

Informatique Embarquée

M2 / 2014

Qui est-ce ?



Objectifs

- Comprendre :
 - Ce que sont les systèmes embarqués,
 - ▶ Définition, « construction », contraintes,...
 - La notion de temps-réel, ordonnancement,
 - Architecture d'un processeur,
 - Architecture mémoire,
 - Systèmes de fichiers, interfaces graphiques...
 - Les notions : OS, micro-noyaux, OS temps-réel...
 - Virtualisation pour systèmes embarqués

Objectifs

- « Maitriser » :
 - Le langage C,
 - Chaîne de production, génération croisée,
 - Programmation système,
 - ▶ Création de threads,
 - ▶ Synchronisation
 - ▶ Mesures de temps,
 - « Production de systèmes »
 - Occupation mémoire, gestion mémoire
 - ...

Objectifs

- Progresser
 - Maîtrise du temps
 - Méthode
 - ▶ Makefile, gestion de versions,...
 - ▶ Écrire du « bon code »
 - ▶ Tester (automatiquement)
 - ▶ Documenter
 - ▶ Recommencer
 - Travailler ensemble
 - S'exprimer...



Informations Relatives au Cours

- Sur le site:

<http://didel.script.univ-paris-diderot.fr/index.php?category=056>

>Sciences>UFR Informatique>M2T2InfoEmb

Vous trouverez:

- Le calendrier des cours
- Les supports de cours mis à disposition avant les cours
- Les informations de dernière minute
- Un forum, un wiki, un chat...

Modalités de Contrôle

- Contrôle continu : TP
 - Compte-rendu de TP et / ou exercices à domicile
 - ▶ 8 (6 min.) exercices proposés durant la session.
 - Individuel et/ou groupes (2 max)
 - Les 6 (5) meilleures notes sont prises en compte
 - ▶ 3 des notes doivent être obtenues sur un travail individuel
 - Les points au dessus de 10 des TP rendus en plus du minima (6 ou 5) améliorent votre note de TP (max : 20)
 - Ex : moyenne sur 6 : 12, TP en plus 16
 - ▶ $(6 \times 12) + (16 - 10) / 6 \Rightarrow 13$

Modalités de Contrôle

- Contrôle continu : TP suite
 - Attention possibilité de petits QCM en début de cours
 - ▶ Absence => 0, 1 point / question
 - ▶ => 1 note / 20 individuelle (par règle de trois)
 - ▶ Par ex 4 meilleurs QCM / 7
- Note de TP / QCM :
 - TP individuels : $TPI = TP_{I1} + TP_{I2} + TP_{I3}$
 - TP individuels ou collectifs: $TPCI = TP_{CI1} + TP_{CI2} + TP_{CI3}$
 - TP suppl. : $TPS = \max(TP_{CI7} - 10, 0) + \max(TP_{CI8} - 10, 0)$
 - $TP = (TPI + TPCI + TPS + QCM) / 7$



Modalités de Contrôle

- Contrôle continu : fin
 - La présence en cours et aux TP constitue une note du CC (10% du CC => 5% note finale)
 - $CC = (9 \times TP + 1 \times Pres.) / 10$
- Contrôle final: examen 3H.
- Note = 1/2 Contrôle Continu + 1/2 Contrôle final



Contrôle continu : Présence



- Après l'introduction : 10 cours et 10 TP
- Feuilles de présence à signer en cours ET en TP
 - Une absence = 1 signature manquante
 - Sauf excuse raisonnable (de préférence avant par courriel))
- 20 présences => 20/20
- 1 à 2 absences : -1 point / absence => 19 ou 18
- 3 à 4 absences : -1,5 points/absence => 16,5 ou 15
- 5 à 6 absences : -2 points/absence => 13 ou 11
- 7 à 8 absences : -2,5points/absence=> 8,5 ou 6
- 9 absences : 3/20
- 10 absences et plus : 0



Plan Général (indicatif)

- Introduction
- Distributions Linux pour systèmes embarqués
- Noyaux, Micro-Noyaux, OS Embarqués
 - Linux, Chorus, eCos...
- Gestion Mémoire
- Temps, Timers, Ordonnancement
- Architecture processeur (ARM / Intel)
- Virtualisation pour systèmes embarqués
- Disponibilité pour systèmes embarqués

Références / Ressources

- Programming Under ChorusOS; Jean-Marie Rifflet
<http://www.pps.jussieu.fr/~rifflet/PUBLICATIONS/book4.html>
- Linux embarqué; Pierre Ficheux, Ed. Eyrolles
<http://www.eyrolles.com/Informatique/Livre/9782212110241/>
- Real-Time Concepts for Embedded Systems; Qing Li and Carolyn Yao
- Metodologie di Progettazione Hardware – Software; Luca Benini
<http://www-micrel.deis.unibo.it/MPHS/Materiali05.html>
- Bienvenue dans l'univers des systèmes embarqués
<http://www.enseirb.fr/~kadionik/embedded/embedded.html>

Références / Ressources

- Embedded System Design: A Unified Hardware/Software Introduction
 - Frank Vahid and Tony Givargis; Ed John Wiley & Sons
<http://esd.cs.ucr.edu/>
- Site de Mme Simonot-Lion
http://www.loria.fr/~simonot/#_Activit%C3%A9s_d%E2%80%99enseignement
- Site de Gilles Grimaud
<http://www.lifl.fr/~grimaud/CoursSEE/HomePage>
- Trinux
<http://www.io.com/~mdfranz/rtos/>
- Et pleins d'autres que vous ne manquerez pas de découvrir.

Introduction: Plan

- Généralités
- Caractéristiques
- Domaines d'application
- Choix de conception
- Environnement logiciels embarqués

Tentative de définition

- Système embarqué: (*embedded systems*)
 - Un système de traitement de l'information inclus dans un produit plus "large", dont la fonction essentielle n'est pas le traitement de l'information
- La raison qui motive l'achat d'un [*produit contenant un*] système embarqué n'est pas le traitement de l'information
- Il arrive que l'on parle de systèmes enfouis



I
N
V
A
S
I
O
N

M
I
D
D
L
E
W
A
R
E

Le middleware est partout.
Vous voyez ?

IBM DB2

Légendes

- 1. Visite virtuelle de la destination touristique.
- 2. Réservation du vol sur une compagnie partenaire.
- 3. Offre simultanée de services annexes.
- 4. Analyse dynamique des horaires.
- 5. Décollage immédiat du chiffre d'affaires.

LE MIDDLEWARE, L'OFFRE LOGICIELLE D'IBM.

- Les solutions de gestion de l'information DB2, associées aux systèmes de gestion de bases de données DB2 et Informix, sont parmi les plus complètes du marché. Fondées sur des standards ouverts, elles permettent l'accès à des contenus hétérogènes. On Demand. Ces solutions intègrent l'information, dopant la productivité et restent faciles à gérer. Vous avez ainsi toutes les cartes en main pour prendre vos décisions.

Le middleware pour le monde du business On Demand. Visitez ibm.com/db2/infomgmt/fr

* Logiciels d'infrastructure. À la demande. IBM, le logo IBM, le logo ON DEMAND BUSINESS, DB2 et Informix sont des marques d'International Business Machines Corporation aux États-Unis, dans d'autres pays ou les deux. Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers. © 2005 IBM Corporation. Tous droits réservés.

Et celle des systèmes embarqués?



2 Millions de
systèmes
embarqués
ici?



Exercice 1

- Nommer au moins 10 systèmes embarqués

Ils sont partout!

- En 2004
 - marché des systèmes embarqués est devenu supérieur au marché des clients/ serveurs et PC
- En 1996
 - Selon le New-York Times, l'américain moyen était en "contact" avec 60 processeurs / jour
- En 2004
 - On estimait ce nombre à 100!

Ils sont polymorphes

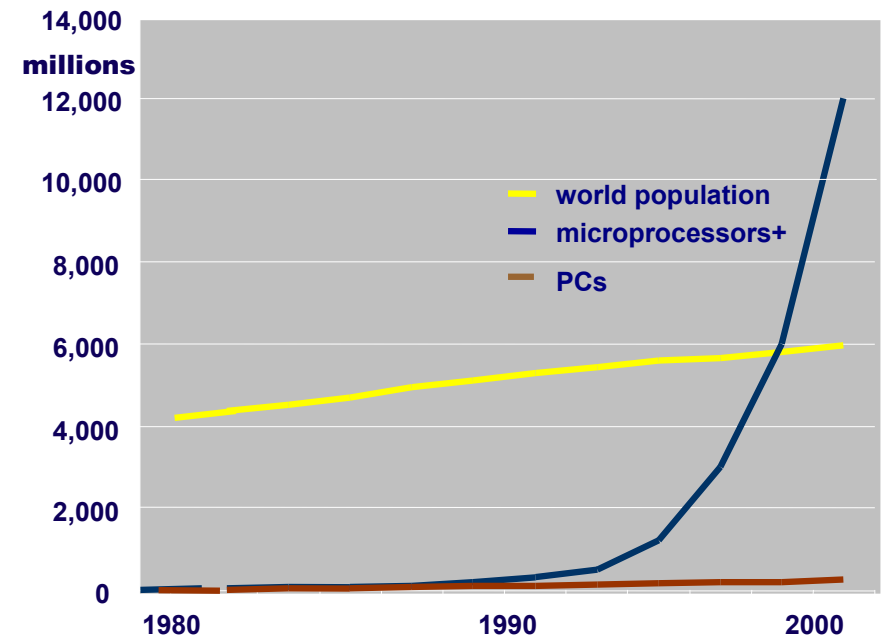


Tentative de définition

- Système embarqué:
 - Un système de traitement de l'information inclus dans un produit plus "large", dont la fonction essentielle n'est pas le traitement de l'information
- La raison qui motive l'achat d'un *[produit contenant un]* système embarqué n'est pas le traitement de l'information

Ils vont croître et prospérer

- Le nombre de "systèmes embarqués" devrait atteindre 16 milliards en 2010
- L'électronique interviendra pour 40 % de la valeur d'une automobile en 2010
- Un téléphone haut de gamme (smart phone) peut contenir des millions de lignes de code
- Croissance > 10% /an



S'immiscer partout

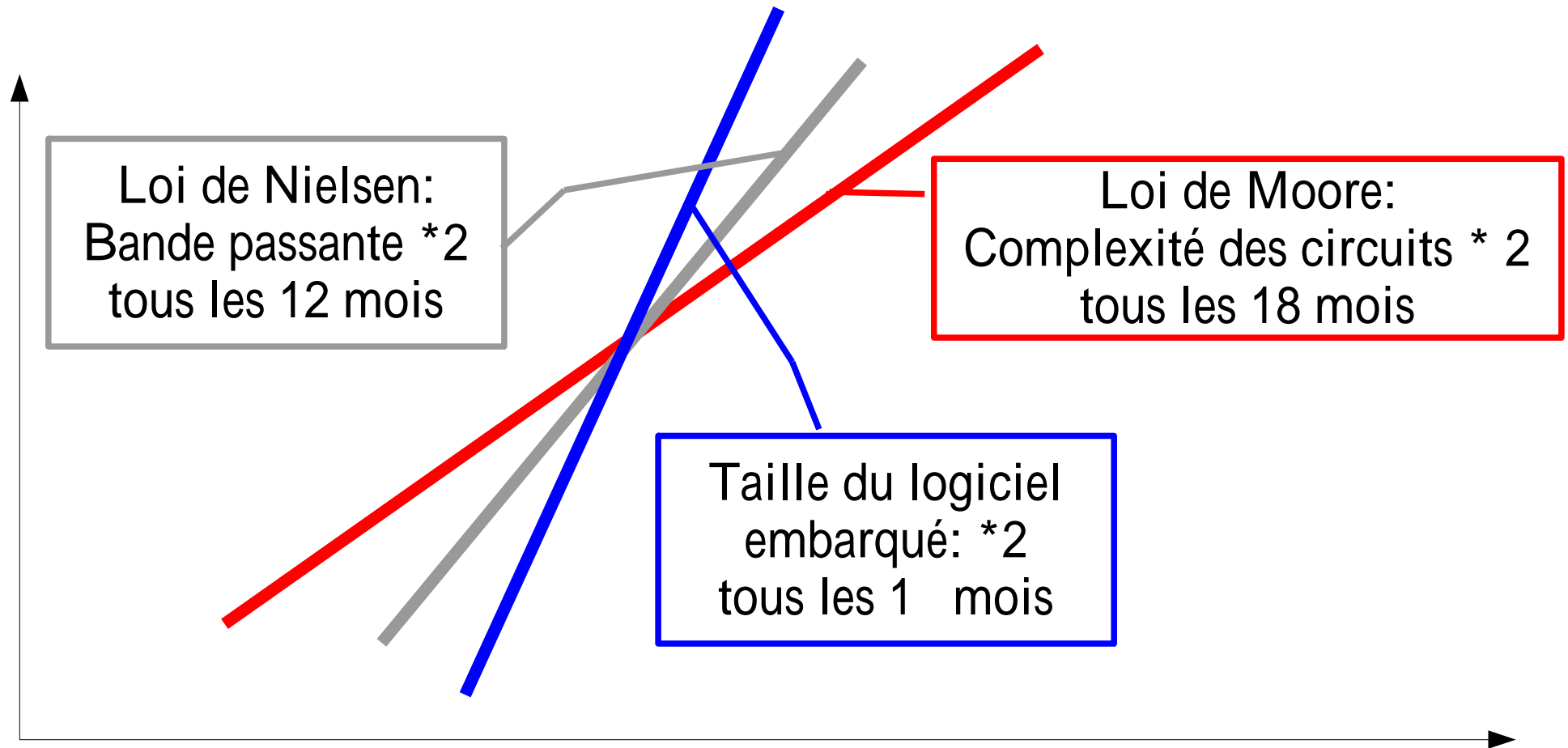
- Intelligence Ambiante (huh?)
 - Des capteurs (forcément intelligents)
 - ▶ consommation μW , performance KB/s
 - **Connectés** via des Concentrateurs sans fil
 - ▶ consommation mW, performance MB/s
 - A des serveurs réseau fixes
 - ▶ consommation W, performance GB/s

Problèmes

As long as there were no machines,
programming was no problem at all;
when we had a few weak computers,
programming became a mild problem,
and now that we have gigantic computers,
programming has become a gigantic problem.

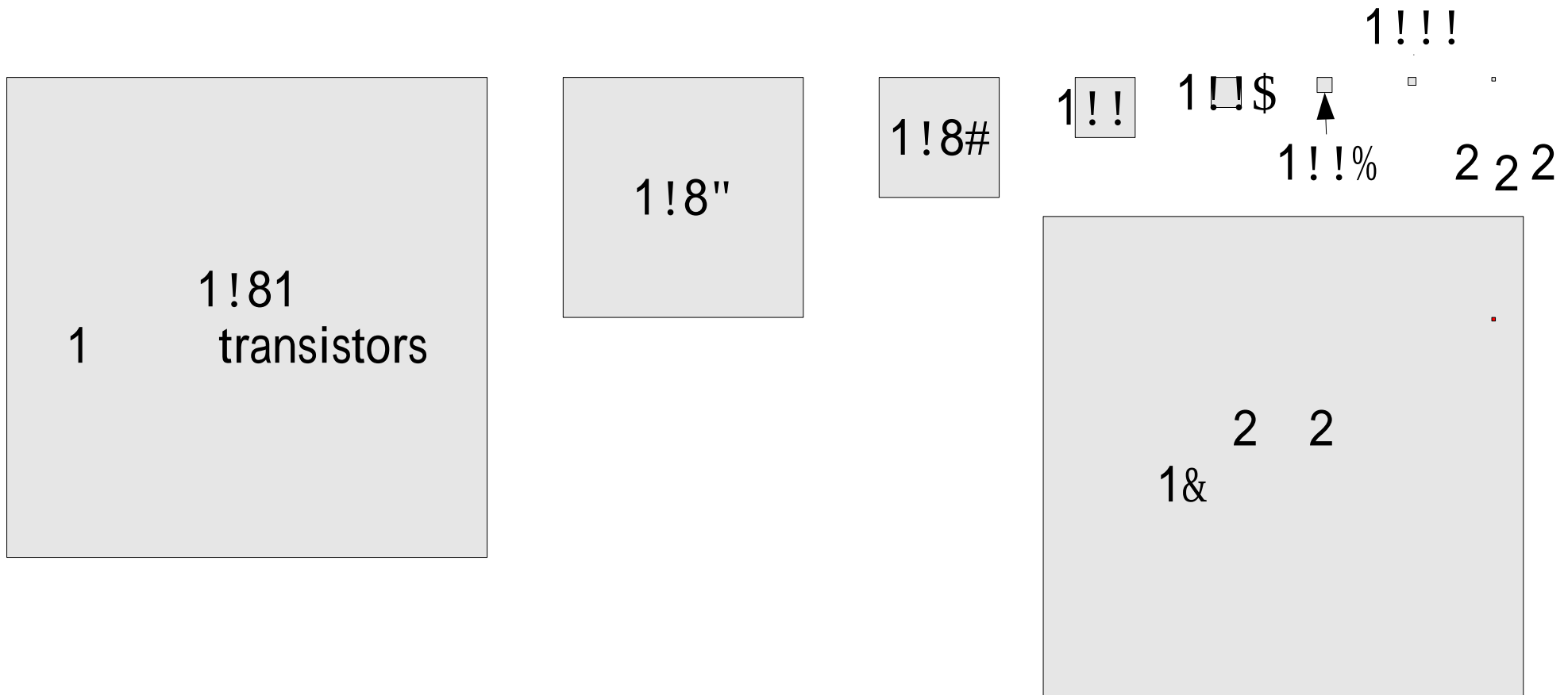
—Edsger W. Dijkstra,
ACM Turing Award Lecture, 1972 (!)

Systemes embarqués: problèmes



- Selon les sources, les taux changent (sauf loi de Moore), mais le principe reste le même

Loi de Moore



- Doublement tous les 18 mois

Évolution

- Vitesse du processeur: doublement tous les ans depuis 85,
 - x 100 sur derniers dix ans
 - Mais c'est fini! => nombre de coeurs!
- Capacité mémoire: doublement tous les 2 ans depuis 96,
 - x 64 sur derniers dix ans
- Capacité disque: doublement tous les ans depuis 97
 - x 250 sur derniers dix ans

Loi de Moore: rupture

- A [récemment] changé de forme
- Au lieu de conduire à une augmentation de la fréquence des horloges, elle conduit maintenant à une augmentation des « cœurs » sur une puce.
- Processeurs multi-cœurs
 - Problèmes de parallélisme.. systèmes SMP...
- Fréquence limitée, consommation réduite

Introduction: Plan

- Généralités
- **Caractéristiques**
- Domaines d'application
- Choix de conception
- Environnement Logiciels embarqués

Caractéristiques systèmes embarqués

- Pouvoir "en dépendre"
 - **Fiabilité**: probabilité que le système fonctionne correctement à t , s'il fonctionnait à $t=0$
 - **Maintenance**: probabilité que le système fonctionne correctement Δt après une erreur
 - **Sûreté**: Pas de conséquence dramatique
 - **Sécurité**: communication authentifiée et confidentielle

Caractéristiques systèmes embarqués

- "Dépendabilité"
 - Définition(s)
 - ▶ Voir le document « laprie_taxonomy.pdf » sur le site du cours.
 - Basée sur des hypothèses de charge
 - Doit être considérée dès le début, pas après coup

Caractéristiques systèmes embarqués

- Doit être efficace:
 - Consommation énergétique
 - Taille de code
 - Réactif
 - Léger
 - Pas cher
- Dédié (exclusivement à un domaine applicatif)
- Interface de communication spécifique

Caractéristiques systèmes embarqués

- Contraintes temporelles (temps-réel)
 - Un système temps-réel doit réagir aux stimuli de son environnement dans un délai temporel imposé par l'environnement, l'application
 - ▶ Ex: système de freinage, Contrôle automatisme, procédés industriels...
 - Pour un système temps-réel, il ne suffit pas de fournir des résultats corrects, il faut aussi les fournir en temps et en heure.
 - ▶ Résultats corrects au sens temporel
 - ▶ Des résultats corrects mais tardifs sont faux.

Caractéristiques: Contraintes temps-réel

- Dures (Hard real-time)
 - Une contrainte est dite "dure", si son non respect résulterait en une catastrophe (Kopetz 1997)
- Lâches (Soft real-time)
 - Toutes les autres contraintes temporelles

Exercice 2

- Un décodeur de télévision est-il temps-réel?
- Un robot sur une chaîne de montage automobile est-il temps-réel?
- La déclaration des revenus sur Internet est-elle temps-réel?
- Un jeu vidéo est-il temps-réel?
- Une application de réservation de billets d'avion est-elle temps-réel?
- Un téléphone mobile est-il temps-réel?

Caractéristiques: Contraintes temps-réel

- Dures (Hard real-time)
 - Une contrainte est dite "dure", si son non respect résulterait en une catastrophe (Kopetz 1997)
 - Ex: Freinage train, Contrôle centrale nucléaire...
 - Temps de réponse garanti, pas de mesure statistiques!
- Lâches (Soft real-time)
 - Toutes les autres contraintes temporelles
 - Ex: systèmes téléphoniques, multimédia...

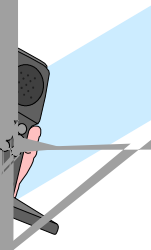
Caractéristiques: Environnement

- Possiblement "immergés" dans des environnements physiques "hostiles"
- Fréquemment couplés à l'environnement physique via des capteurs et des "actionneurs"
- Systèmes hybrides: numériques et analogiques
- Systèmes réactifs:
 - ▶ interaction continue à un rythme déterminé par l'environnement
 - ▶ Comportement basé sur état et stimuli
 - ▶ Modèle "automate" approprié



para

le d



Introduction: Plan

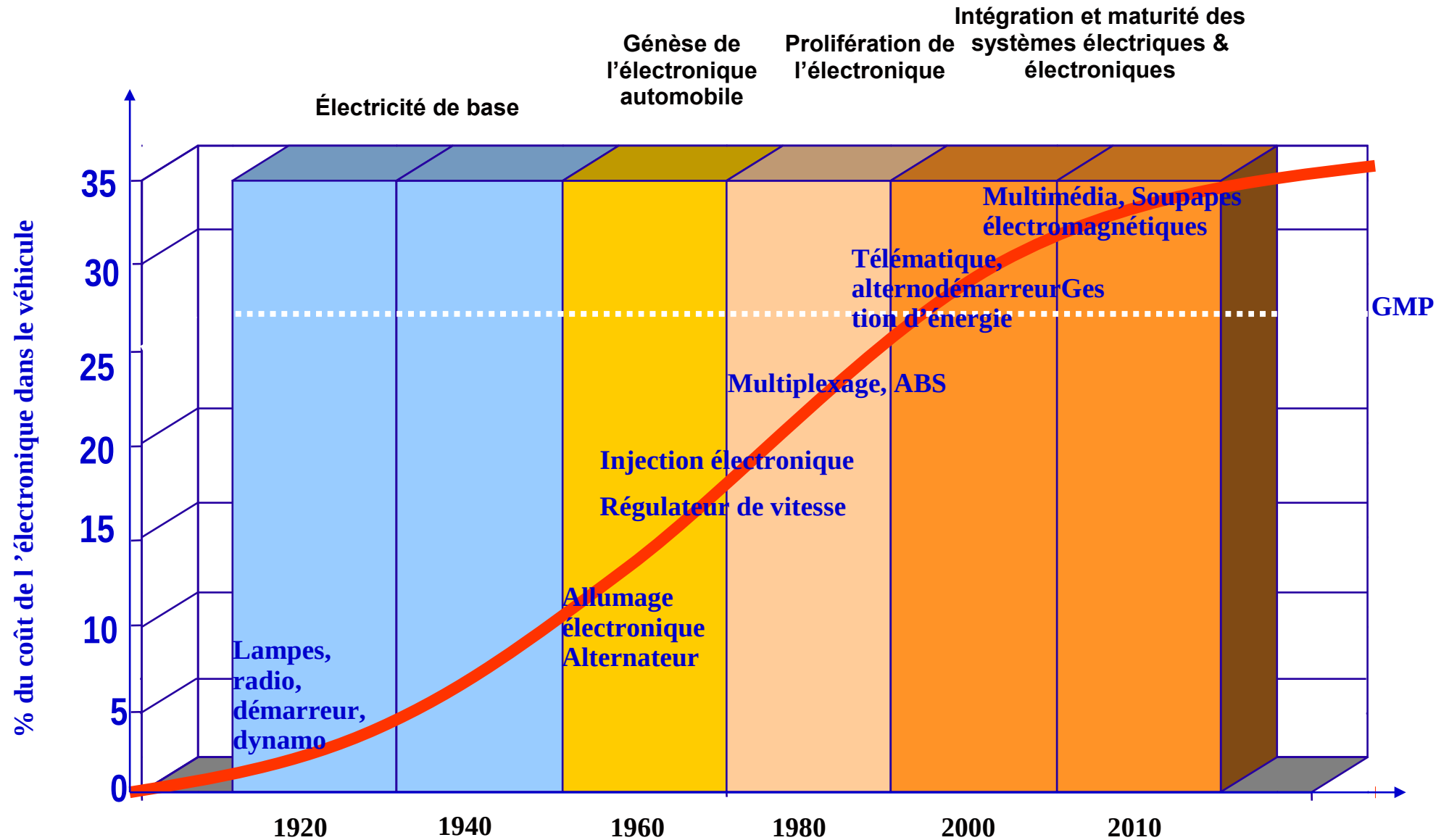
- Généralités
- Caractéristiques
- Domaines d'application
- Choix de conception
- Environnement Logiciels embarqués

Domaines d'applications

- Automobile
 - de 20 à 80 processeurs, (BMW > 100!!)
 - 2 Mo de code dans la 607
 - 90% de la R&D chez DC
 - Châssis, Moteur,
 - Corps de voiture (lumière, essuie-glace...),
 - "Télématique"
 - Réalité augmentée?

Automobile: Evolution

Extrait de la présentation de Joseph Beretta / PSA - 16 et 17 Juin 2003 – <http://www.systemes-critiques.org/SECC/>



Domaines d'applications

- Aviation, Transport ferroviaire
- Télécommunication
- Systèmes médicaux
- Militaires
- Authentification
- Électronique domestique
- Procédés industriels
- Immeubles intelligents

Domaines d'applications

- Électronique domestique
- Procédés industriels
- Immeubles intelligents
- Robotique

Introduction: Plan

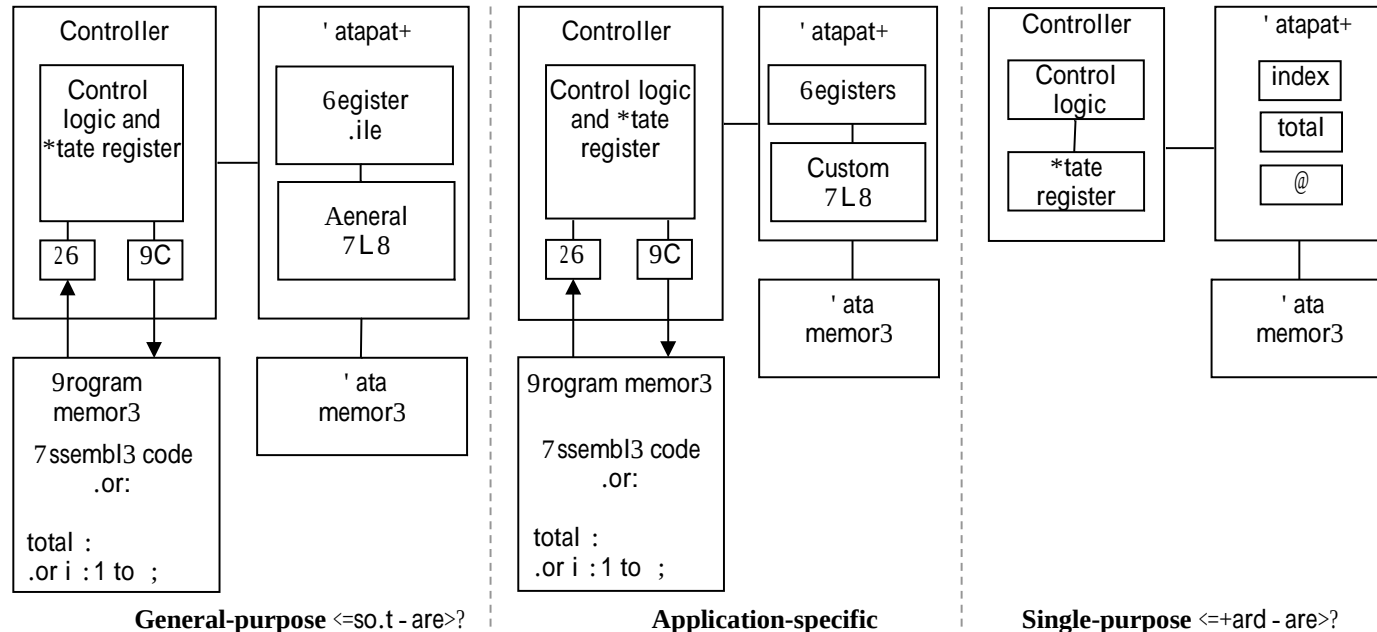
- Généralités
- Caractéristiques
- Domaines d'application
- **Choix de conception**
- Environnement Logiciels embarqués

Choix de conceptions

- Processeurs généralistes, spécialisés ou dédiés
- Si Processeurs spécialisés ou dédiés,
 - Choix de réalisation du circuit

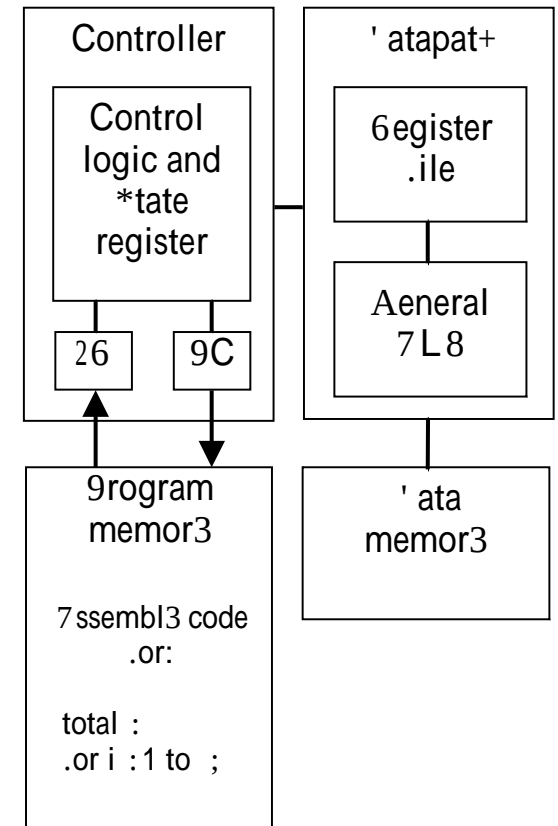
Choix de processeur

- Plusieurs possibilités pour fournir un service
 - processeur généraliste
 - Spécifique pour application
 - Dédié



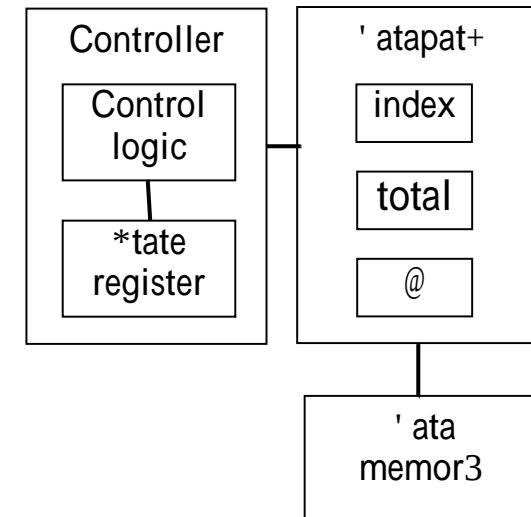
Processeurs généralistes

- Microprocesseurs programmables
- Fonctions:
 - Mémoire programmable
 - Chemin de données avec banc de registres et ALU généraliste
- Bénéfices
 - Mise sur le marché rapide, faibles coûts de développements
 - Très flexible
- Pentium, ARM, MiPS, PPC, SH...



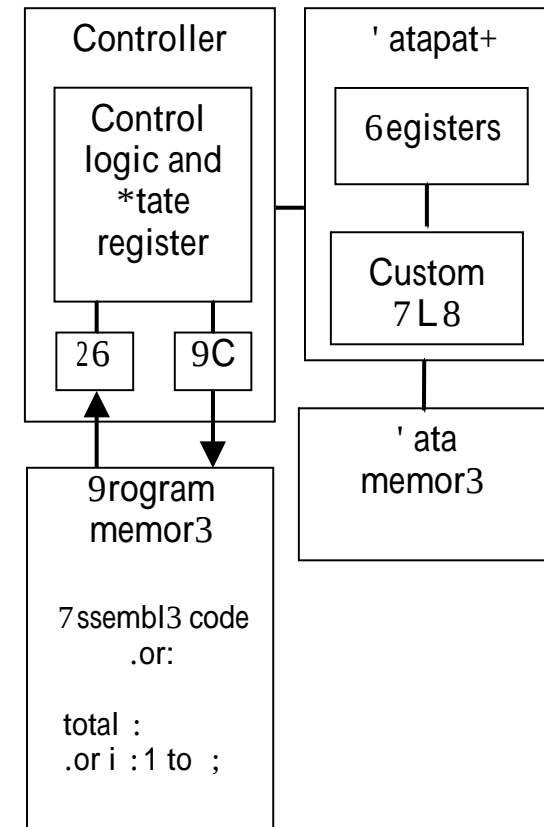
Processeurs dédiés

- Circuit conçu pour exécuter UN programme unique:
 - coprocesseur, accélérateur, périphérique...
- Fonctions:
 - Pas de mémoire programmable
 - Contient les seuls composants nécessaires pour exécuter ce programme
- Bénéfices
 - Rapide
 - Faible encombrement, petite taille
 - Faible consommation



Processeurs Spécifiques Application

- Processeurs programmables optimisés pour une classe d'applications ayant des caractéristiques communes
 - Compromis entre les 2 "extrêmes" précédents
- Fonctions:
 - Mémoire programmable
 - Chemin de données optimisés
 - Unités fonctionnelles (ALU) spécialisées
- Bénéfices
 - Relativement flexible
 - Bonnes performances, tailles, mise sur le marché



Co-Design

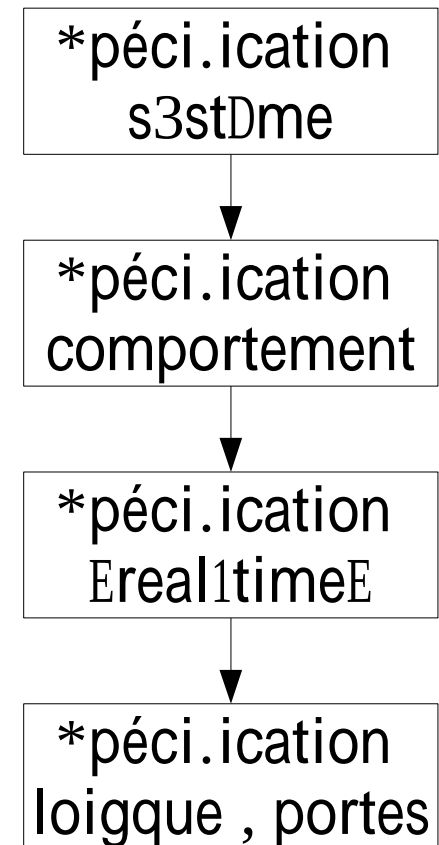
- Très souvent, on conçoit le matériel et le logiciel simultanément
 - Matériel spécialisé: ASIC, FPGA et SoC... MPSoC
 - ASIC: Application Specific Integrated Circuit,
 - FPGA: Field-Programmable Gate Array,
 - SoC: System on Chip
 - MP SoC: Multi-Processor System on Chip

Choix de technologie

- VLSI: conception du circuit,
 - très cher et très long mais très efficace (consommation et rapidité)
- ASIC: conception de quelques couches du circuit
 - Bonne performances, taille
 - moins cher mais encore (trop) long
- FPGA: couches du circuit disponibles, connexion entre portes à établir
 - disponibilité immédiate, cher à l'unité, gros, grosse consommation, lent

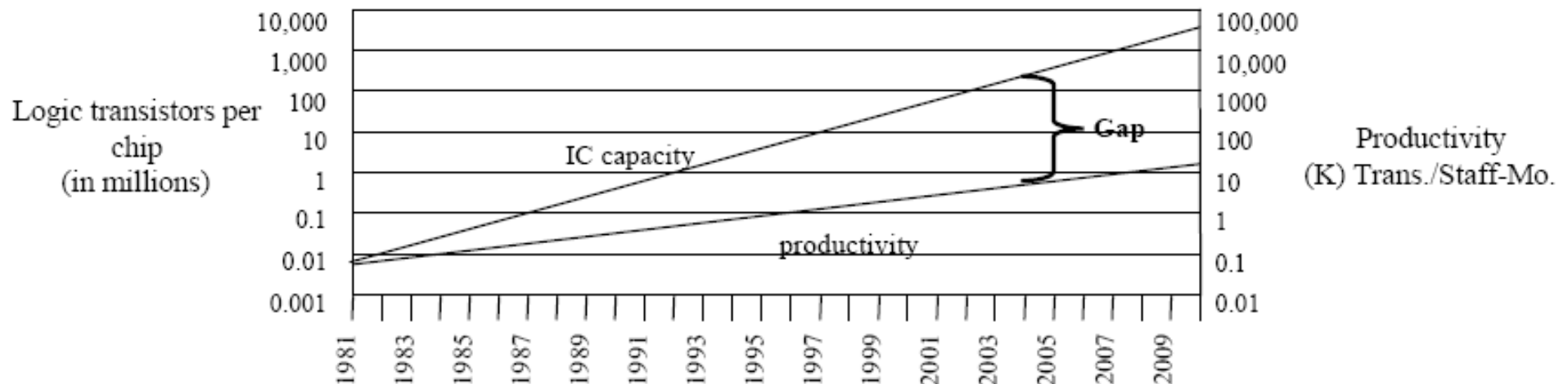
Conception

- Utilisation,
 - modèles
 - outils
 - simulateurs
 - Langages conception
 - ▶ VHDL, SystemC
- Pour raccourcir le cycle
- Logiciel dépendant du matériel



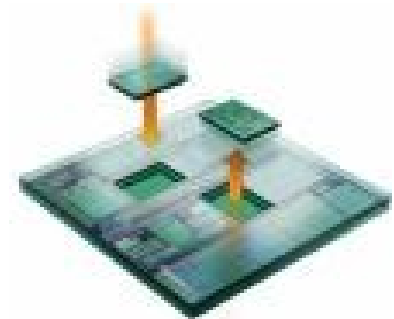
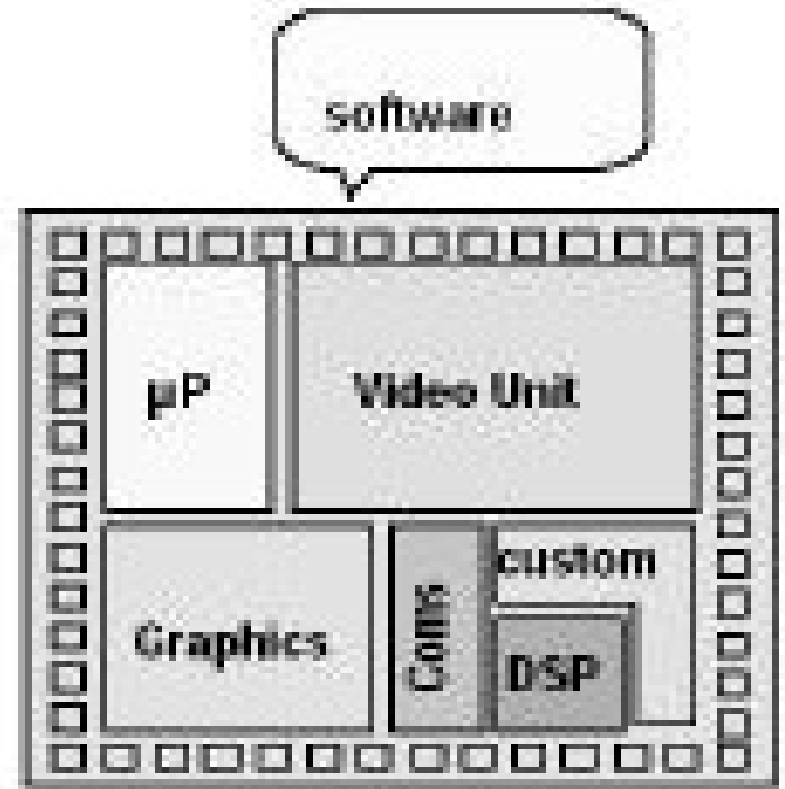
Complexité croissante

- En 81, un "chip" nécessitait 100 homme x mois
 - 10 000 Transistors, 100 transistors/mois
- En 2002: 30 000 homme x mois
 - 150 000 000 Transistors, 5000 transistors/mois
- Coûts de conception: de 1M\$ à 300M\$



SoC

- Conception des SoC
 - par réutilisation de "blocs"
 - ▶ "blocs d'IP"
 - ▶ type boîte noire
 - permet de raccourcir la conception du système
 - Vérification système et non plus portes logiques
- Importance croissante du logiciel



Introduction: Plan

- Généralités
- Caractéristiques
- Domaines d'application
- Choix de conception
- Environnement Logiciels embarqués

Logiciel Embarqué

- 70% du coût de développement de systèmes complexes électroniques est dû au développement du logiciel
 - [A. Sangiovanni-Vincentelli, 1999]
- Dans de nombreux produits électroniques grand-public, le volume de logiciel double tous les 2 ans

Quelques Problèmes

- Le problème de l'homme mois...
- Comment décrire le comportement attendu de systèmes complexes?
- Comment valider des spécifications?
- Comment passer efficacement des spécifications à la réalisation?
- Les programmeurs se soucient-ils jamais de la consommation électrique?

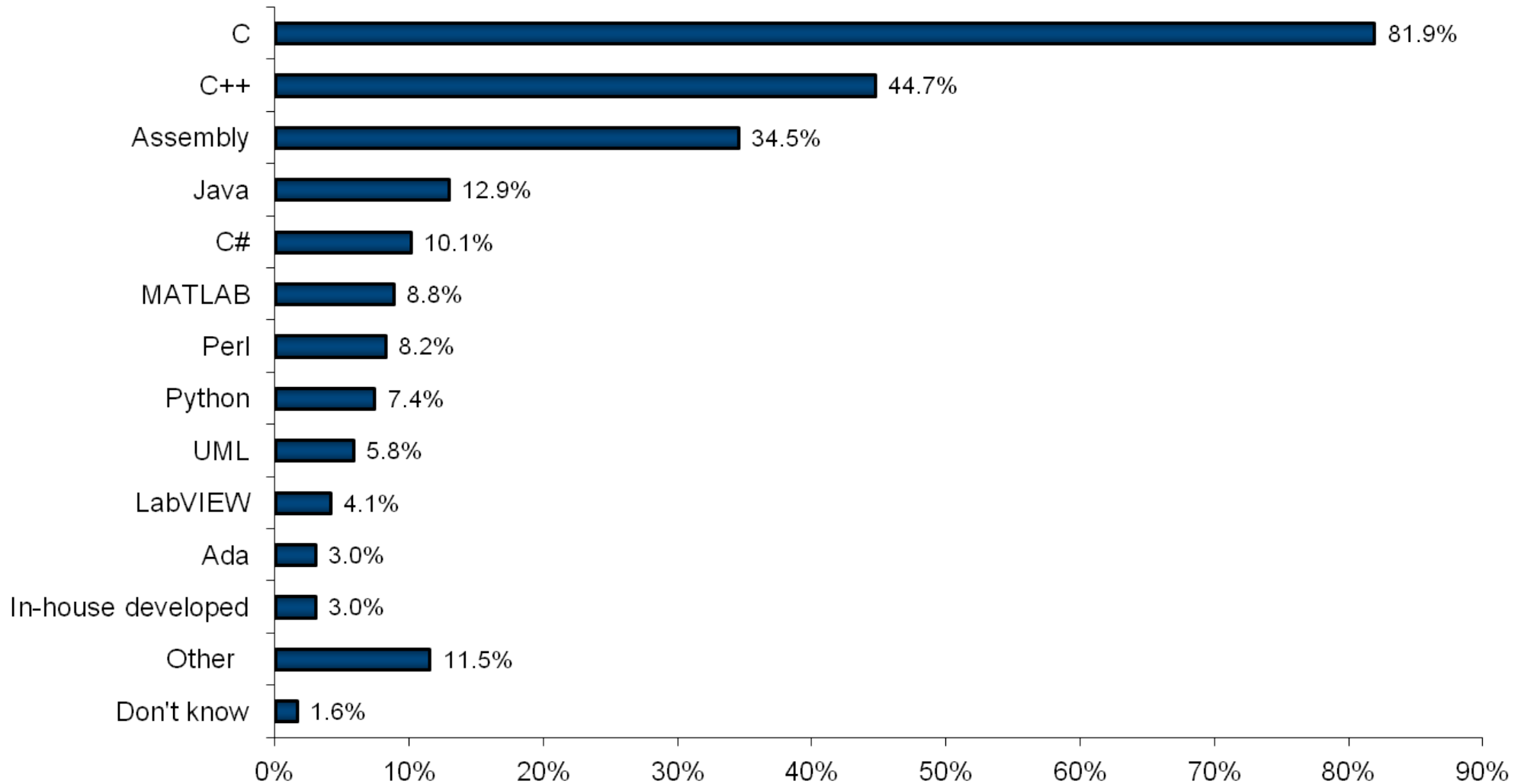
Problèmes

- Comment s'assurer que les contraintes temporelles sont respectées?
- Quels langages de programmation sont-ils appropriés?
- Comment valide-t-on des systèmes embarqués temps-réel:
 - Grands volumes de données,
 - Systèmes critiques
 - ▶ Ex: Ariane, Mars PathFinder, plus "prosaïquement" systèmes médicaux, systèmes de transport...

Logiciel Embarqué

- Par la suite, on s'intéressera essentiellement aux problèmes liés aux logiciels embarqués
 - Architecture matérielle dans l'embarqué
 - Développement de logiciel pour l'embarqué
 - Noyaux pour l'embarqué et le temps-réel
- Dans les systèmes embarqués le langage C/C++ est prédominant
- http://blog.vdcresearch.com/embedded_sw/2010/09/what-languages-do-you-use-to-develop-software.html
- <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Langages utilisés



Note: Percentages sum to over 100% due to multiple responses.

B ' C surCe3 2 1 (Coir le lien page précédente)

Acteurs et Environnement

- Quelques fournisseurs de processeurs
 - ARM, Intel, MiPS, Freescale
 - IBM/Sony Cell
 - Texas Instrument
 - Fabricants de cartes à puces
- Fournisseurs de cartes et périphériques...

Acteurs et Environnement

- Quelques Noyaux
 - VxWorks, Nucleus, PsoS, VRTX, OSE
 - QNX, Chorus/C5, L4
 - ThreadX, eCOS
 - Linux , μ Linux, MontaVista Linux, Lynux
 - Itron, WindowsCE, Symbian OS, PalmOS,
 - Osek
 - Beaucoup plus encore de systèmes "propriétaires"!

Autres logiciels

- Pilotes de périphériques,
- Interfaces graphiques
- Protocoles réseaux
- Java embarqué J2ME, JavaCard
- Codecs
- Systèmes de gestion de fichiers
- Bases de données...

Organisation/Consortiums

- Embedded Linux Consortium
 - <http://www.embedded-linux.org/>
- Consumer Electronics Linux Forum
 - <http://www.celinuxforum.org/>
- Open Group Real-Time
 - <http://www.opengroup.org/rtforum/>
- Plus spécifiques:
 - Communications (USB,...)
 - PICMG <http://www.picmg.org/>



Ce qu'il faut retenir

- Définition système embarqué
- Définition système temps-réel
- Diversité des applications et des contraintes
- Sur le plan personnel, il vous est demandé un travail important pour les TP et le contrôle continu