel
  - Expertise Hardware et Software ce qui est rare de nos jours...
- Il faut avoir une connaissance précise des technologies
- Ceci permet d'avoir une connaissance complète pour prendre des décisions

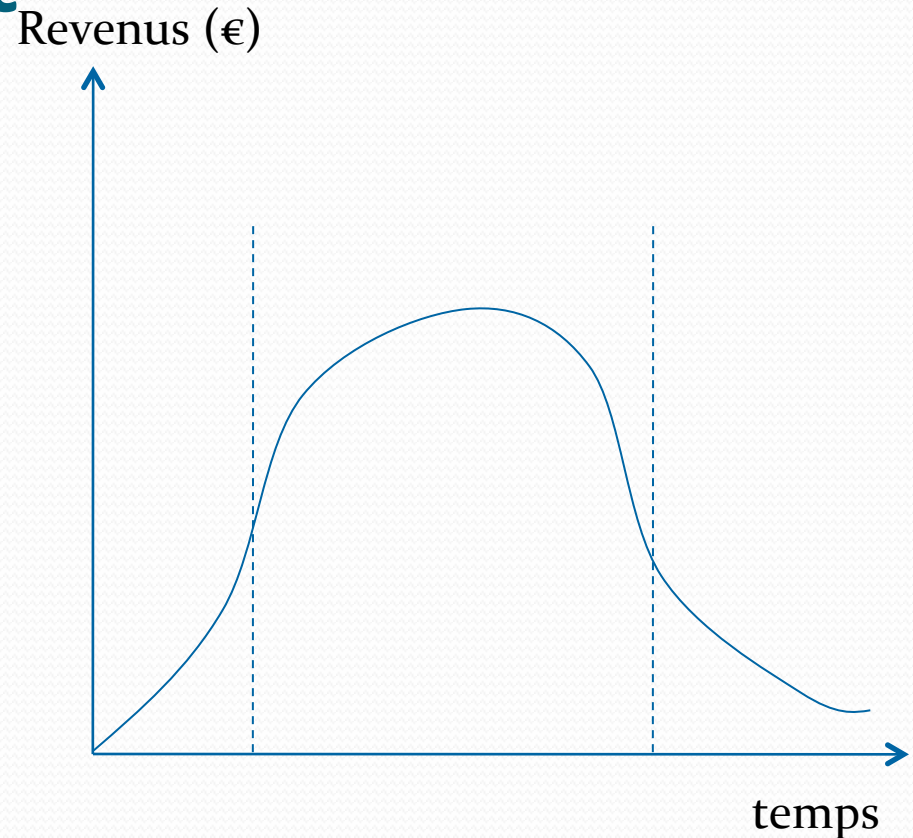
# Optimisation des métriques

- Les métriques sont interdépendantes
- Améliorer une métrique peut détériorer une autre

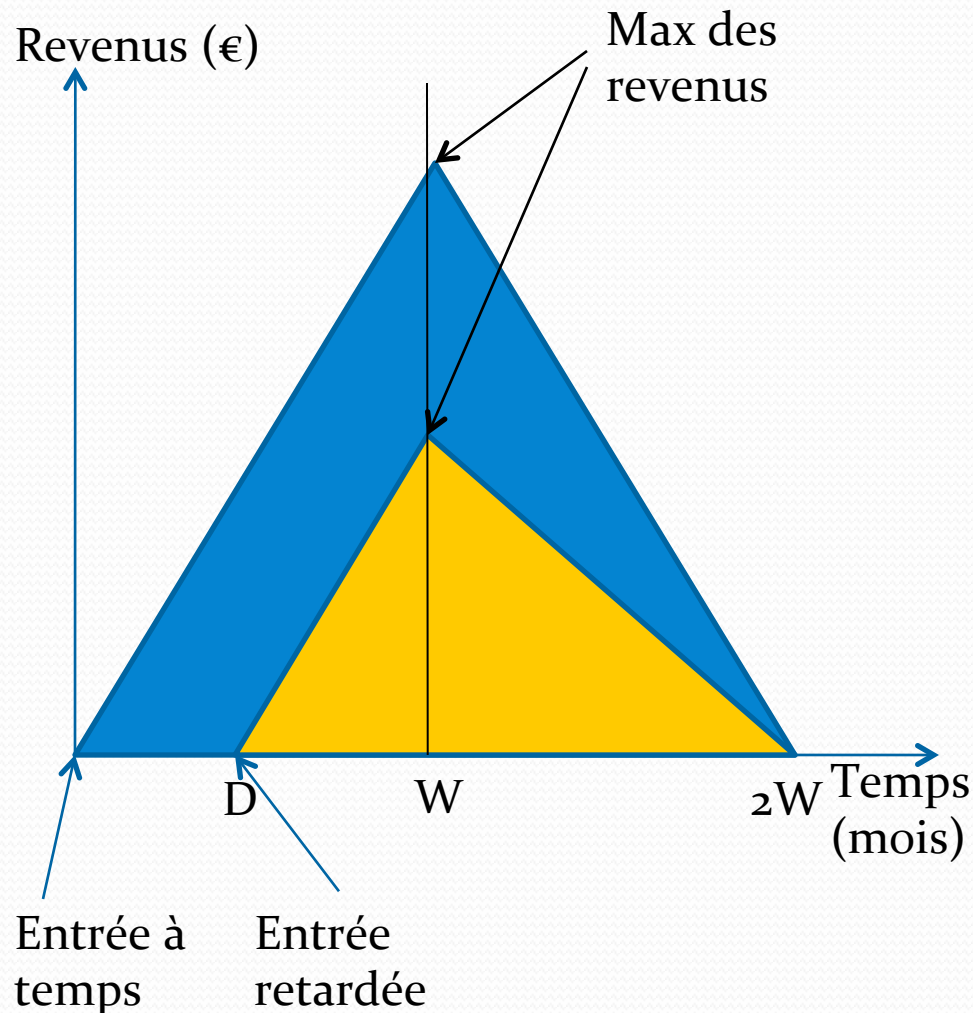


# Time to Market

- Temps nécessaire pour amener un produit sur le marché
- Fenêtre du marché (market window)
  - Période de temps où un produit réalise des ventes
- Il faut respecter le TTM moyen de son secteur (~8 mois pour SE)
- Les retards sont extrêmement pénalisants

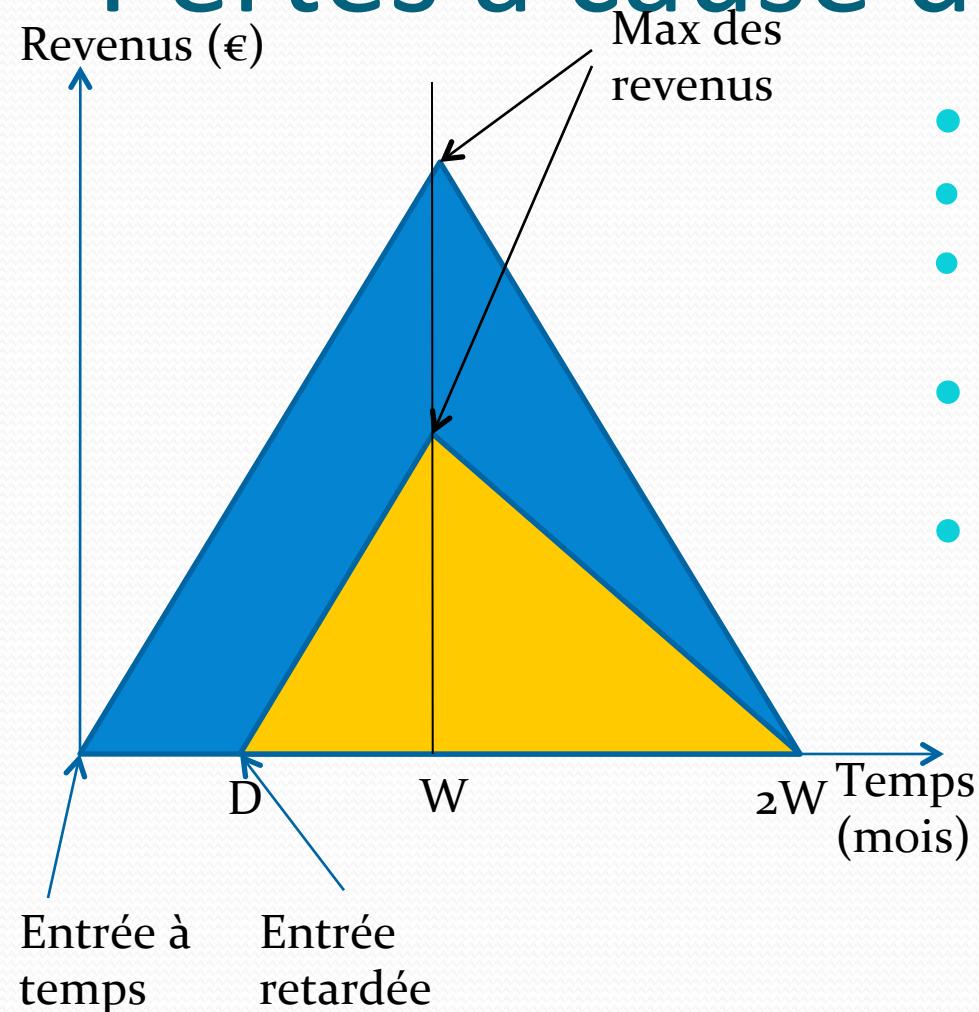


# Pertes à cause des retards



- Modèle de revenu d'un produit
  - Vie du produit =  $2W$  avec un max à  $W$
  - Le TTM va définir un triangle qui représente la pénétration du marché (modèle)
  - Le revenu pour le produit = surface du triangle
- Perte:
  - Différence entre les surfaces

# Pertes à cause des retards



- Surface = base \* hauteur \* 1/2
- À temps =  $1/2 * 2W * W$
- Retardée =  $1/2 * (W - D + W) * (W - D)$
- Pourcentage de perte =  $(D(3W - D) / 2W^2) * 100\%$
- Exemples:
  - Vie 2W = 52 sems, retard D = 4 sem
  - $(4 * (3 * 26 - 4) / 2 * 26^2) = 22\%$
  - Vie 2W = 52 wks, retard D = 10 sems
  - $(10 * (3 * 26 - 10) / 2 * 26^2) = 50\%$

# NRE et coût unitaire

- Coût unitaire: coût monétaire de chaque copie du système sans inclure le NRE
- NRE: coût singulier de conception du système
- Coût total =  $\text{NRE} + \text{coût unitaire} * \# \text{ d'unités}$
- Coût par produit  $= \text{Coût total} / \# \text{ d'unités}$   
 $= (\text{NRE} / \# \text{ d'unités}) + \text{coût unitaire}$



# NRE et coût unitaire

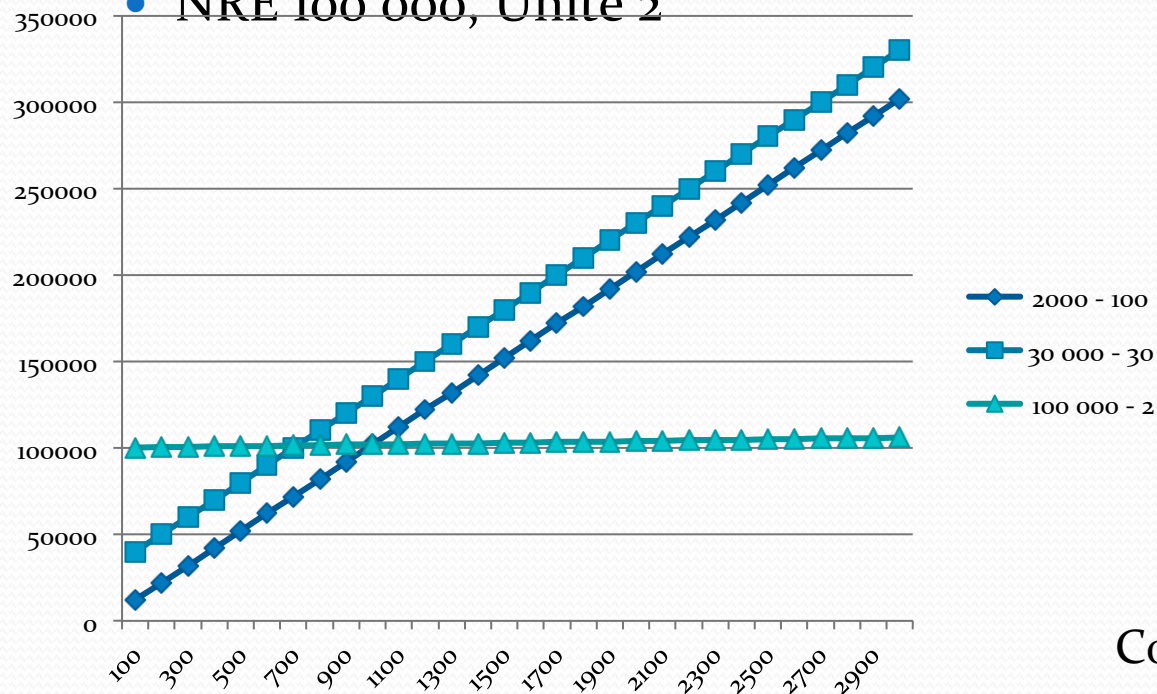
- Exemple:

- NRE = 10000€, coût unitaire = 500€
- Pour 100 unités:
  - Coût total =  $10000 + 500 \times 100 = 60000\text{€}$
  - Coût par produit =  $10000/100 + 500 = 600\text{€}$

# NRE et coût unitaire

- Comparaison

- NRE 2000, Unité 100
- NRE 30000, Unité 30
- NRE 100 000, Unité 2

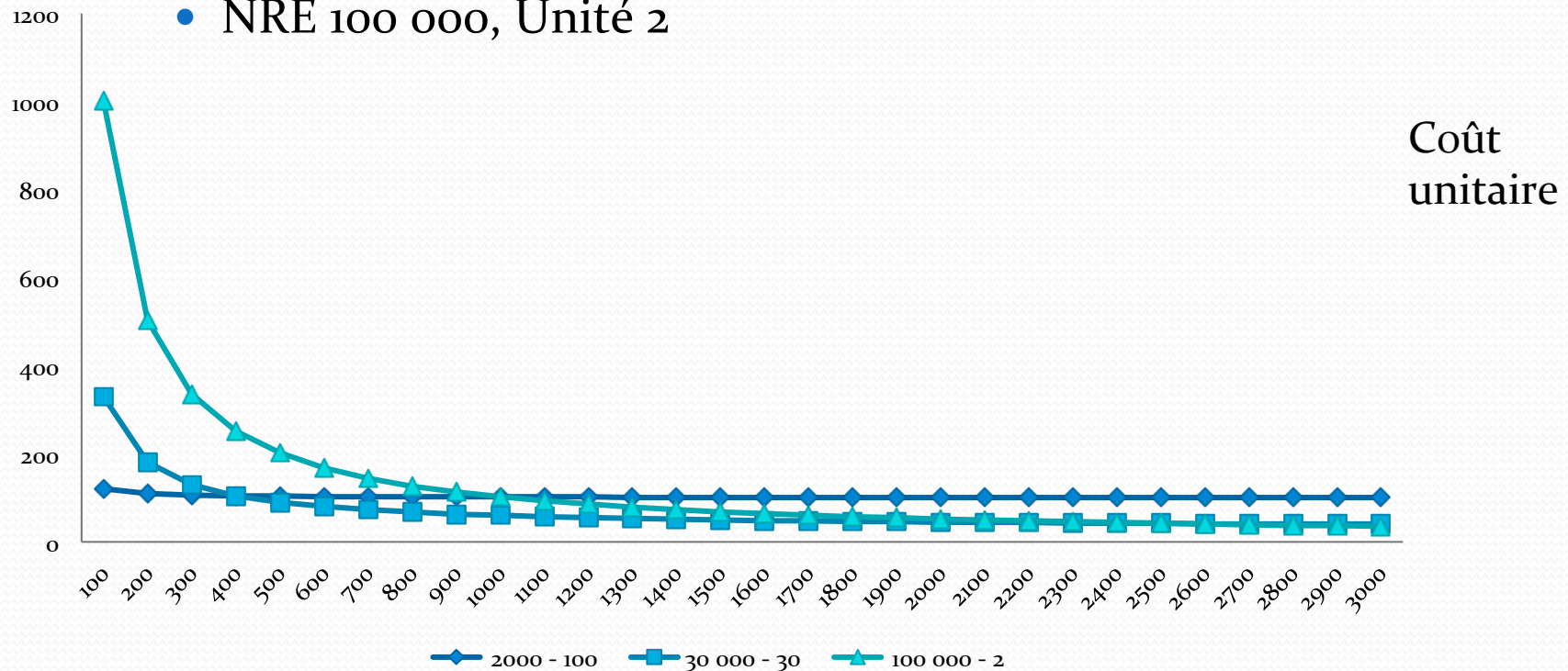


Coût total

# NRE et coût unitaire

- Comparaison

- NRE 2000, Unité 100
- NRE 30000, Unité 30
- NRE 100 000, Unité 2



# Les métriques de performance

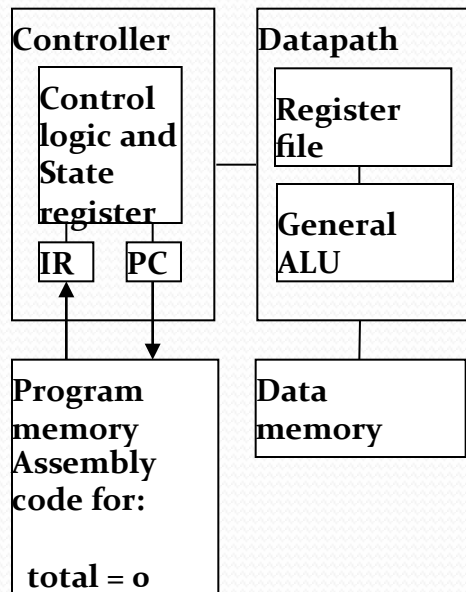
- Mesures utilisées pour indiquer les perfs
  - Fréquence horloge, instruction par sec.
  - Des mesures limitées car quantitatives et pas qualitatives: Ex. Appareil photo numérique
- Temps de latence (temps de réponse)
  - Temps de début et de fin d'une tâche
- Production
  - Nombre de tâches par sec.

# Technologies des systèmes embarqués

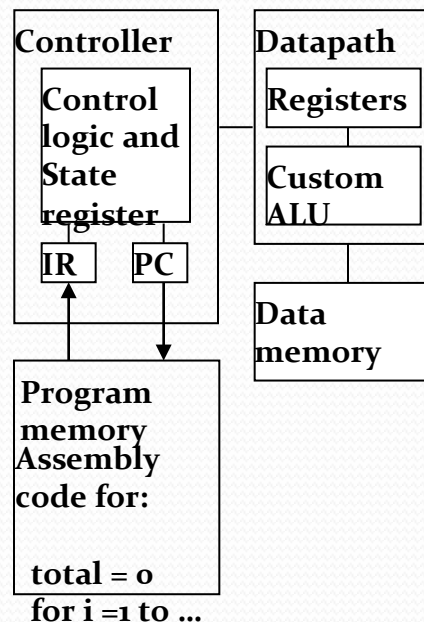
- Technologie:
  - accomplir des tâches en utilisant des procédés techniques, méthodologiques ou du savoir faire
- 3 technologies clefs à maîtriser
  - Processeur
  - Circuits intégrés
  - Méthodes de conception et d'implémentation

# Technologies: Processeur(s)

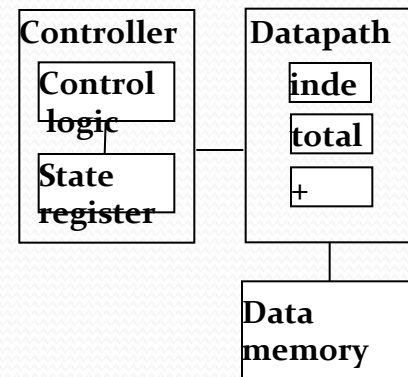
- C'est le moteur de la computation
- Dans le monde des SE on utilise plusieurs familles de processeurs
- Ne pas confondre processeur et processeur programmable



General-purpose (“software”)



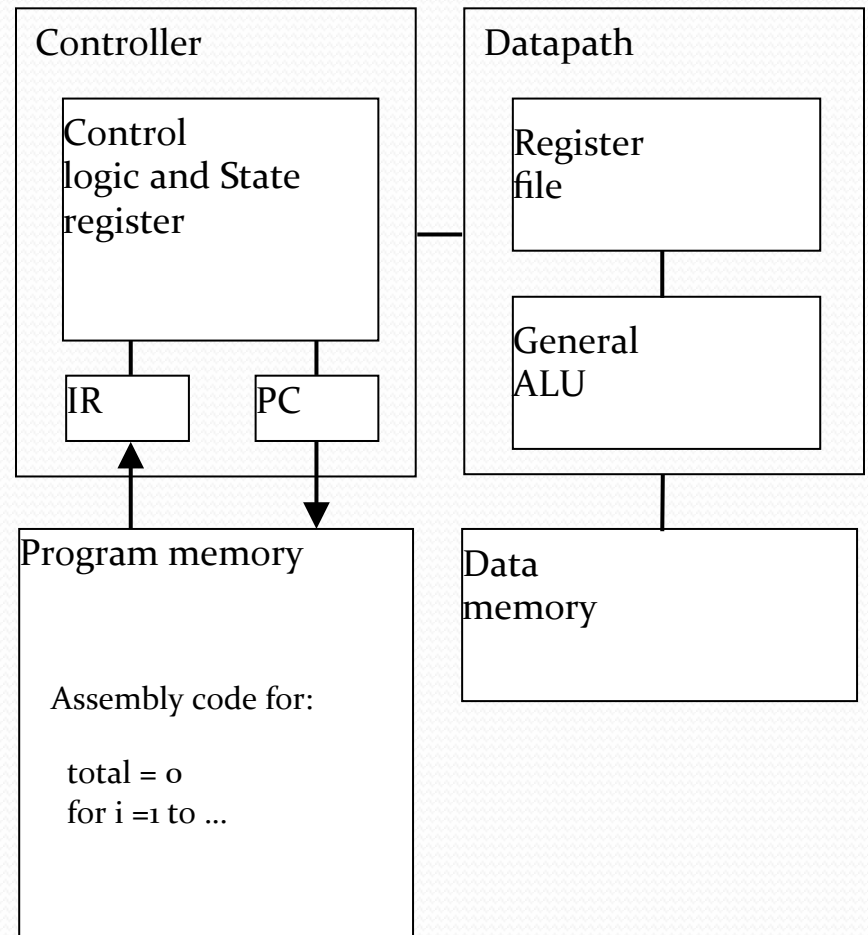
Application-specific



Single-purpose (“hardware”)

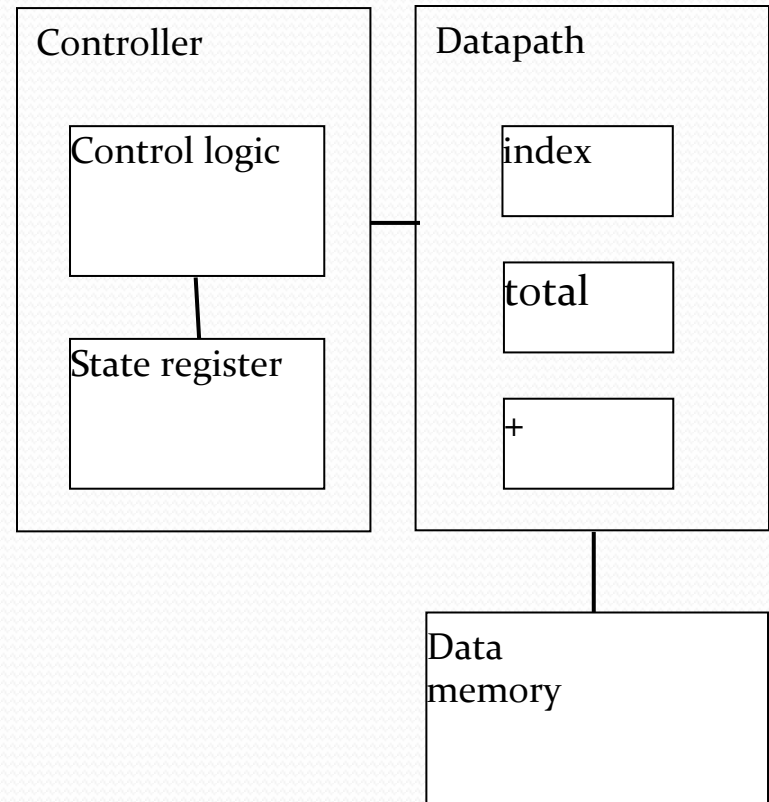
# Processeur: générique

- Processeur programmable
- Caractéristiques
  - Programme traité comme une donnée
  - Programme dans la mémoire
  - ALU générale + Registres de données
- Avantages:
  - TTM très court + NRE très bas
  - Très flexible
- PC => Intel domine
- Dans les SE une variété plus large d'architectures



# Processeur: Monofonction

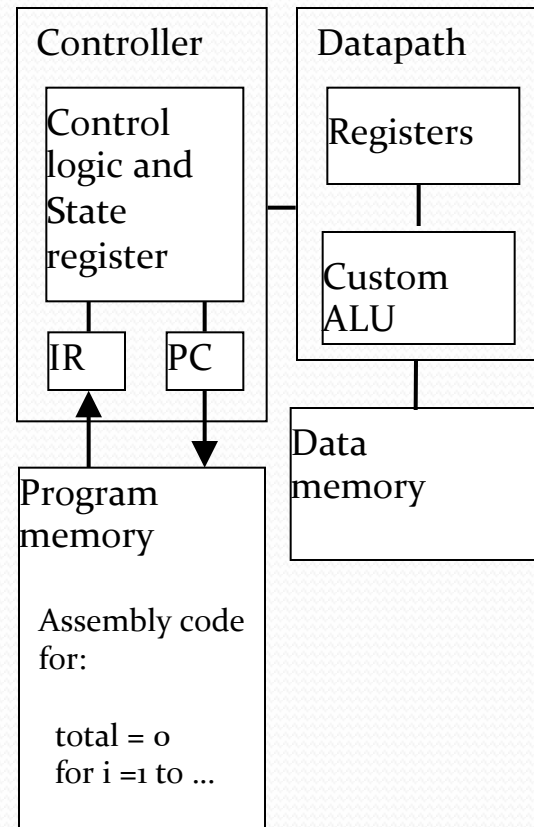
- Circuit numérique qui exécute une seule fonction
  - Coprocesseur, contrôleurs
- Caractéristiques
  - Contient seulement les composants requis pour l'exécution du programme
  - Pas de mémoire programme
- Avantages:
  - Rapidité
  - Economie d'énergie
  - Taille réduite



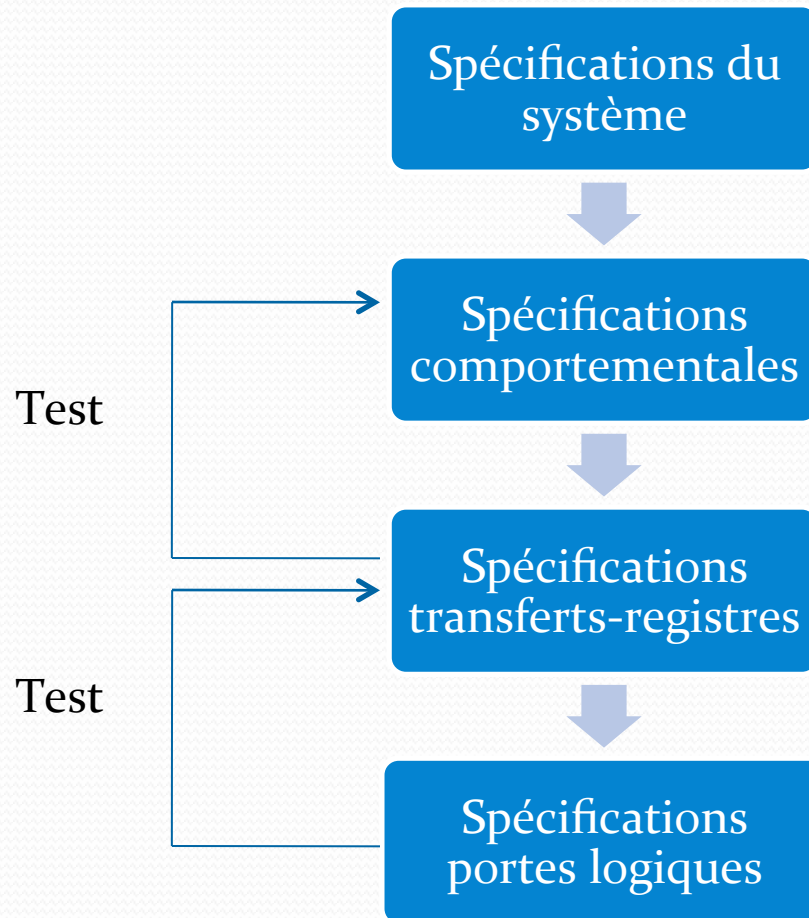


# Processeur: Spécialisé

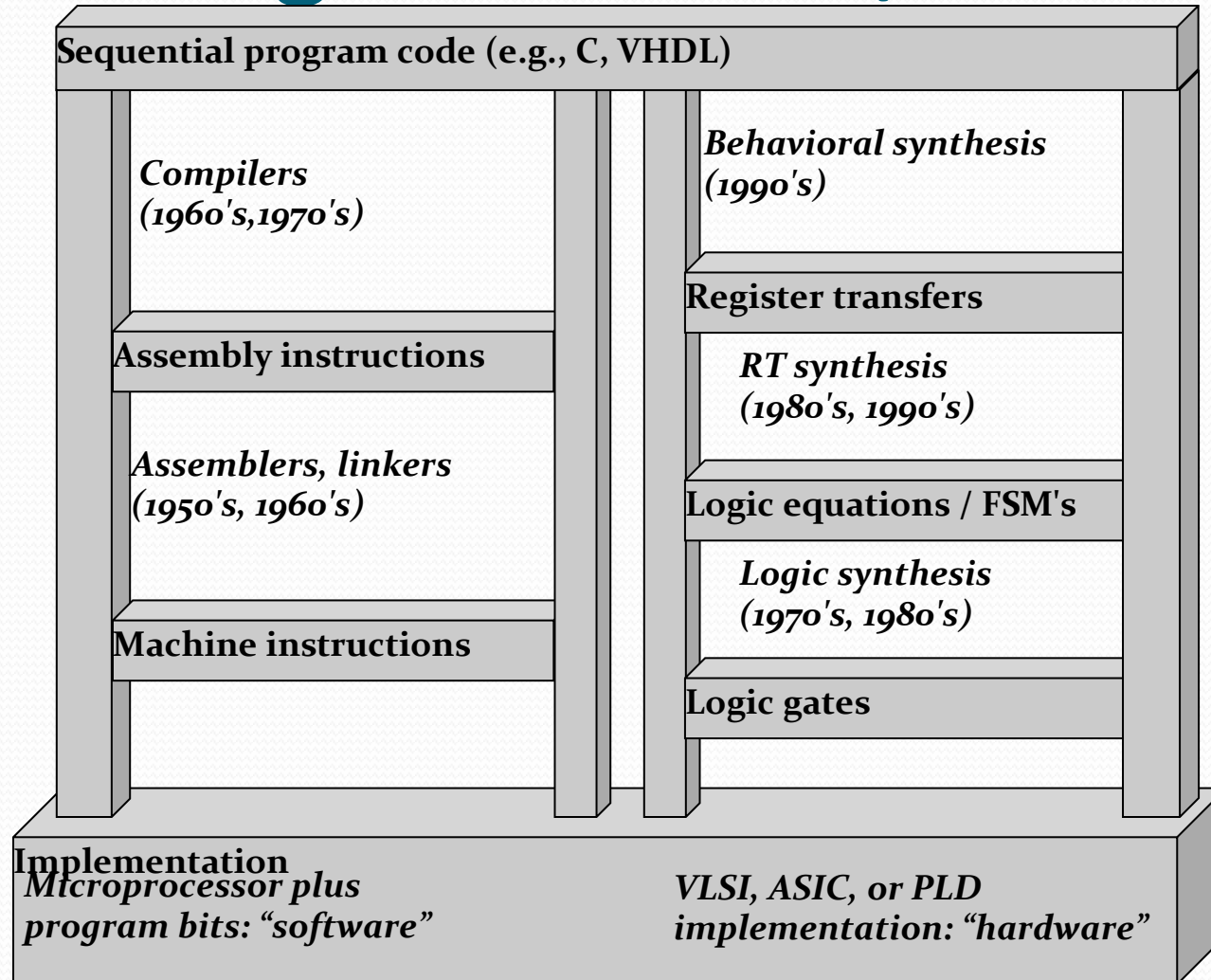
- Processeur programmable spécialisé pour un domaine d'applications (ex. GPU)
- Caractéristiques:
  - Mémoire programme
  - Très forte optimisation du datapath
  - Unités fonctionnelles spécialisées
- Avantages:
  - Une certaine flexibilité, performance, taille et énergie



# Processus de conception



# Co-design software/hardware



# Conclusion

- Système embarqué
- Métriques différentes
- Savoir faire et compétences soft. Et hard.
- Technologies
- Le co-design et rapprochement soft et hard.

# Référence

- [1] Embedded System Design, Vahid/Givargis
  - excellente introduction

# La suite...

- Dans ce cours:
  - on s'intéresse aux aspects logiciels
- Mon téléphone portable est-il un système embarqué?
  - Ensemble de composants embarqués + OS
- J'ai une formidable plateforme pour programmer de nouvelles générations d'applications « embarquées et communicantes » !
- iOS