École Nationale des Sciences de l'Informatique

Chapitre 1

Introduction

Module : Introduction aux systèmes embarqués
Niveau : II2 – Tronc commun

AU : 2011/2012

Plan du cours

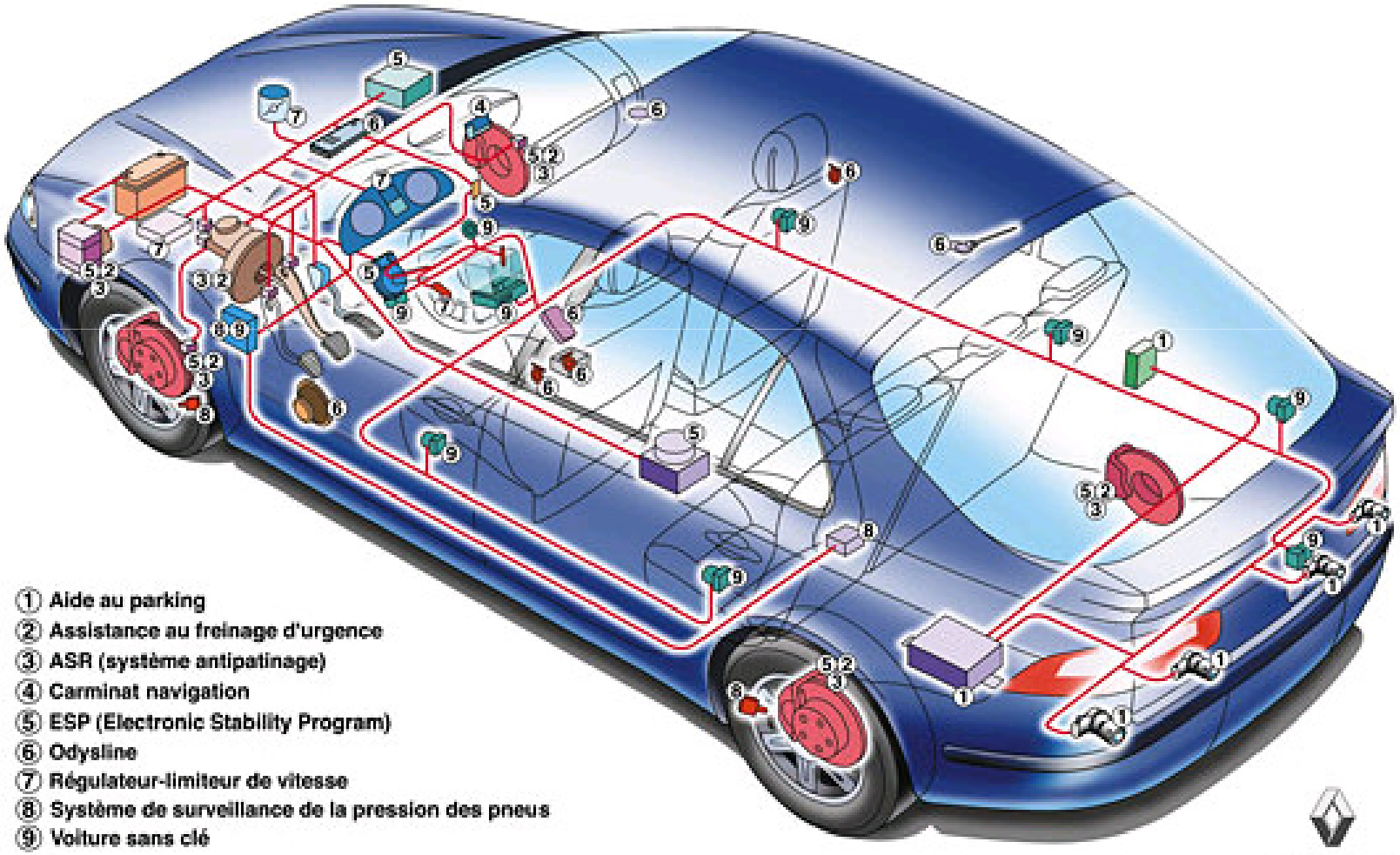
- Définition et généralités
- Conception des systèmes embarqués
- Cible logiciel
- Cible matériel
- Cible mixte

Pourquoi les systèmes embarqués?

«Les systèmes embarqués nous entourent et nous envahissent littéralement, fidèles au poste et prêts à nous rendre service. Ils sont donc partout, discrets, efficaces dédiés à ce à quoi ils sont destinés. Omniprésents, ils le sont déjà et le seront de plus en plus»

Patrice Kadionik de l'ENSEIRB

Exemple d’application des SE



Qu'est ce qu'un système embarqué ?

- Un système embarqué (ou enfoui) est un système électronique intégré dans un système (voiture, avion, ...) qui sert à exécuter une tâche particulière (contrôle, communication, …).

- Un système embarqué peut être défini comme un système électronique et informatique autonome ne possédant pas des entrées/sorties standards comme un clavier ou un écran d'ordinateur.

- Tout système numérique autonome constitué de parties organisées pour assurer une fonction ou un ensemble de fonctions dans son environnement.

- C'est tout système numérique autre qu'un PC, ou station de travail ou serveur.

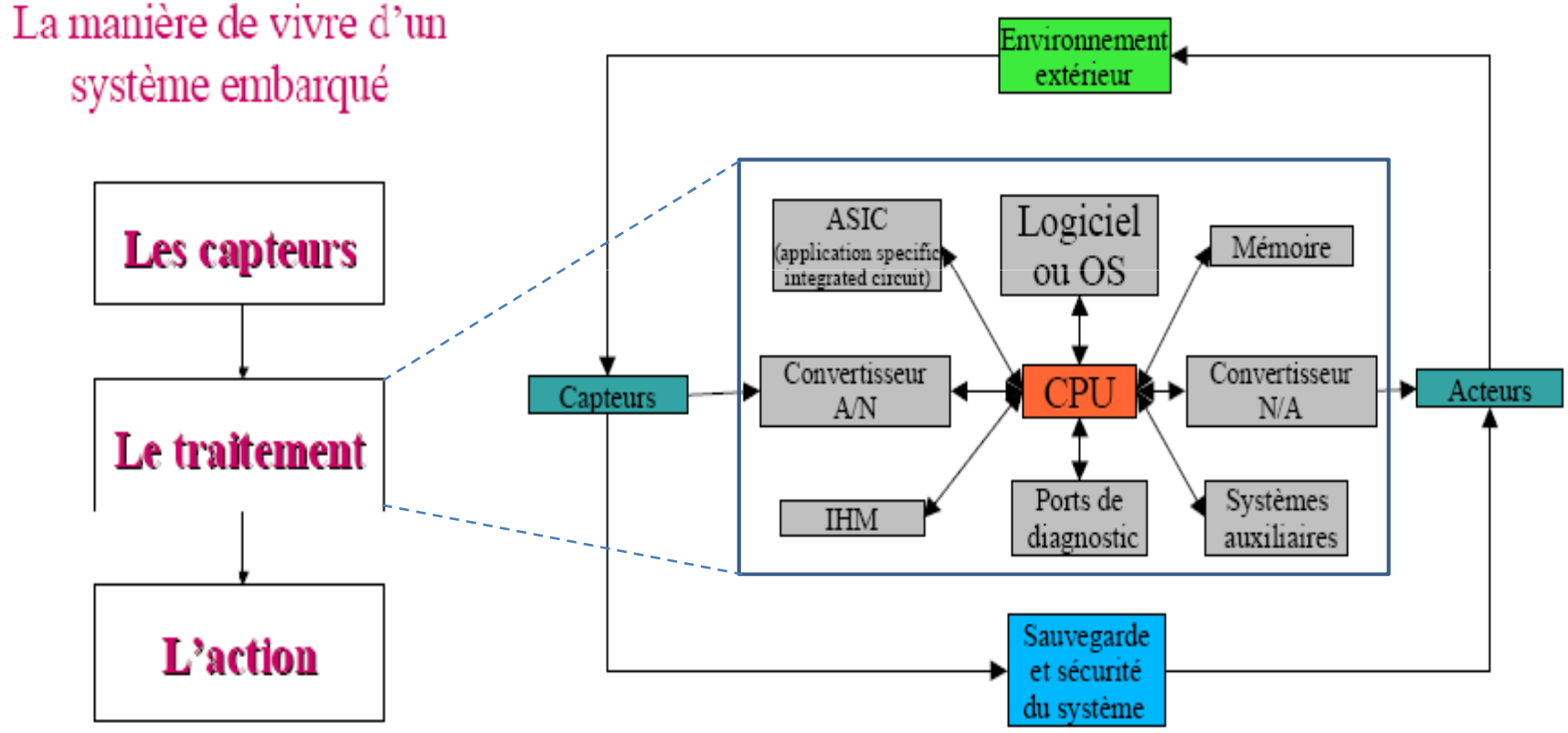
- Calculateurs dédiés à quelques applications/fonctions enfouis dans un appareil. Généralement pas de clavier ni d'affichage.

Qu’est ce qu’un système embarqué ?

- Un système embarqué :
  - Est un système numérique.
  - Utilise généralement un processeur.
  - Exécute un logiciel dédié pour réaliser une fonctionnalité précise.
  - Remplace souvent des composants électromécaniques.
  - N'a pas réellement de clavier standard (BP, clavier matriciel...).
  - L'affichage est limité (écran LCD …) ou n'existe pas du tout.
  - N'est pas un PC.

Caractéristiques et organisation

La manière de vivre d'un système embarqué



Secteurs d'activité adressés

Secteurs d'activité adressés

Secteur	Prestataire
Aeronautique, Militaire, Espace	38%
Automobile	31%
Equipement médical	15%
Commerce distribution	14%
Industrie de la fabrication des machines	14%
Autres Industries	13%
Energies	11%
Logiciels pour téléphones portables, PDA	10%
Electronique grand public	9%
Fabrication de produits télécoms	8%
Fabrication de téléphone mobile	7%
Secteur financier	5%
Fabrication de carte à puces	4%
Opérateur Télécom	2%
Autre	7%

Plan du cours

- Définition et généralités
- Conception des systèmes embarqués
- Cible logiciel
- Cible matériel
- Cible mixte

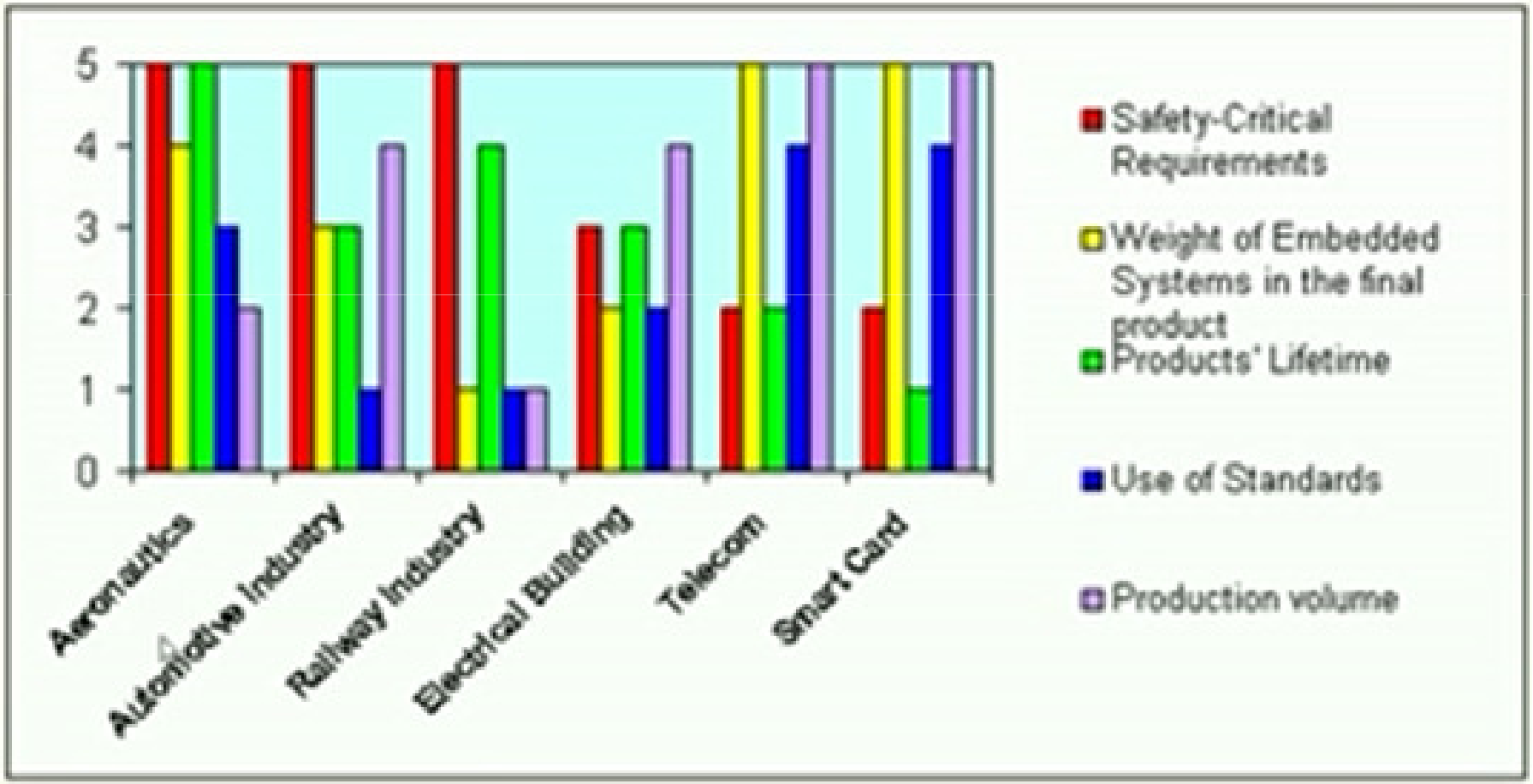
Contraintes de conception des systèmes embarqués

- Performance: puissance de calcul MIPS (Million d'Instructions Par Seconde)

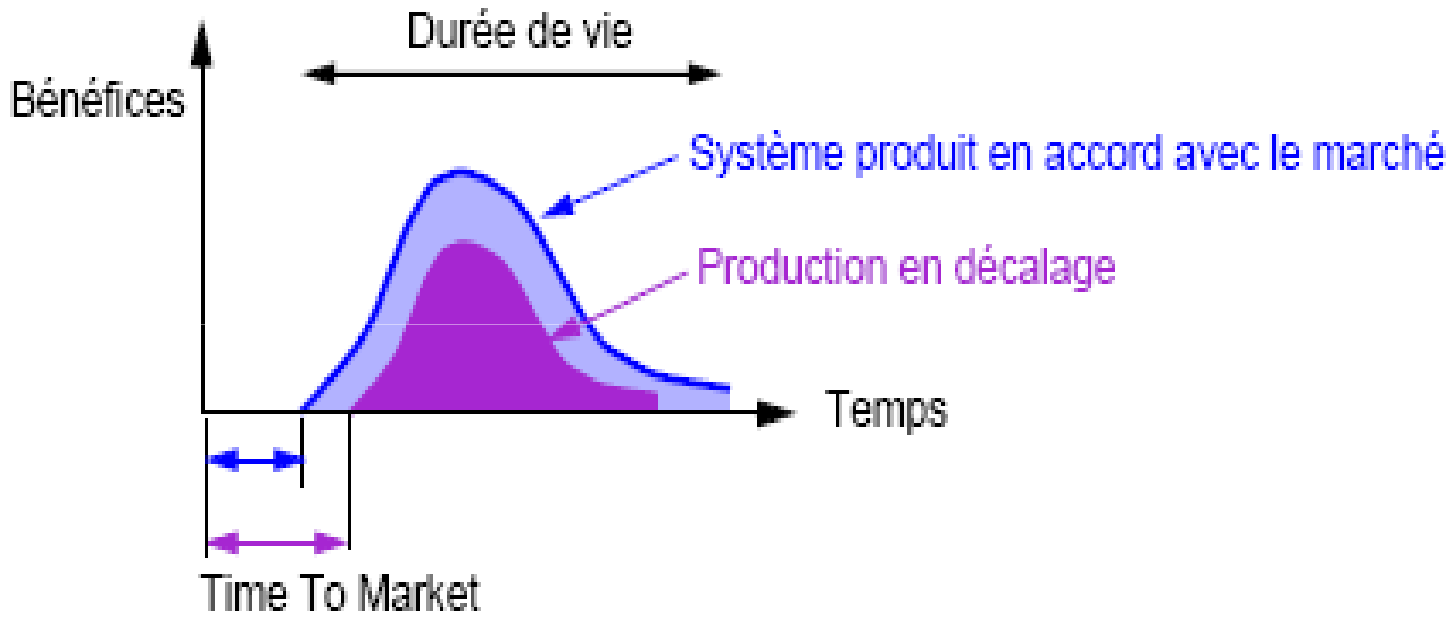
  ➔ temps d'exécution

- Fiabilité: la fiabilité est l'aptitude d'un système à accomplir une fonction requise dans des conditions données pour une période de temps donnée
- Surface et encombrement: GSM…
- Consommation énergétique: lié à l'autonomie des batteries (PDA, téléphone mobile…)
- Sûreté: aucun dommage (automobile, avionique…)
- Coût et temps de développement: faible coût

Contraintes



Contraintes de conception des systèmes embarqués : TTM



Les produits ont une durée de plus en plus faible

Réduire le «time to market»

Réutilisation pour concevoir d'autres produits (rentabiliser)

Contraintes de conception des systèmes embarqués

- Un système embarqué doit répondre à des contraintes temps réel:
  - Un système temps réel doit réagir à un stimuli dans un intervalle de temps dépendant de l'environnement.
  - Un système temps réel qui produit une bonne réponse mais trop tard est défaillant.
  - Contrainte de temps réel: dure (hard) ou souple (soft)
    - Une contrainte temps réel est appelé dure si le non respect de cette contrainte peut mener à des situations critiques voir catastrophiques
    - Les autres contraintes sont appelées soft

Evolution des systèmes embarqués

1. Evolution technologique:

La miniaturisation des transistors a permis d'augmenter considérablement la capacité d'intégration dans les systèmes embarqués :

7,2 10^9 transistors par cm^2 est envisagée pour 2020

➔ intégration

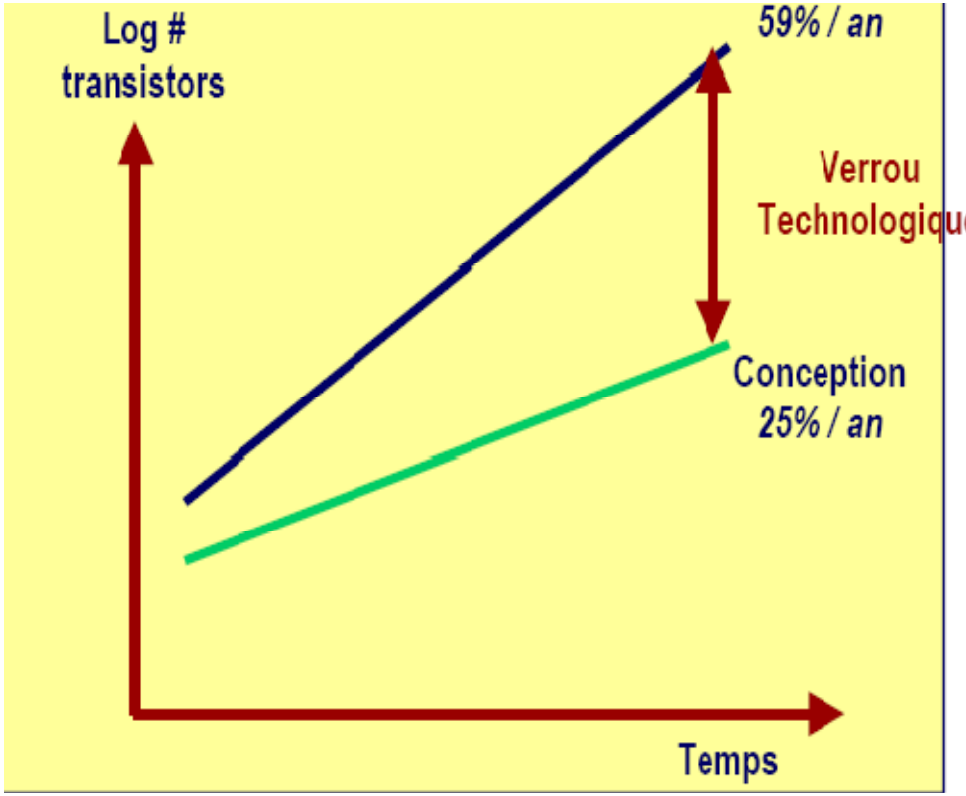
2. Evolution applicative:

Des applications plus complexes avec un nombre important et varié de fonctionnalités:

➔ Les téléphones portables intègrent de plus en plus de fonctionnalités et ils ont des tailles toujours réduites et sont alimentés par batteries.

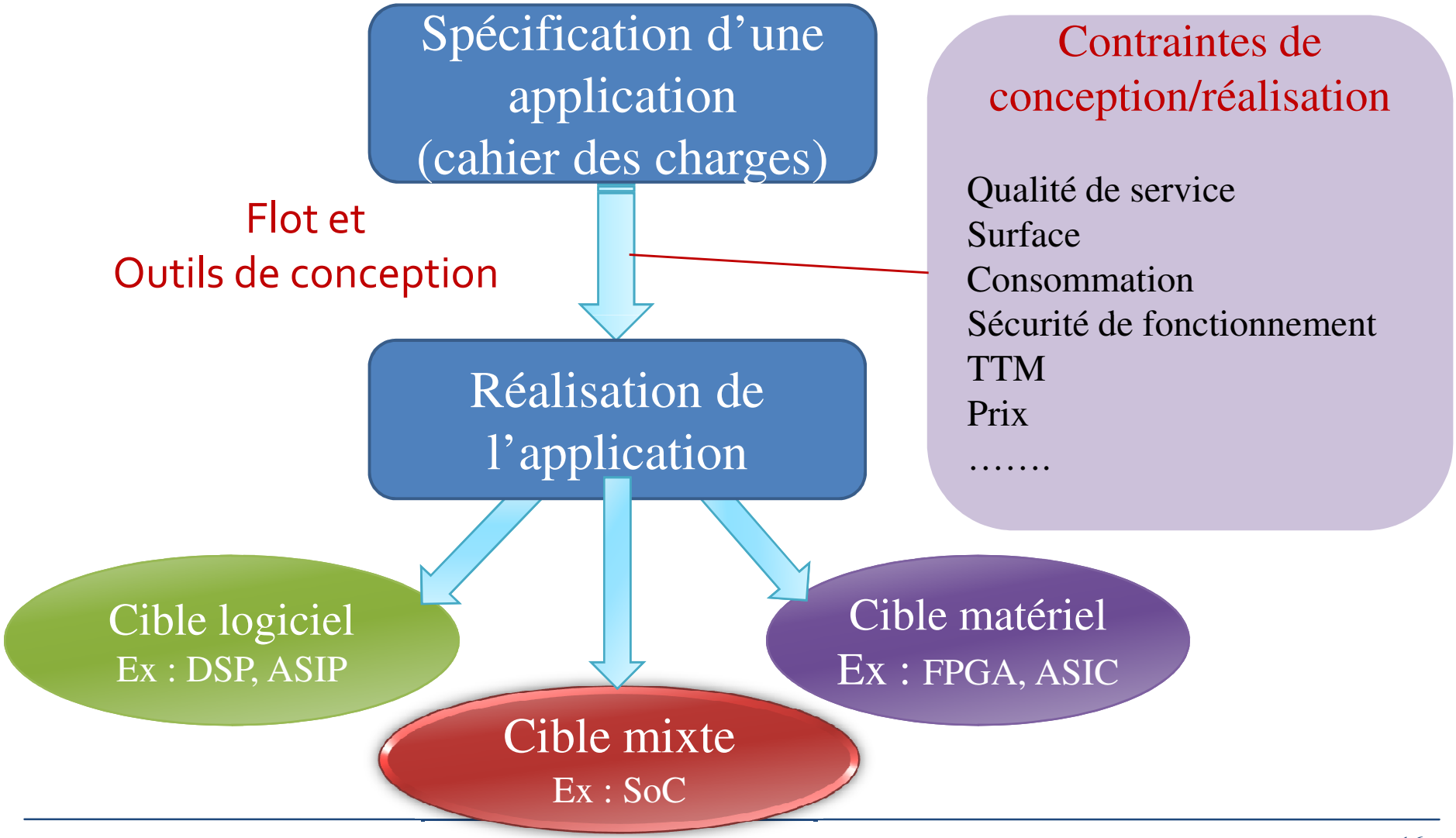
Evolution technologique/Conception

Le processus technologique autorise un accroissement de la complexité de 59% par an → Loi de MOORE



L'efficacité des concepteurs n'augmente "que de" 25% par an

Conception des systèmes embarqués



Différentes cibles utilisables pour concevoir un système embarqué

- Cible logiciel
  - Les processeurs généralistes
  - Les DSPs
  - Les Microcontrôleurs
- Cible matériel
  - Les ASICs : Application Specific Integrate Circuit
  - Les circuits reconfigurables :
    - FPGA : Field Programmable Gate Array
- Cibles mixte
  - Les SOCs(système sur puce) : L'intégration de plusieurs unités matérielles et logicielles sur une même puce

Plan du cours

- Définition et généralités
- Conception des systèmes embarqués
- Cible logiciel
- Cible matériel
- Cible mixte

Cible logiciel

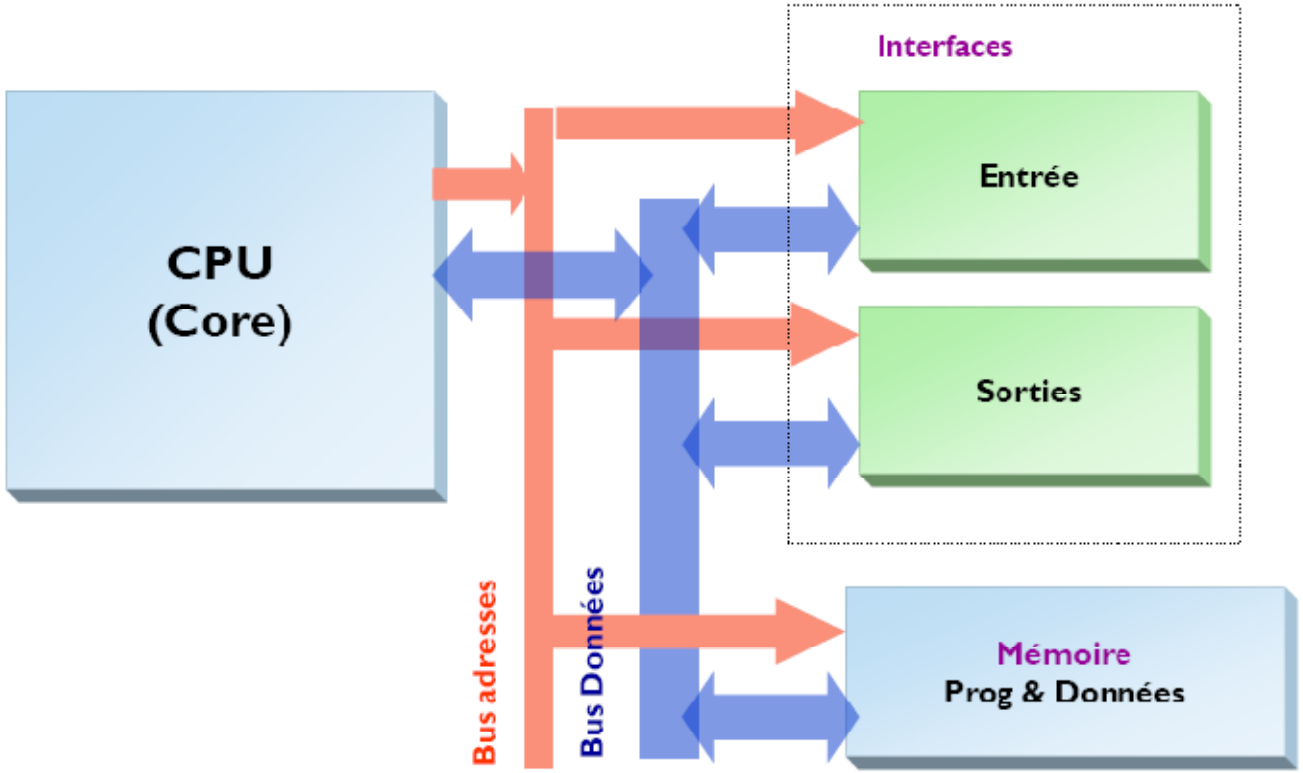
- Ce sont des cibles programmables, c'est-à-dire qu'on peut modifier l'application dédiée juste en modifiant le code, à travers :
  - Les processeurs généralistes (GPP : General Purpose Processor)
  - Les DSP (Digital Signal Processing)
  - Les Microcontrôleurs

Processeurs Généralistes

- Processeurs a usage général qui ne dépendent d'aucun langage de programmation
- Choix des processeurs embarqués
  - Coûts
  - Consommation
  - etc.
- Exemple :
  - Famille ARM
  - Famille MIPS
  - Famille PowerPC
- Différentes architectures
  - Architecture de Von Neumann
  - Architecture de Harvard

Architecture de Von Neumann

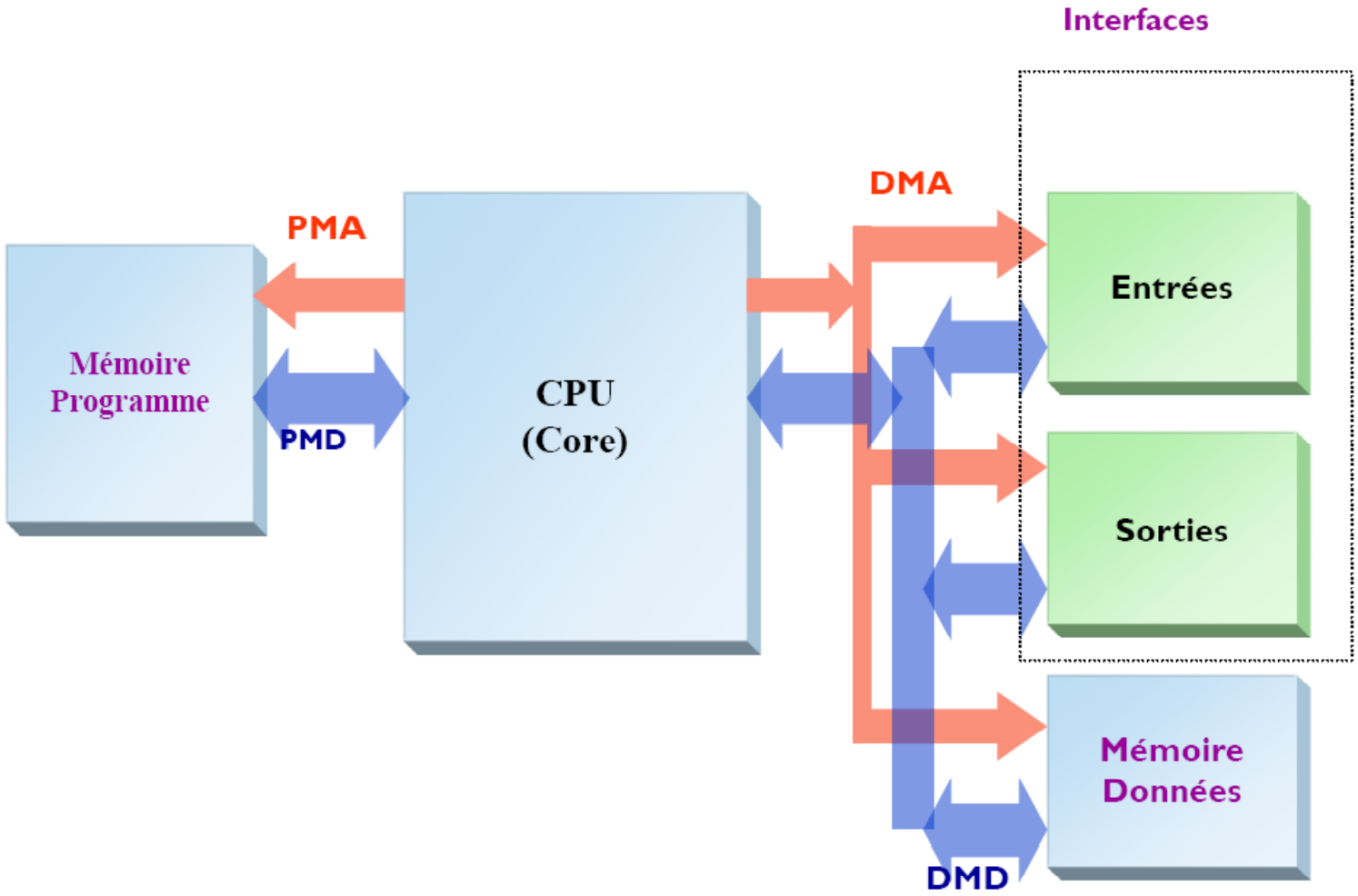
- Architecture de Von Neumann



Architecture de Von Neumann

- Mémoire de donnée et mémoire de programmes partagée
- L'exécution d'une instruction peut se faire en plusieurs cycles processeur :
  - Recherche de l'instruction (Instruction fetch)
  - Recherche de l'opérande 1 (data 1 fetch)
  - Recherche de l'opérande 2 (data 2 fetch)
  - …
- Performances de calcul limitées
  - Ex : non appropriée aux opérations de traitement du signal

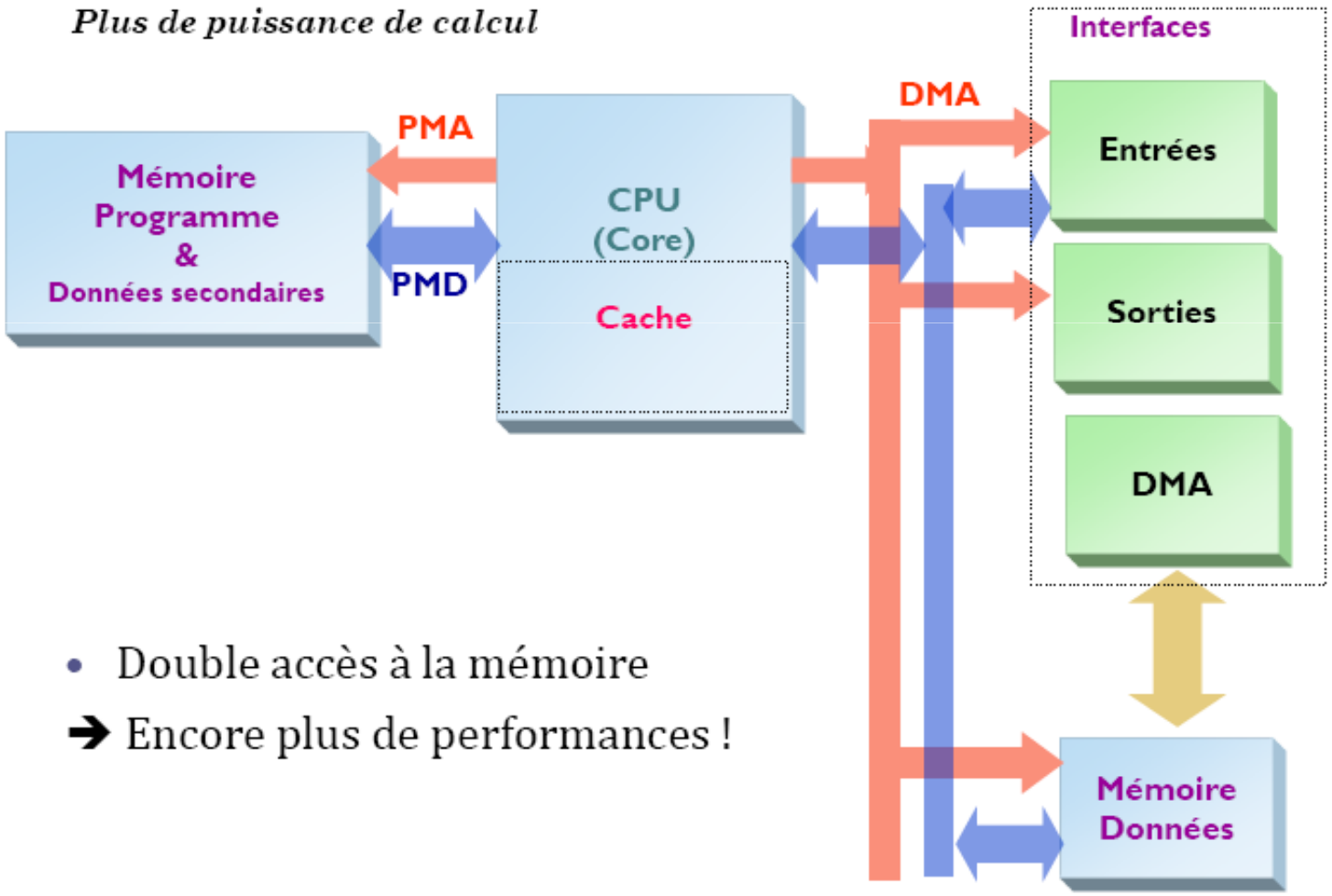
Architecture de Harvard



Architecture de Harvard

- Séparation entre la mémoire de donnée et la mémoire de programme
- Chaque mémoire comporte ses bus propres à elle
- Recherche de l'instruction et de la donnée en 1 cycle d'horloge
- Le CPU (core) comporte un chemin de donnée plus organisé
- Puissance de calcul meilleure

Architecture de Harvard modifiée



- Double accès à la mémoire

➔ Encore plus de performances !

Caractéristiques générales des microprocesseurs

- Performances
  - Temps d'exécution par tâche = I x C x T
    - I : nombre d'instructions par tâche
    - C : nombre de cycles machine par instructions
    - T : temps d'un cycle machine (dépend de la technologie et de l'efficacité de l'ALU)

- Types des architectures des processeurs :
  - CISC (Complex Instruction Set Computer) : I faible, C grand
  - RISC (Reduce Instruction Set Computer) : I élevé, C faible
  - VLIW (Very Large Instruction Word) : I réduit car macro-instruction RISC, C faible

Les architectures CISC

- Ancienne Architecture des processeurs
- Architecture présentant un jeu d'instructions complexe
  - Plusieurs opérations peuvent être codés par une même instruction
- Plusieurs modes d'adressage
- Nécessite moins de mémoire par rapport à une architecture RISC
- Exemple :
  - Motorola 680x0 ,
  - S/360 d'IBM,
  - Intel Pentium
  - Intel Pentium Pro, Pentium II III et 4 : PeusdoCISC(cœur de RISC mais vue comme un CISC) permet de garder la compatibilité ascendante des processeurs (x86)

Les architectures RISC

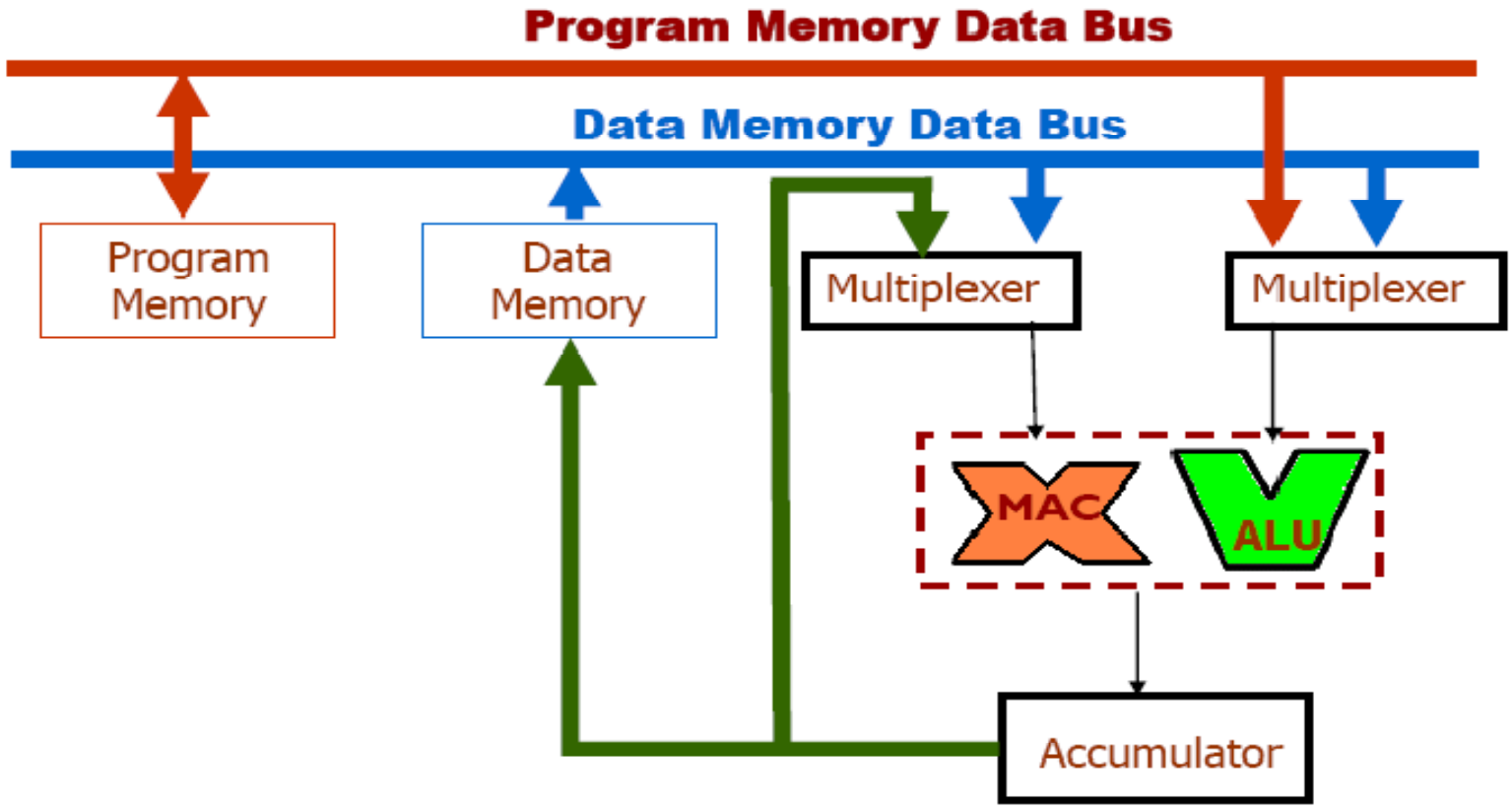
- Architecture présentant un jeu d'instructions relativement réduit
  - Une seule opération /instruction
  - Taille fixe pour les instructions
- Modes d'adressage simples
- Ont permis une augmentation de la fréquence
- Présente un nombre important de registres généraux
- Les seules instructions ayant besoin d'accès à la mémoire sont les instruction de chargement et de rangement
- Exemple :
  - PowerPC
  - ARM
  - SPARC
  - MIPS

Les architectures VLIW

- Very Long Instruction Word
  - Famille d'ordinateurs dotés d'un Processeur à mot d'instruction très long (> 128 bits)
  - Une instruction = {instructions indépendantes}
  - Mot VLIW= Bundle
- Les compilateurs génèrent un code en fonction des ressources disponibles
- Exemple :
  - TriMedia de Philips
  - Crusoe de Transmeta
  - Itanium d'Intel

Digital Signal Processing (DSP)

- Processeurs dédiés et optimisés pour le traitement numérique du signal (filtrage, extraction de signaux, etc.).

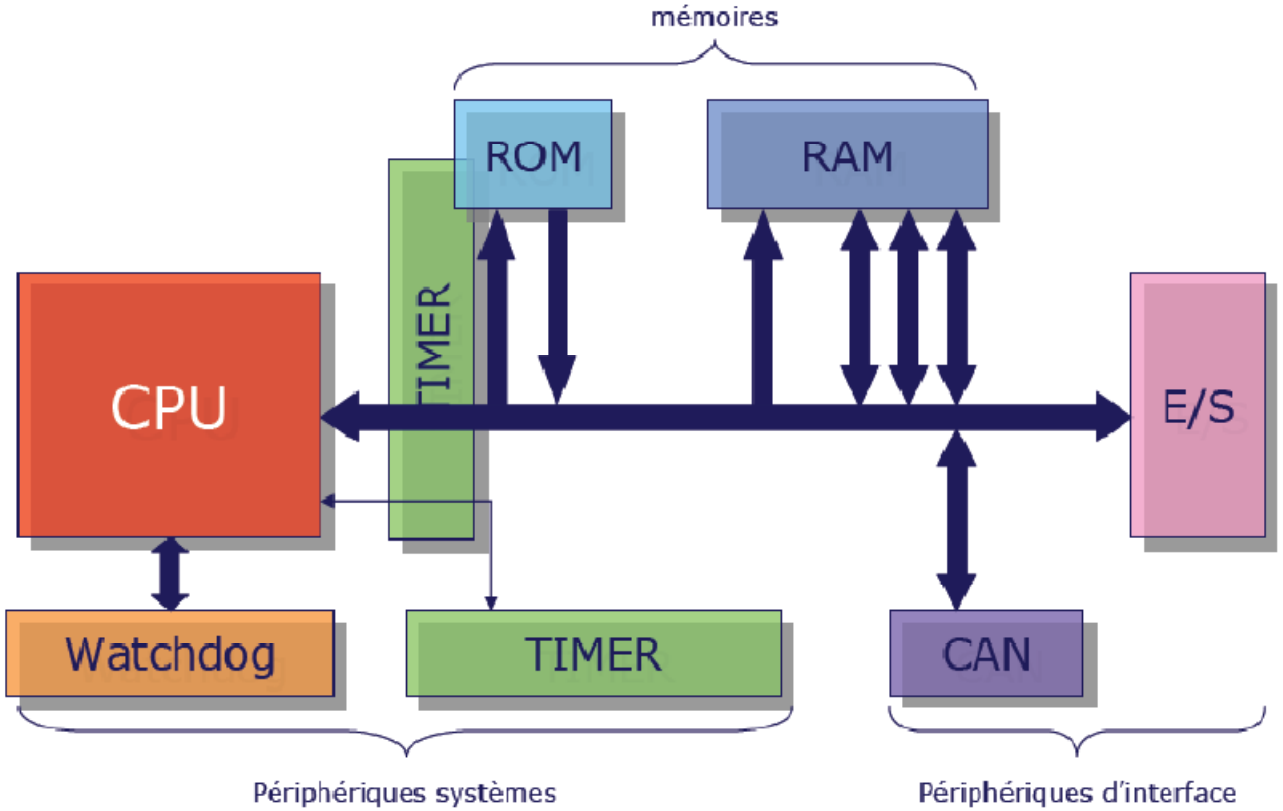


Digital Signal Processing (DSP)

- Caractéristiques
  - Architecture RISC complexe, superscalaire(plusieurs unités de traitements), pipeline
  - Architecture Harvard et Super Harvard (nombreux bancs mémoire)
  - Instructions complexes mais jeux d'instructions réduit
  - Exemple : Texas Instrument C6x
- Avantages
  - Très économique : pas besoin d'acheter des périphériques
  - Spécialisés dans le traitement du signal
  - Peuvent mélanger calcul flottant et virgule fixe
- Inconvénients
  - Coûts élevés
  - Consommation d'énergie élevée

Les Microcontrôleurs

Un circuit intégré rassemblant dans un même boîtier un microprocesseur, plusieurs types de mémoires et des périphériques (Entrées-Sorties).



Les Microcontrôleurs

- Caractéristiques
  - Architecture simple, jeux d'instructions réduit
  - Basé sur des architectures de processeurs connus
  - Exemple : 68HC11 PIC de Microchip, STM32 de ST
- Avantages
  - Très économique : pas besoin d'acheter des périphériques Spécialisés
  - Simple d'utilisation
- Inconvénients
  - Spécialisé: ne convient pas à tous les domaines d'application

Cible logiciel

- Avantages
  - Flexibilité : il suffit de modifier le programme pour modifier l'application
  - Simple à mettre en œuvre : grâce à la programmation de haut niveau (langage C) (possibilité de grande abstraction par rapport au matériel)
  - Temps et coût de conception faibles
  - Prix de reviens faible
  - Composants spécialisés, ex : DSP, microcontrôleur

Cible logiciel

- Inconvénients
  - Faibles performance : temps d'exécution élevé
  - Consommation électrique importante
  - Le passage par un compilateur peut dégrader les performances
  - Complexité du langage machine et de l'assembleur

Plan du cours

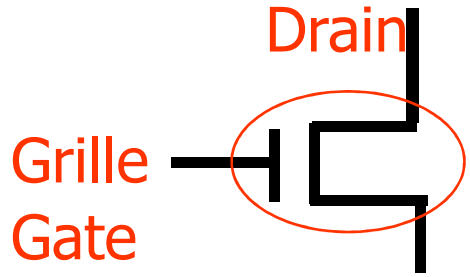
- Définition et généralités
- Conception des systèmes embarqués
- Cible logiciel
- Cible matériel
- Cible mixte

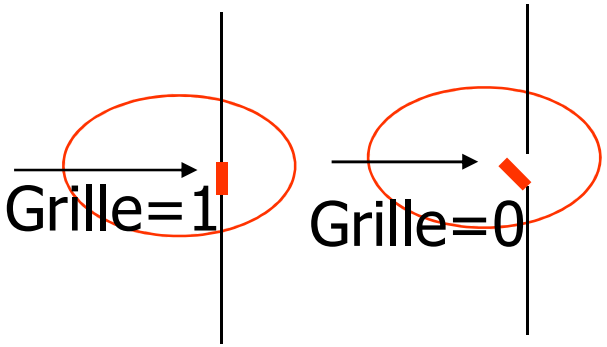
Technologie de base : les transistors

- Les circuits intégrés (CI ou chips) représentent la réalisation courante de tout système informatique
- Un CI = un ensemble de transistors + connections
- MOS: Metal Oxcyde Semiconductor
  - NMOS: NChannel MOS
  - PMOS: PChannel MOS

Technologie de base : les transistors

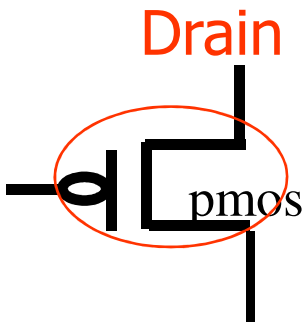
NMOS

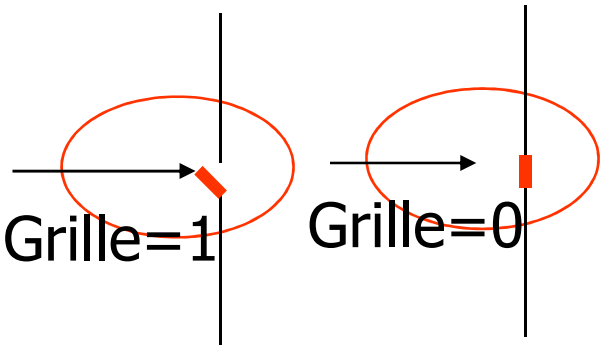




Si Grille ==1, alors Drain et Source connectées
Si Grille ==0, alors Drain et Source non connectées

PMOS

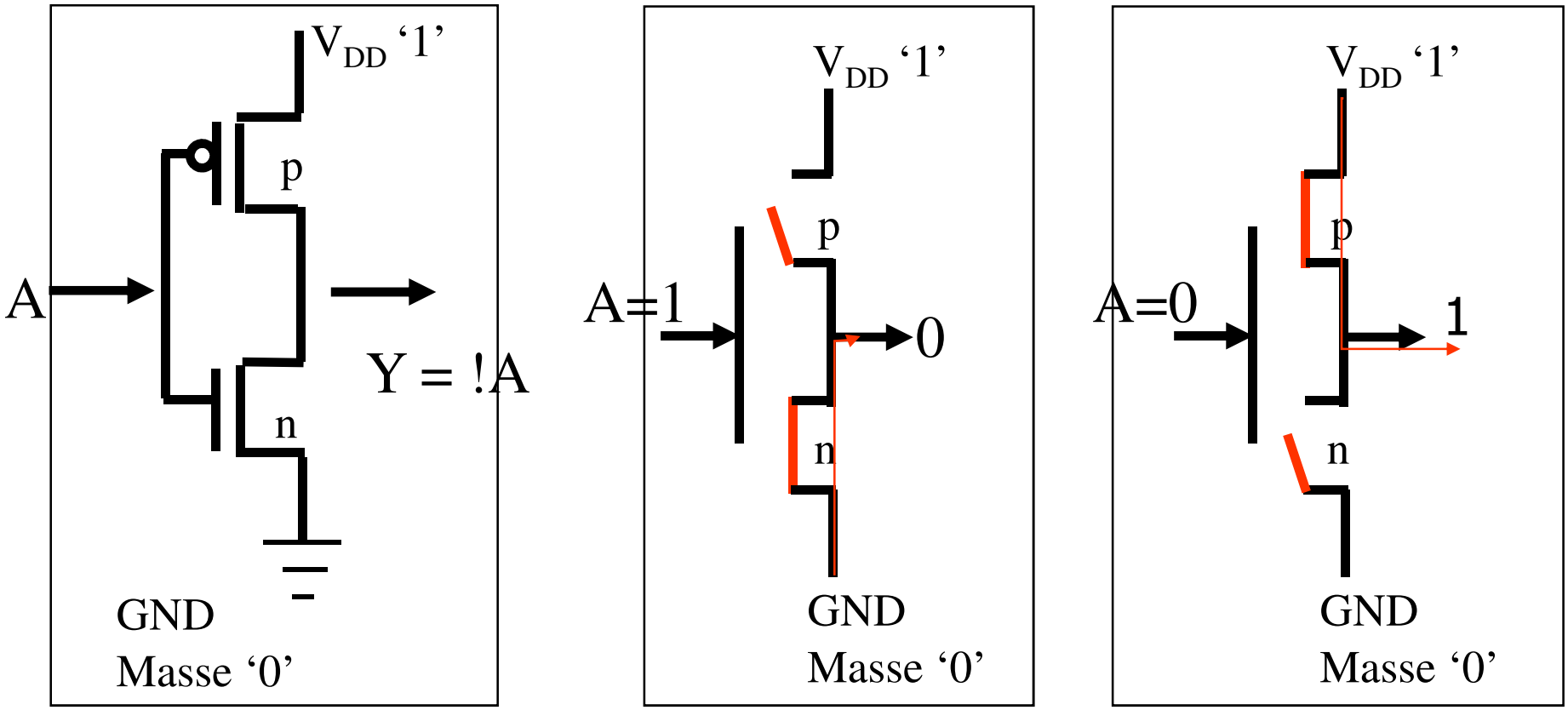




Si Grille ==0, alors Drain et Source connectées
Si Grille ==1, alors Drain et Source non connectées

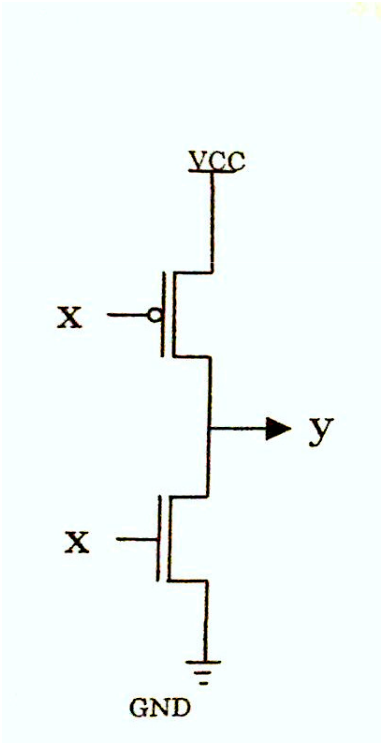
Technologie de base : les transistors

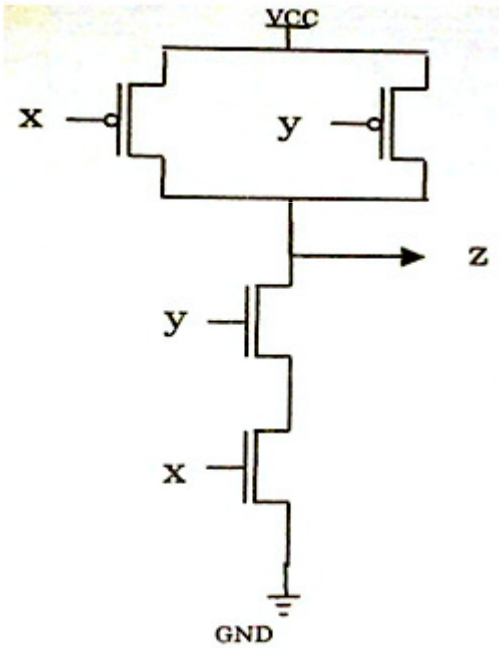
CMOS : Complementary MOS (circuit comportant des PMOS et des NMOS)

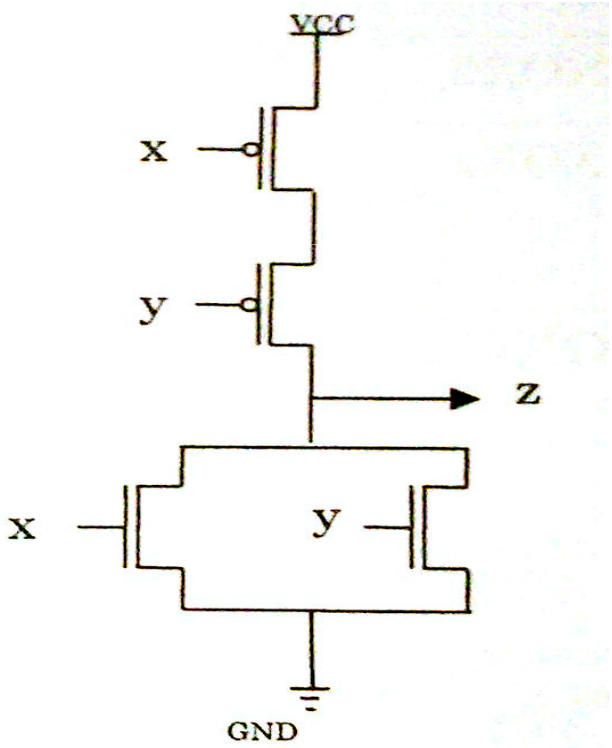


Exercice

De quel type de porte s'agit-il ?

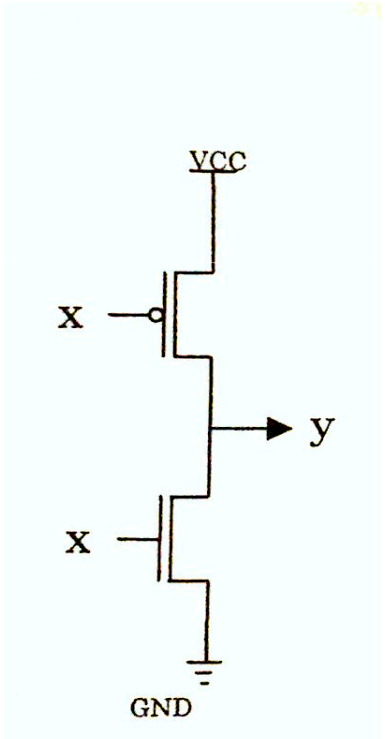






Solution de l'exercice

Une porte NOT en technologie CMOS

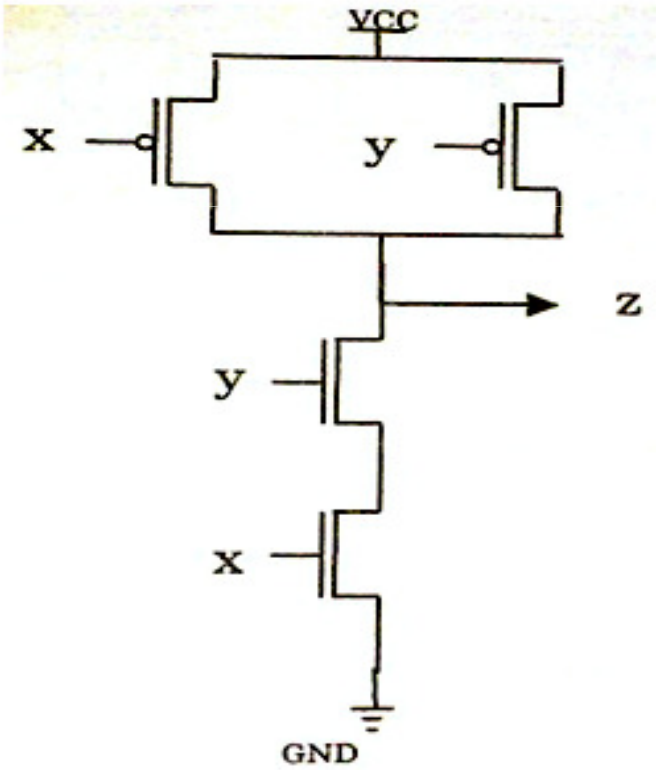


Inverseur en technologie CMOS

x	y
0	1
1	0

Solution de l'exercice

Une porte NAND en technologie CMOS

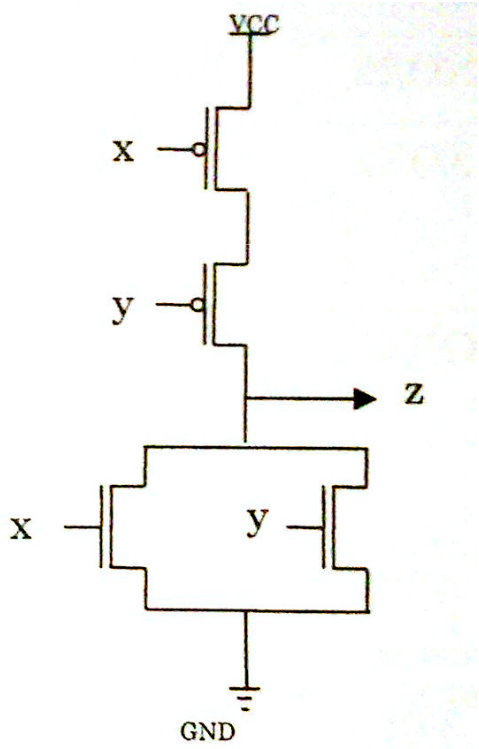


Z = NAND(x,y)

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

Solution de l'exercice

Une porte NOR en technologie CMOS



z= NOR(x,y)

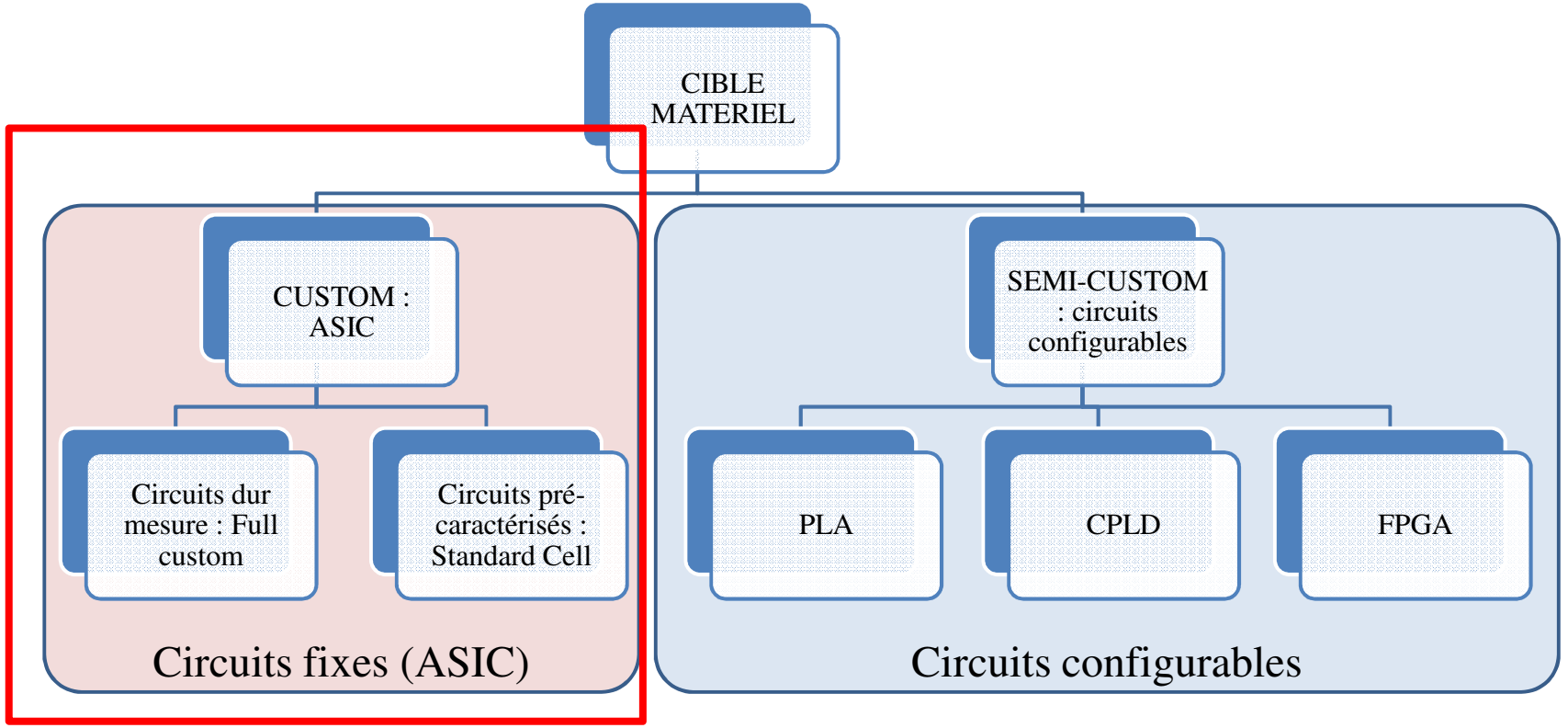
x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

Cible matériel

- Ce sont des cibles programmées dédiées et conçues pour des tâches bien déterminées et dont les traitements ne peuvent pas être modifiés

- Deux familles :
  - Les circuits fixes
    - ASICs : Application Specific Integrate Circuit
  - Les circuits configurables
    - Composants standards programmables électriquement:
      - Une seule fois (fusibles, antifusible)
      - Ou plusieurs fois : RE-PROGRAMMABLES (RECONFIGURATION)

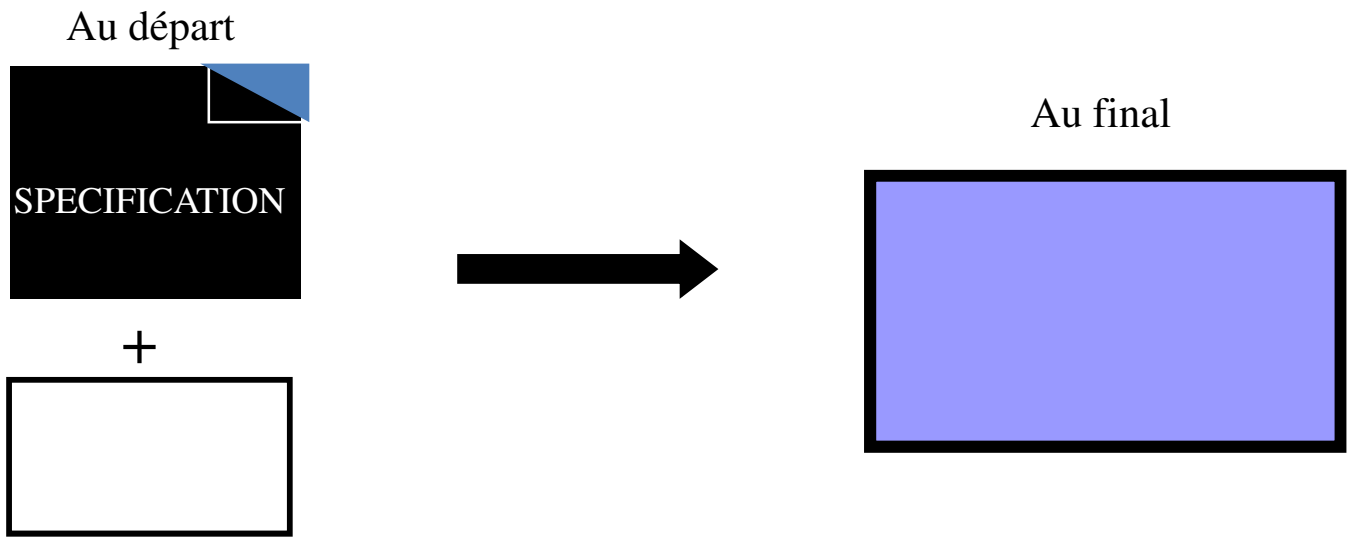
Cible matériel



- ASIC : Application Specific Integrated Circuit
- PLA: Programmable Logic Array
- CPLD : Complex Programmable Logic Device
- FPGA : Field Programmable Gate Array

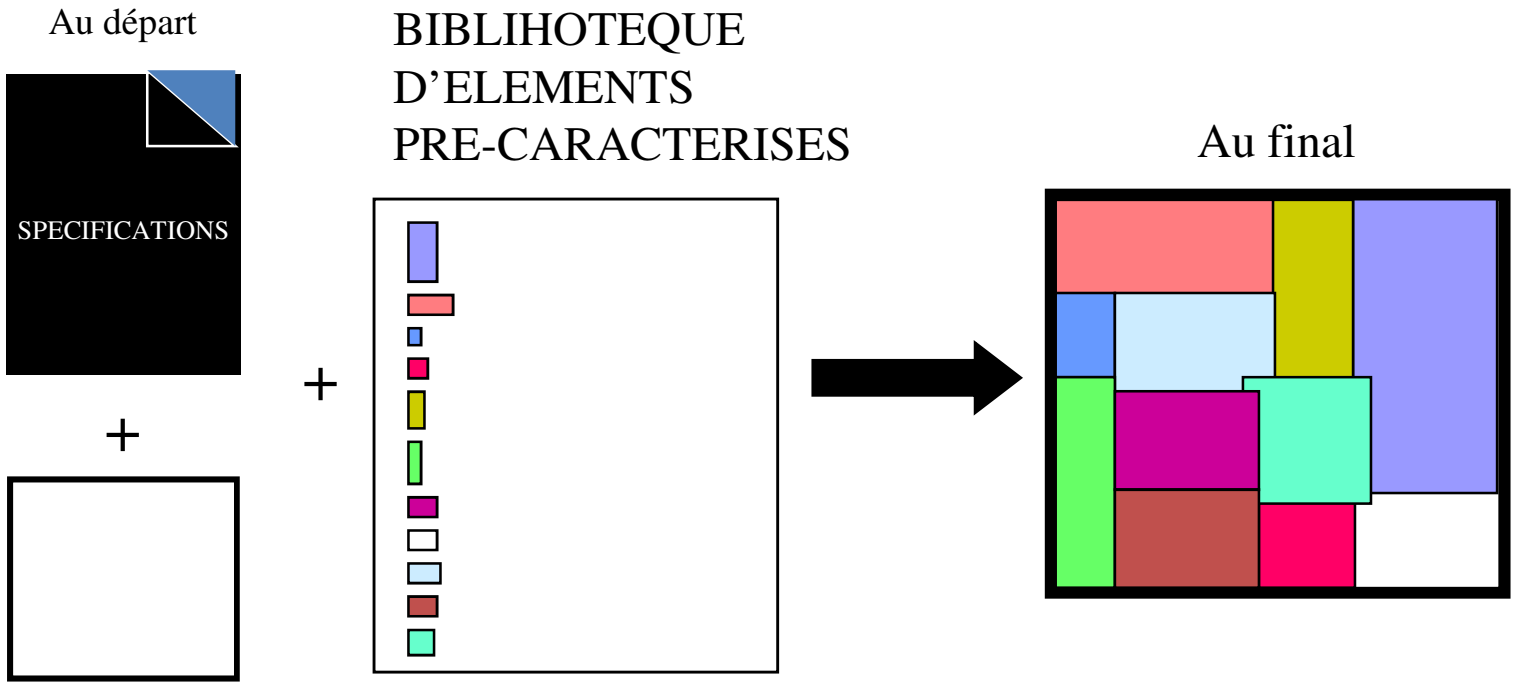
ASIC Full Custom

- Le concepteur place à la main les transistors



- Approche « full custom »
  - Conception au niveau transistor
  - Permet des circuits mixtes analogique/numérique
  - Effort de conception très important
  - Surface réduite, performance très importantes

ASIC Standard Cell



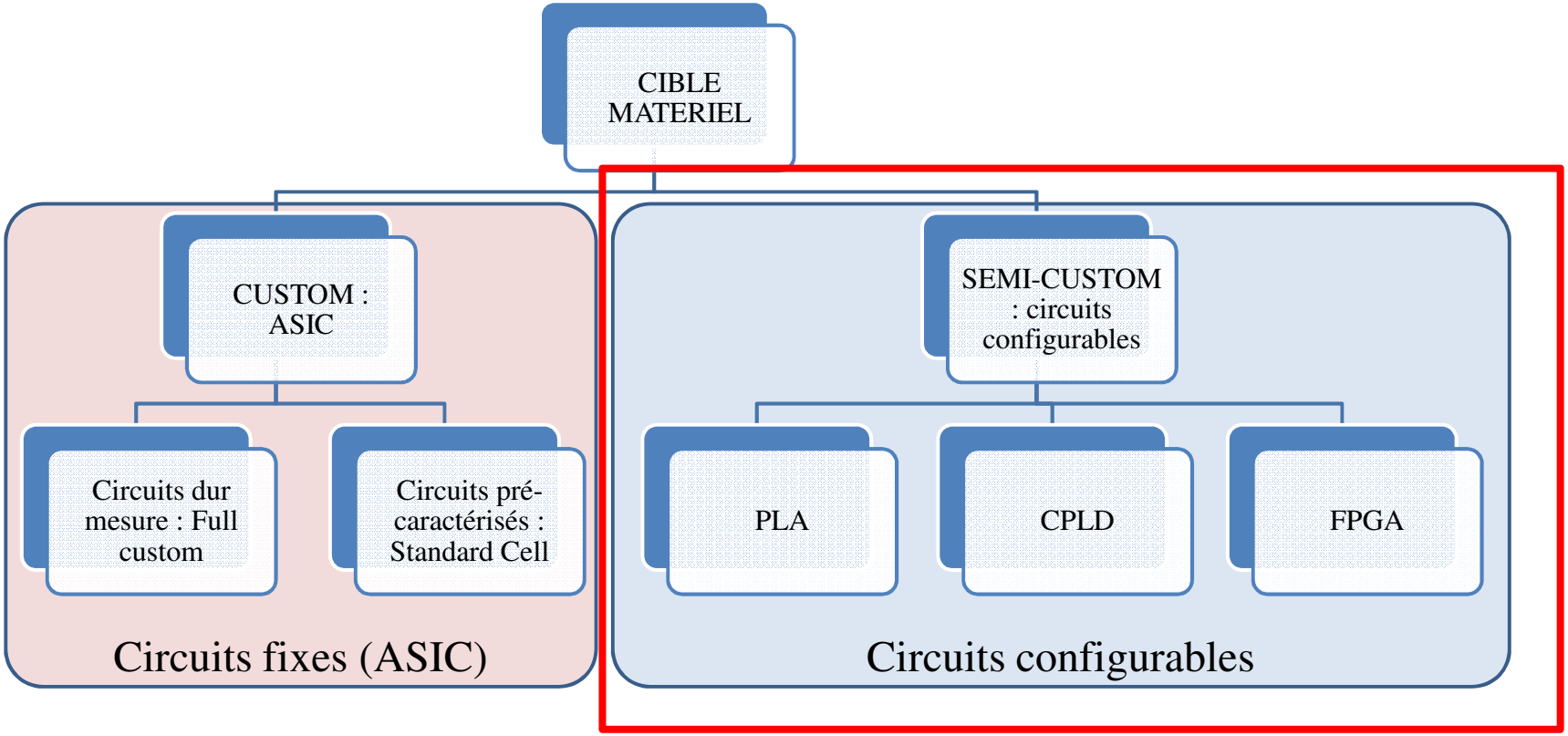
- Approche « standard cell»
  - Utilise des librairies de cellules primitives
  - Portes AND, OR, registres, SRAM, etc.
  - Effort de conception réduit, performances souvent proches du full-custom

Les ASICs

- AVANTAGES
  - hautes intégrations
  - hautes performances (vitesse, low-power)
  - coûts faibles pour de gros volumes de production
  - Personnalisation
  - sécurité industrielle

- INCONVENIENTS
  - prix du 1er exemplaire
  - pas d'erreur possible
  - non-flexible
  - time-to-market élevé
  - fabrication réservée aux spécialistes (fondeur)

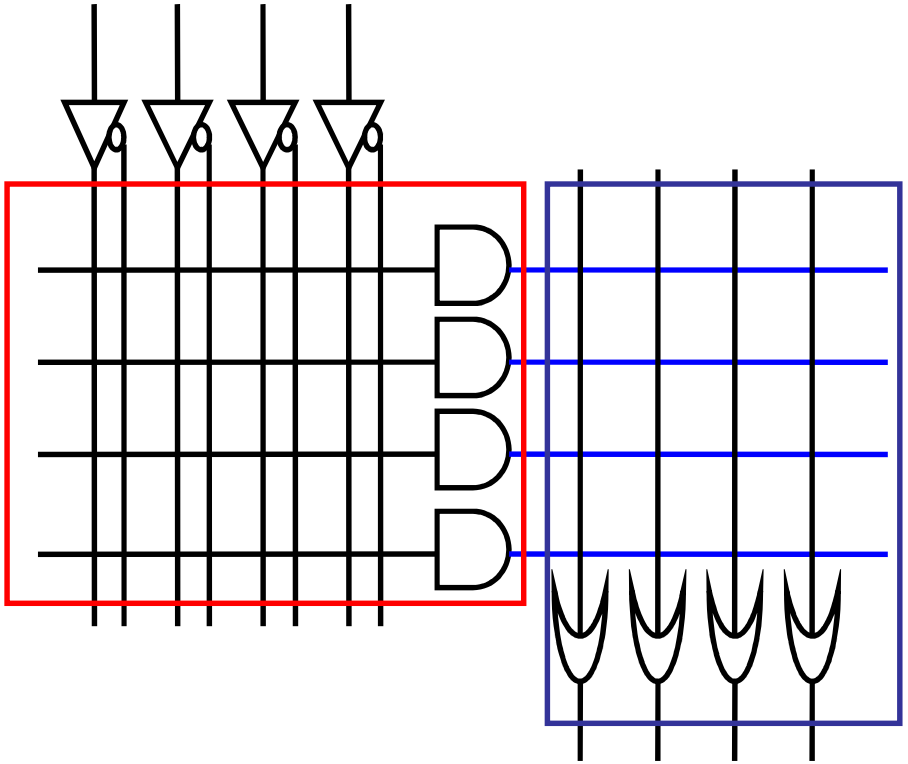
Cible matériel



- ASIC : Application Specific Integrated Circuit
- PLA: Programmable Logic Array
- CPLD : Complex Programmable Logic Device
- FPGA : Field Programmable Gate Array

PLA: Programmable Logic Device

- Matrice de « ET » programmables réalisant tous les produits possibles connectée aux sorties par des « OU » programmables



- Grande surface de Silicium utilisée
- Ces circuits ne sont plus utilisés aujourd'hui

PLA: exemple

- Donnez la configuration du circuit assurant les fonctions suivantes :

$$F_0 = ABC$$

$$F_1 = ABC + \overline{A}\,\overline{B}$$

$$F_2 = ABC + \overline{B}C + \overline{A}\,\overline{C}$$

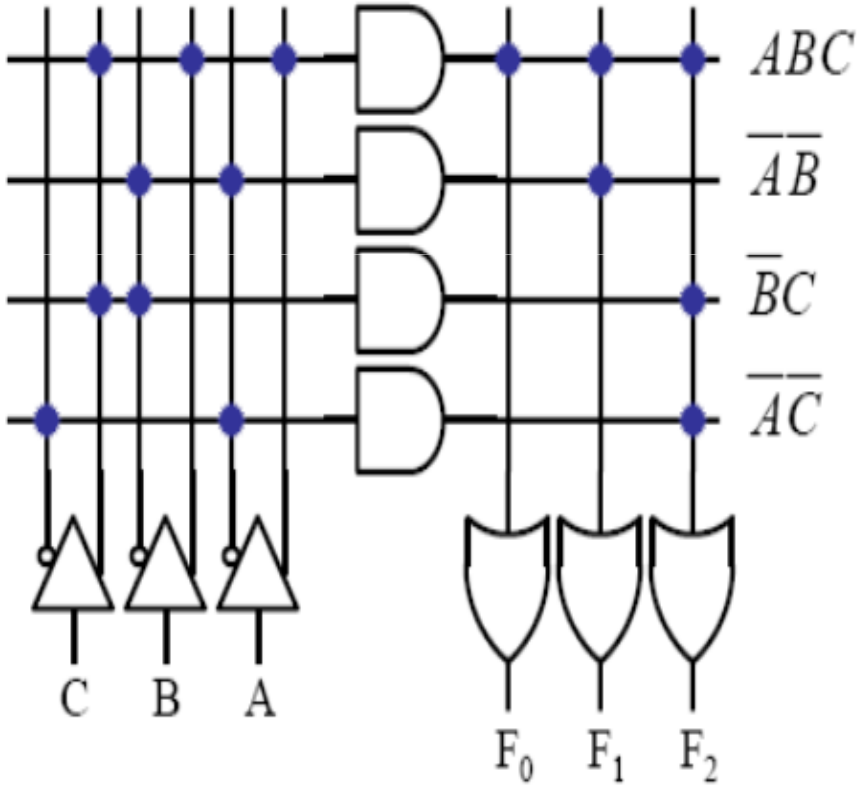
PLA : solution

- La configuration du circuit assurant les fonctions suivantes :

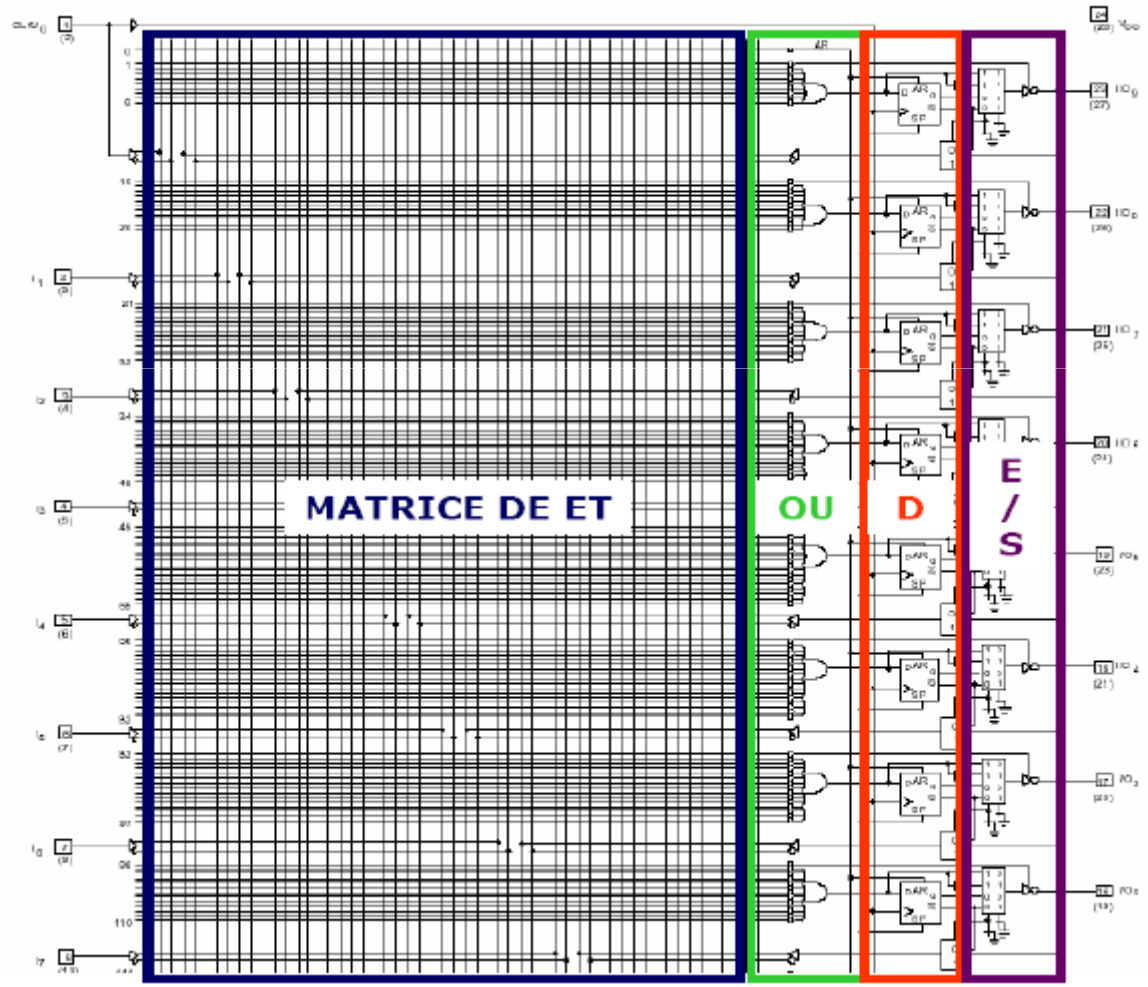
$$F_0 = ABC$$

$$F_1 = ABC + \overline{A}\,\overline{B}$$

$$F_2 = ABC + \overline{B}C + \overline{A}\,\overline{C}$$

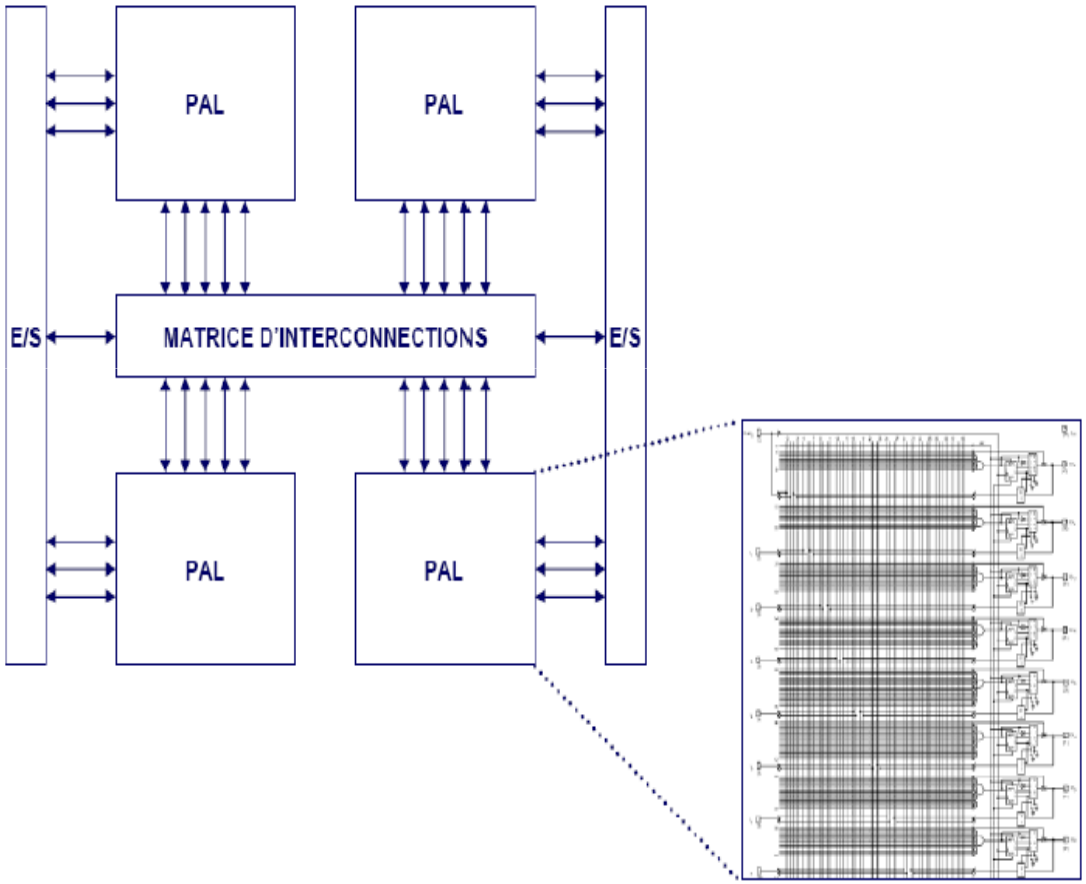


PLA: Programmable Logic Device

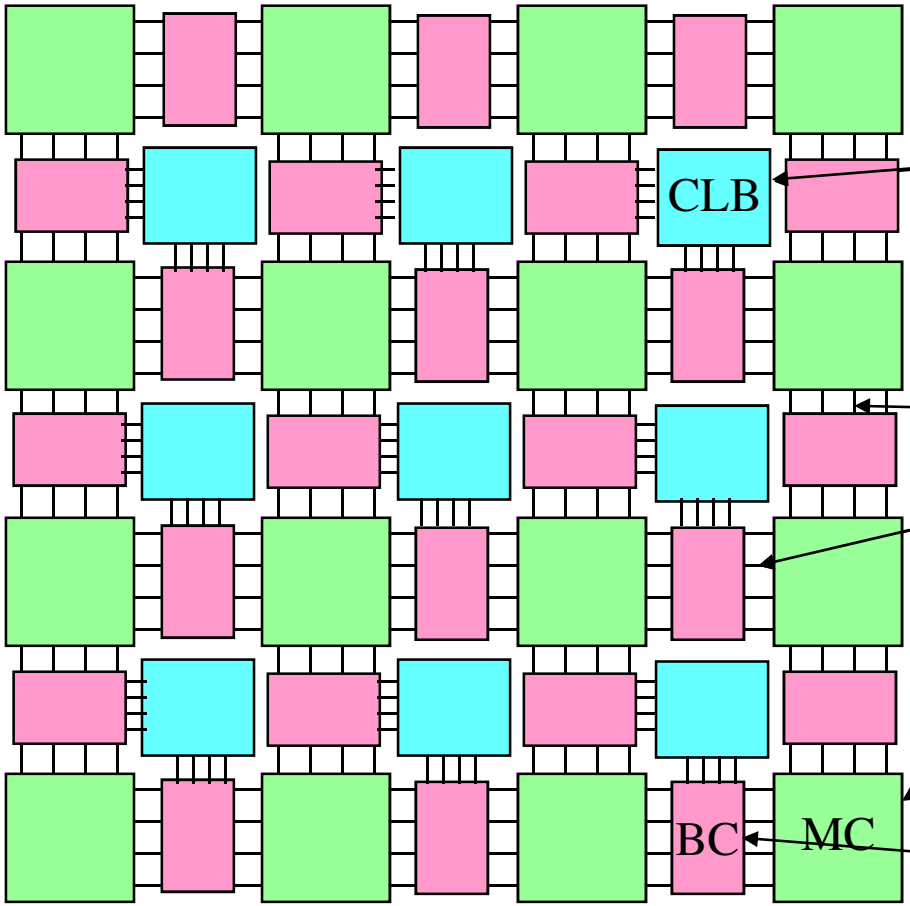


CPLD : Complex Programmable Logic Device

- Les CPLDs regroupent plusieurs PALs interconnectés par un réseau de connexions programmables.
- Les CPLDs sont les prémisses des premiers FPGAs.
- Ces circuits ne sont plus utilisés aujourd'hui car remplacés par les FPGAs.



FPGA : Field Programmable Gate Arrays



Élément configurable :

. élément logique

. élément de mémorisation

. élément arithmétique

. entrée/sortie

Réseau de routage :

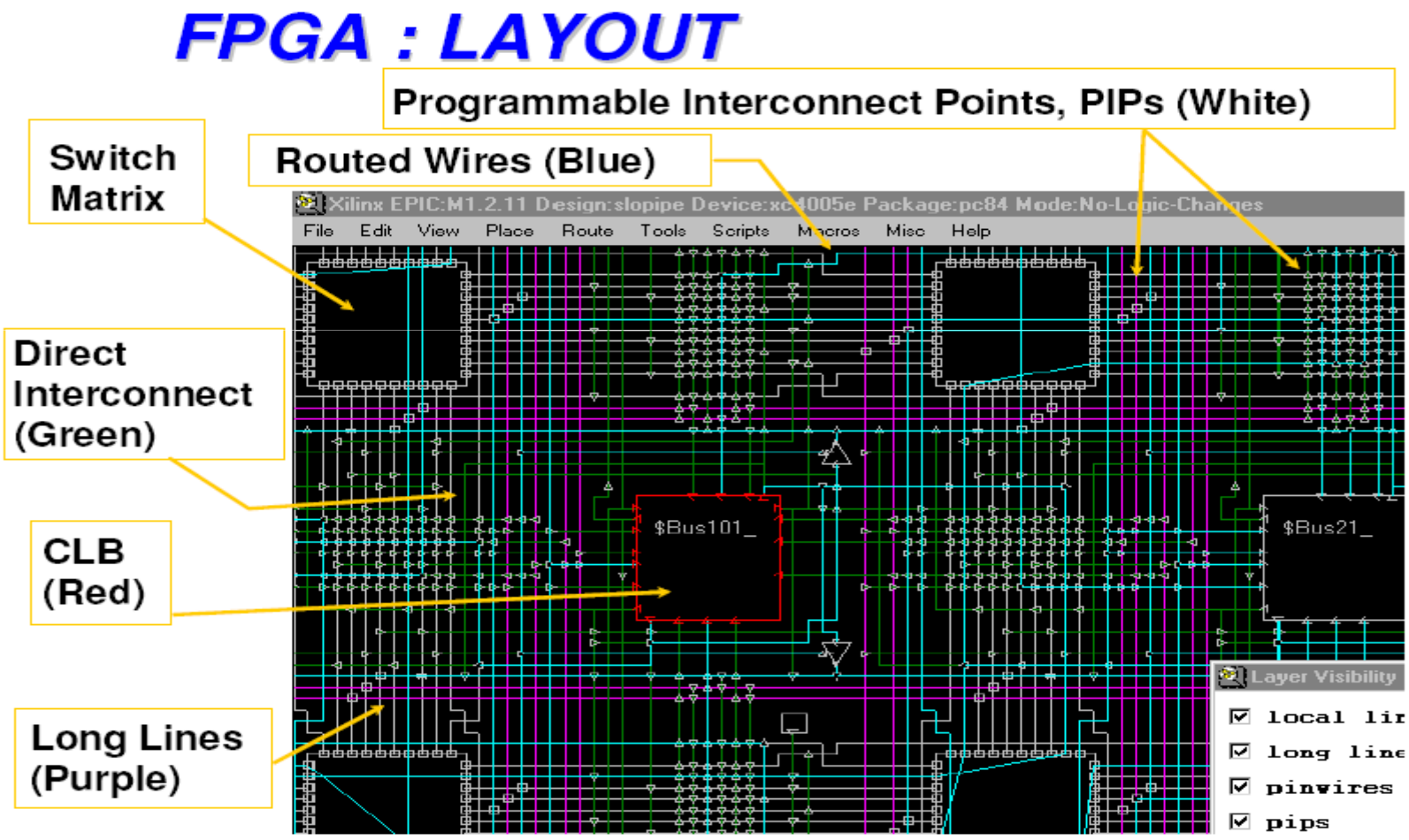
. lignes horizontales

. lignes verticales

Matrice de connexions

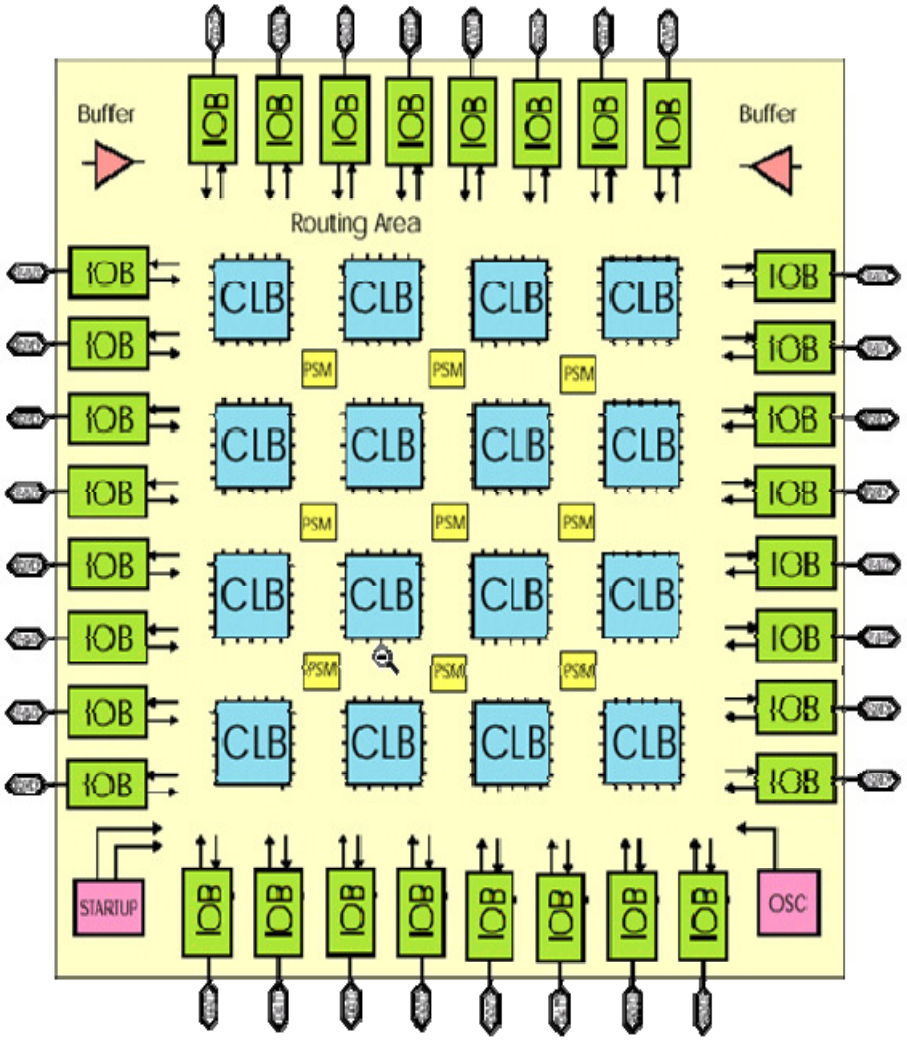
Bloc de connexions

FPGA : LAYOUT


FPGA : LAYOUT
Programmable Interconnect Points, PIPs (White)
Switch Matrix
Routed Wires (Blue)
Direct Interconnect (Green)
CLB (Red)
Long Lines (Purple)
File Edit View Place Route Tools Scripts Macros Misc Help
$Bus101_
$Bus21_
Layer Visibility
local lir
long line
pinwires
pips

FPGA : Field Programmable Gate Arrays

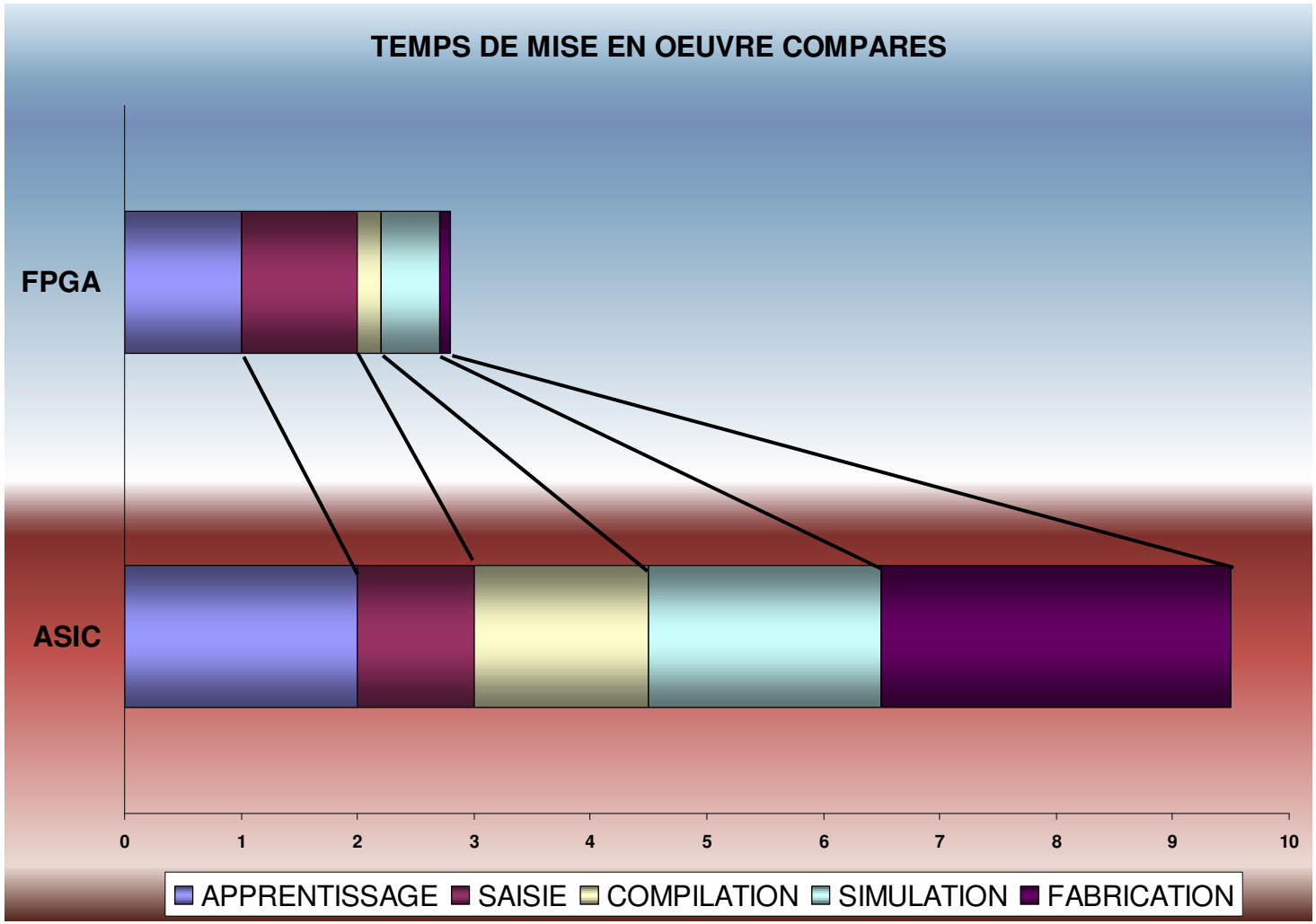
- Arrangement Matriciel de blocs logiques avec configuration des :
  - Interconnexions entre les blocs logiques,
  - La fonction de chaque bloc.
    - CLB: Configurable Logic Bloc
    - IOB: Input/Output Bloc



FPGA *vs* ASIC

Caractéristiques	FPGA	ASIC	
		Semi custom	Full custom
Densité	Faible	Moyenne	Grande
Flexibilité	Grande	Moyenne	Faible
Analogique	Non	Oui	Oui
Rapidité	Faible	Bonne	Très bonne
Temps de conception	Très petit	Moyen	Grand
Coût de conception	Très petit	Moyen	Très grand
Utilisation des outils	Simple	Complexe	Très complexe
Volume de production	Petit	Grand	Grand

Temps de mise en œuvre



Conclusion

- Le choix entre FPGA ou ASIC, se fait en fonction du cahier des charges de l'application :
  - Temps de mise sur le marché et durée de vie courte ➔ FPGA
  - Très petit nombre de circuits ➔ FPGA
  - Optimisation des performances ➔ ASIC
  - Grande série ➔ ASIC

Cible Logiciel vs cible Matériel

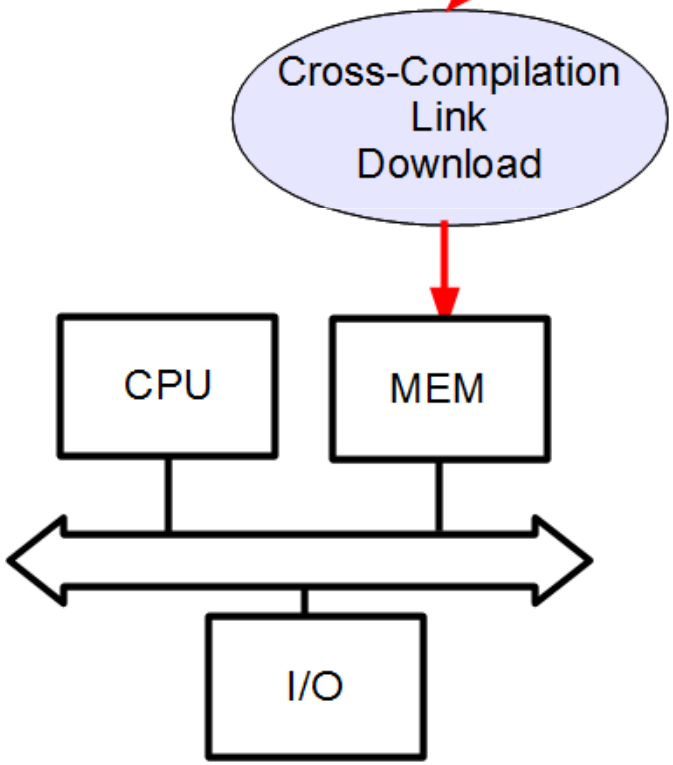
	Logiciel	Matériel
positifs	• Réduction de la surface • Partie contrôle prépondérante • Connexions avec d'autres modules logiciels • Fonctions spécialisées disponibles dans l'UAL du processeur • Evolution / Flexibilité • Coût	• Meilleure performance • Traitement du parallélisme • Traitement de données • Interactions avec l'extérieur • Connexions à d'autres modules matériels
négatifs	• Relative lenteur • Communication avec le matériel • Synchronisation avec le matériel	• Coût • Surface • Communication avec le logiciel • Synchronisation avec le logiciel

Plan du cours

- Définition et généralités
- Conception des systèmes embarqués
- Cible logiciel
- Cible matériel
- Cible mixte

Exemple : FFT

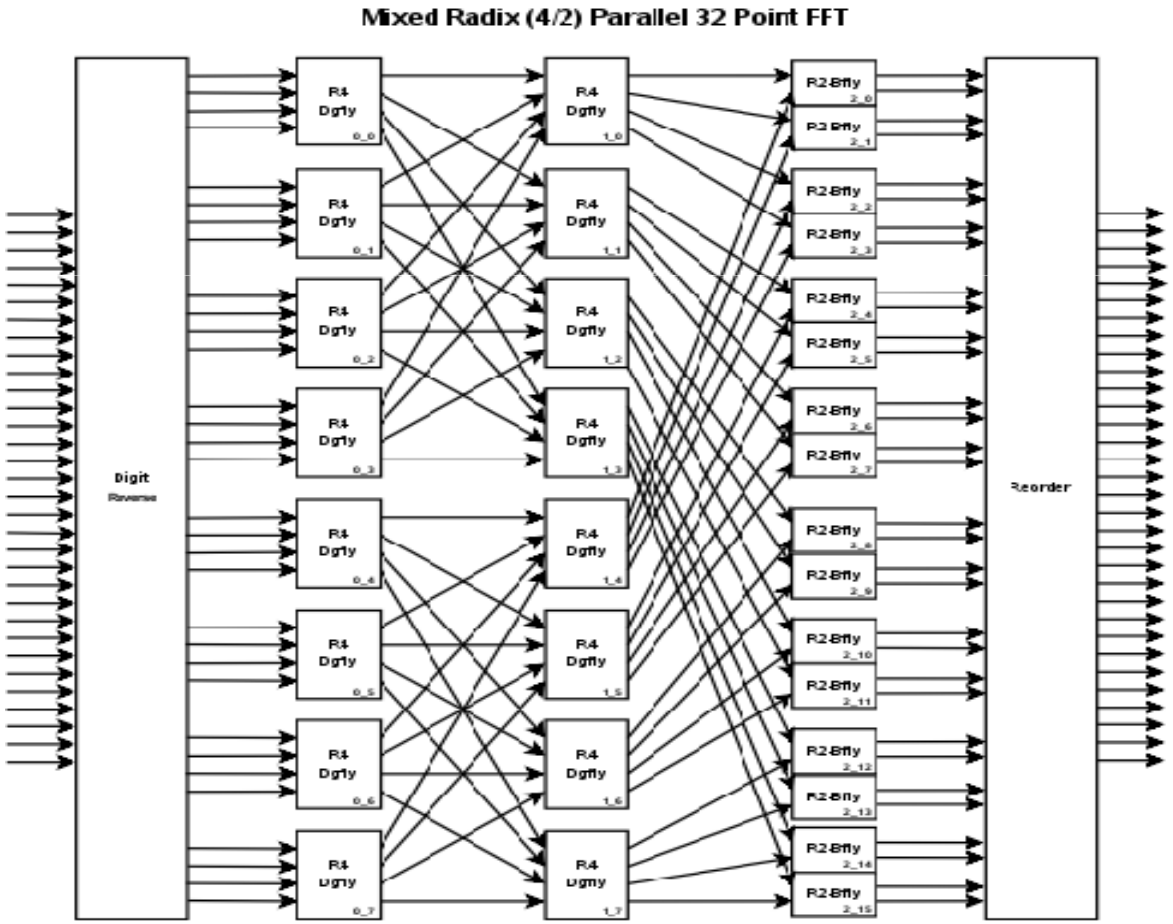
- Résolution purement logicielle




```
static void FFT_R4(complex *xin, int N, int m)
{
int i, L, j;
double ps1, ps2, ps3;
int le,B;
struct complex w[4];
for( L = 1; L <= m; L++){
le = pow(4 ,L);
B = le/4; /*the distance of buttefly*/
for(j = 0; j <= B-1 ; j++)
{ ps1 = ((TWICEPI)/le)*2*j;
w[1].real = cos(ps1);
w[1].imag = -sin(ps1);
ps2 = (TWICEPI/le)*j;
w[2].real = cos(ps2);
w[2].imag = -sin(ps2);
ps3 = (TWICEPI/le)*3*j;
w[3].real = cos(ps3);
w[3].imag = -sin(ps3);
for(i = j; i <= N-1; i = i + le) /* controle those same
butteflies*/
{
xin[i + B] = multicomplex(xin[i + B], w[1]);
xin[i + 2*B] = multicomplex(xin[i + 2*B], w[2]);
xin[i + 3*B] = multicomplex(xin[i + 3*B], w[3]);
/* DFT-4 */
DFT_4(xin + i, xin + i + B, xin + i + 2*B, xin + i + 3*B);
}
}
```

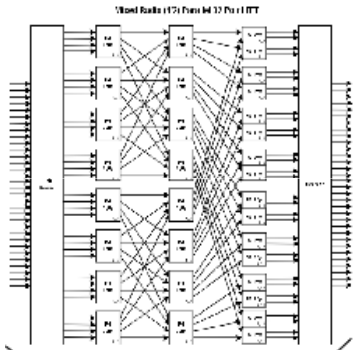
Exemple : FFT

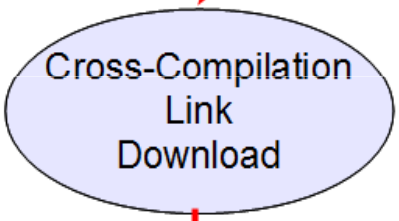
- Résolution purement matérielle



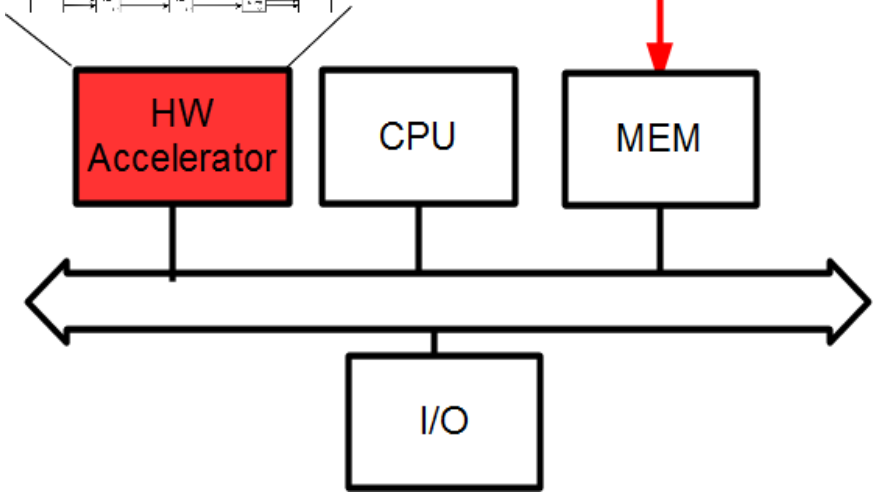
Exemple : FFT

- Résolution mixte
  - Logiciel/Matériel






```
static void FFT_R4(complex *xin, int N, int m)
{
int i, L, j;
double ps1, ps2, ps3;
int le,B;
struct complex w[4];
init_configuration()
start_hardware_accelerator()
}
```

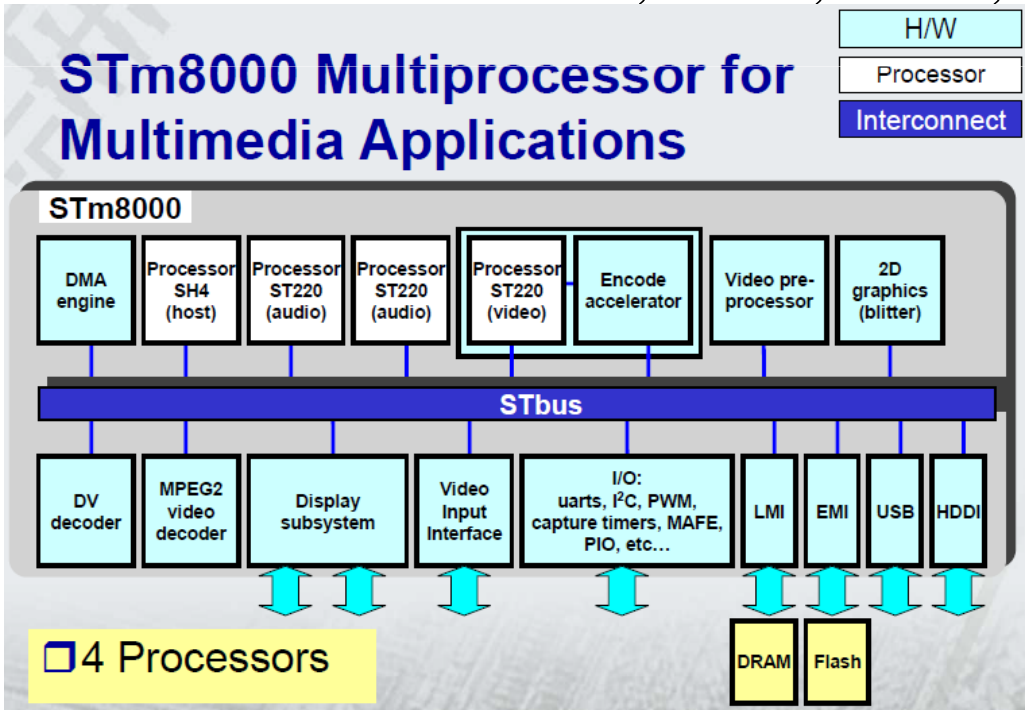


Cible mixte

- Utilisation des blocs matériels spécifiques et logiciels dédiés à une application bien déterminée
- Nouvelles approches de conception : Intégration logicielle/matérielle
  - Travail coopératif entre différentes équipes
  - Co-conception
  - Co-vérification
  - Approche de la réutilisation (IP Reuse)
- Exemple : les systèmes sur Puce : System on Chip (SoC)

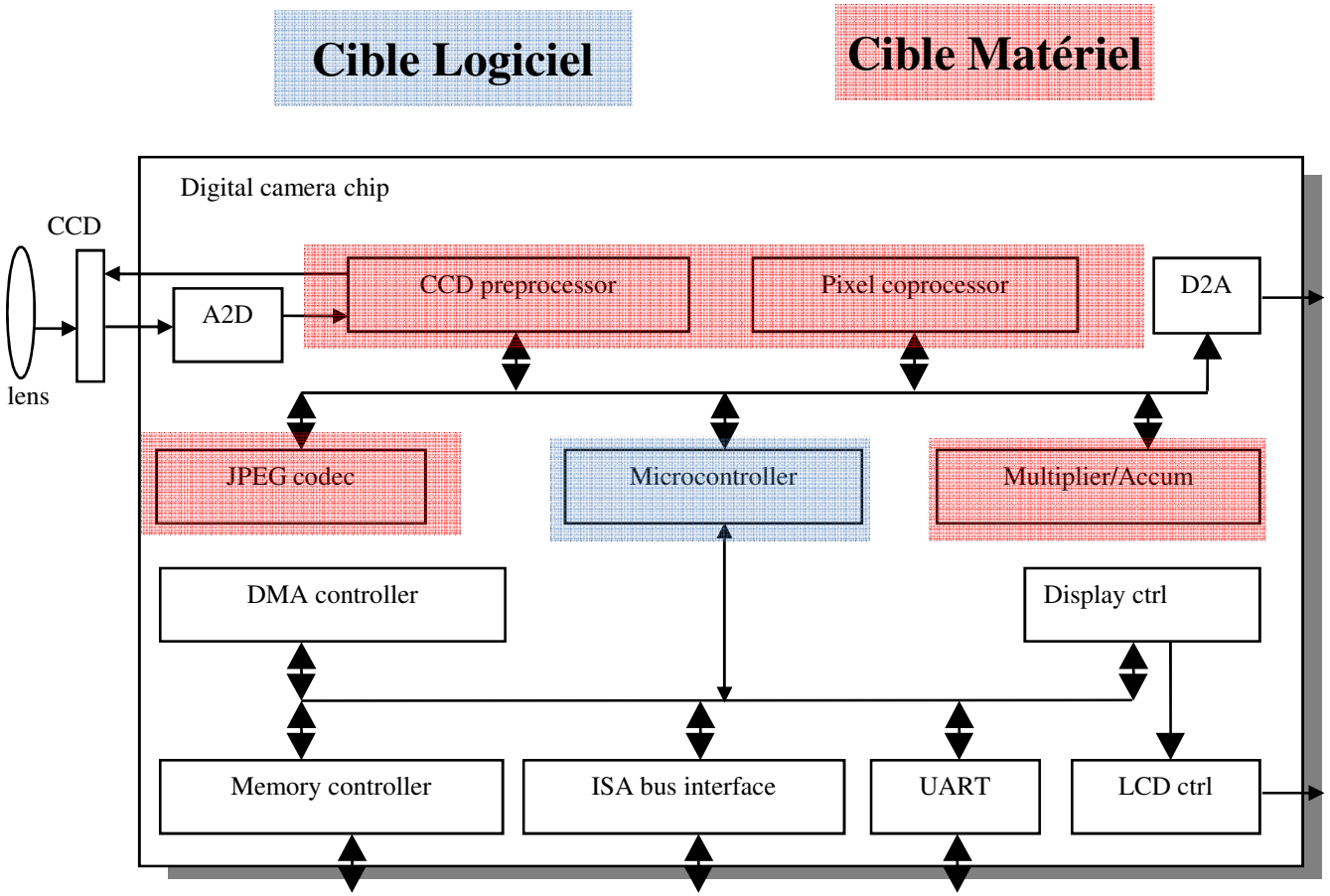
Qu'est ce qu'un système mono-puce ?

- Système mono-puce = System on a Chip (SoC)
  - Système éléctronique complet intégré dans une puce
  - SoC pour les télécoms : un microprocesseur, un DSP, IPs, accélérateurs matériels, RAM, ROM,



- Circuit STM8000 (STMicro)
- Décodeur multi-standards pour lecteurs DVD (audio+vidéo)
- Intègre processeurs, RAM, coprocesseurs dédiés, etc.
- Circuit mixte = numérique + analogique

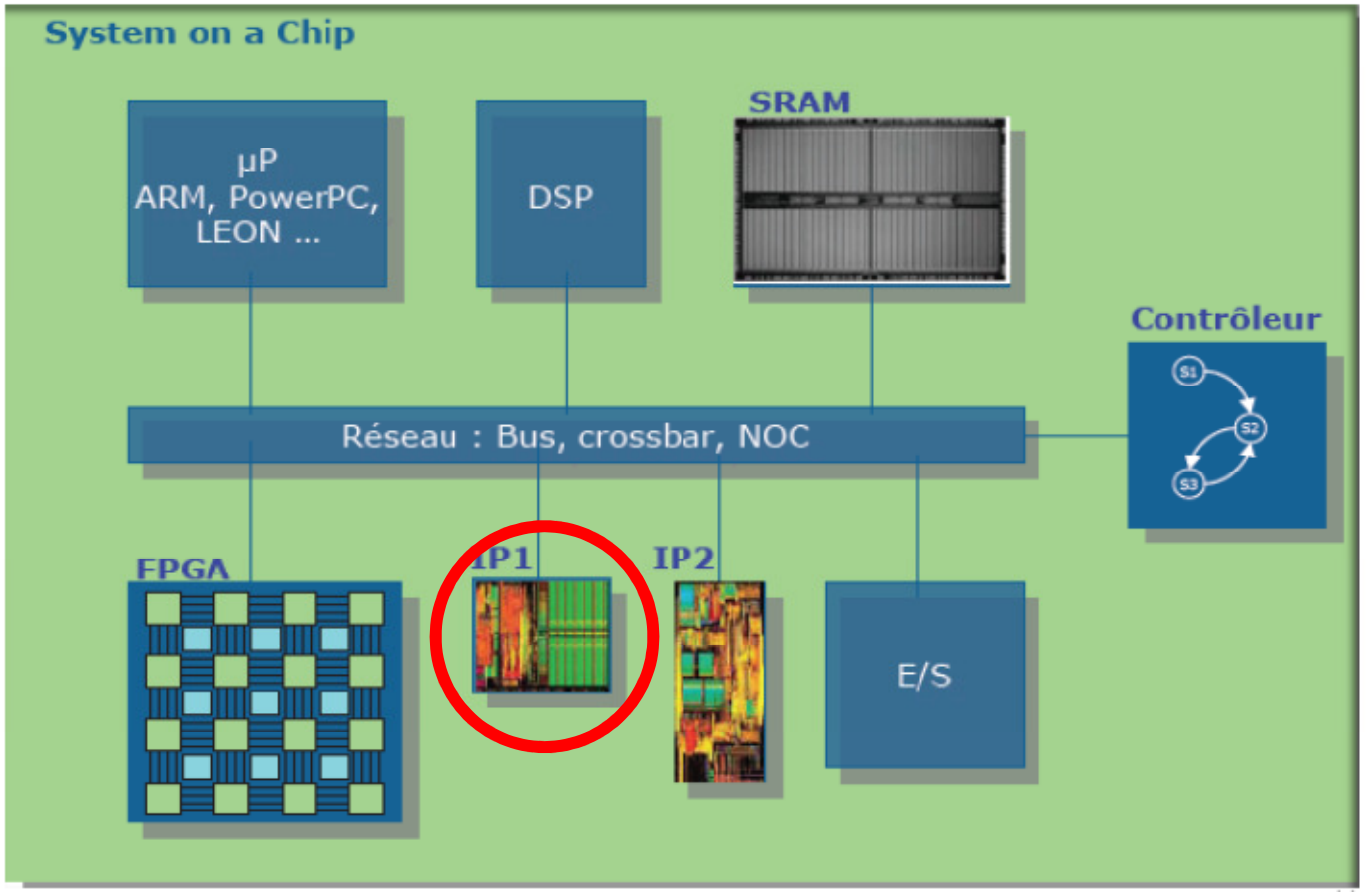
Exemple : Appareil photos numérique



Conception modulaire : Notions d'IP (Intellectual Property)

- IP: ce sont des blocs prêts à être utilisés dans des applications et selon le besoin
  - Permettent un gain en coût et en temps de conception.
  - Blocs fonctionnels complexes réutilisables
    - Hard: déjà implanté, dépendant de la technologie, fortement optimisé
    - Soft : dans un langage de description matériel (VHDL, Verilog…), paramétrables

Cible mixte : Les Systems on Chip



Choix de l'IP

- Intégration de l'IP avec les autres composants du SoC
  - Connaître les fonctionnalités
  - Normalisation des interfaces (OCP: Open Core Protocol)
- Estimation des performances dans un système
  - Performances moyennes (peu optimisé)

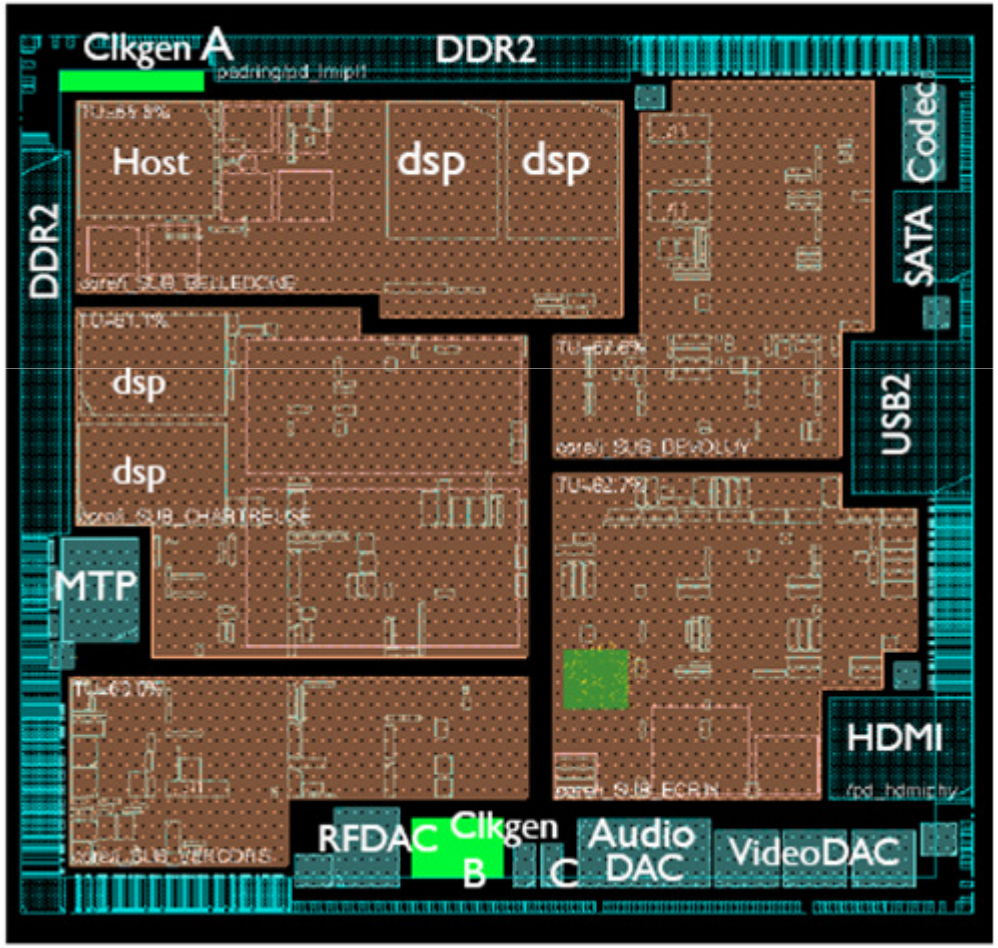
Exemples d'IPs

Communications	Bus Interface	Digital Signal Processing	Processor, Peripheral
ADPCM (u-law, a-law)	PCI Target	Color Space Converter	Nios™ Processor
ATM Controller	PCI Master-Target	Correlator	Tensilica X-tensa Processor
CRC	PCI-X	Digital Modulator	PalmChip Bus
Ethernet MAC (10/100/Gigabit)	CAN Bus	Discrete Cosine Transform	SDRAM Controller
HDLC Protocol Core	IIC Master & Slave	Fast Fourier Transform	DDR-SDRAM Controller
IMA Controller	IEEE 1394	FIR Compiler	QDR-SDRAM Controller
SONET/SDH Framer	PowerPC Bus Arbiter	IIR Filter	8237 DMA Controller
T3/E3 Framer	PowerPC Bus Master	Image Processing Library	8255 Peripheral Interface
Packet Over SONET Processor	PowerPC Bus Slave	NCO	8259 Interrupt Controller
Telephony Tone Generator	USB Function Controller	Reed Solomon Encoder/Decoder	8254 Timer/Counter
Utopia Master & Slave	USB Host Controller	Interleaver/Deinterleaver	8051, 6502, Z80
POS-PHY Interface		Viterbi Decoder	
		Turbo Decoder	

Et plus encore !

Exemple : circuit de décodage vidéo de ST (STi7200)

- Circuit dédié au décodage de la TV numérique HD
  - 150 millions de transistors
  - 4 processeurs (2 DSP pour la vidéo, 1 DSP pour l'audio et 1 généraliste pour la configuration)
  - 36 Softs IPs et 2 Hard Ips
  - 16 IP analogiques,

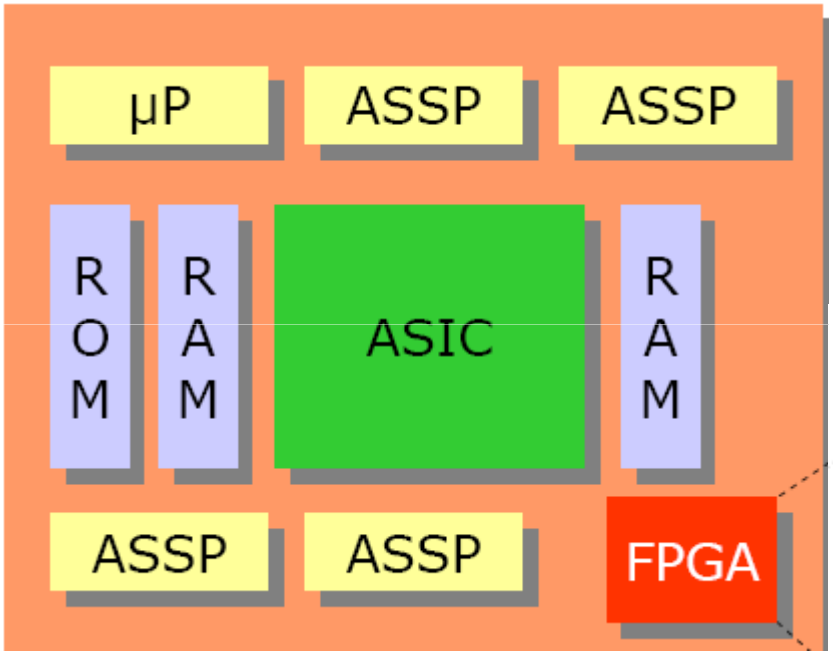


Approches d'implémentation physique d'un SoC

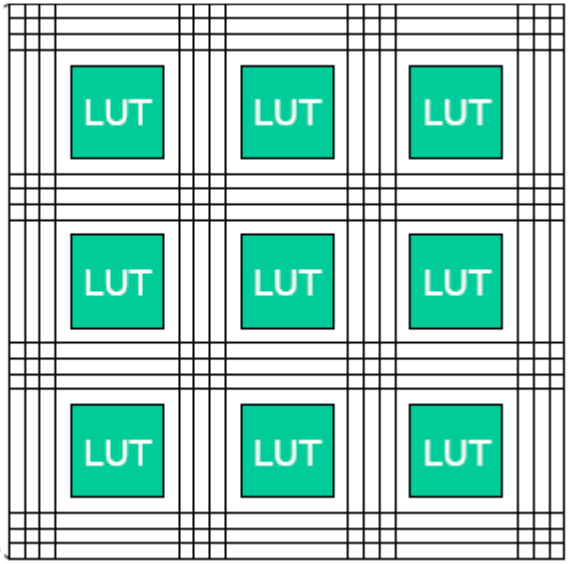
- ASIC
  - Assemblages de composants Processeurs +composants matériels dans un circuit
- FPGA
  - L'utilisation d'un FPGA. On parle alors d'un Système programmable sur puce SoPC

Système programmable sur puce SoPC

1990 : FPGA

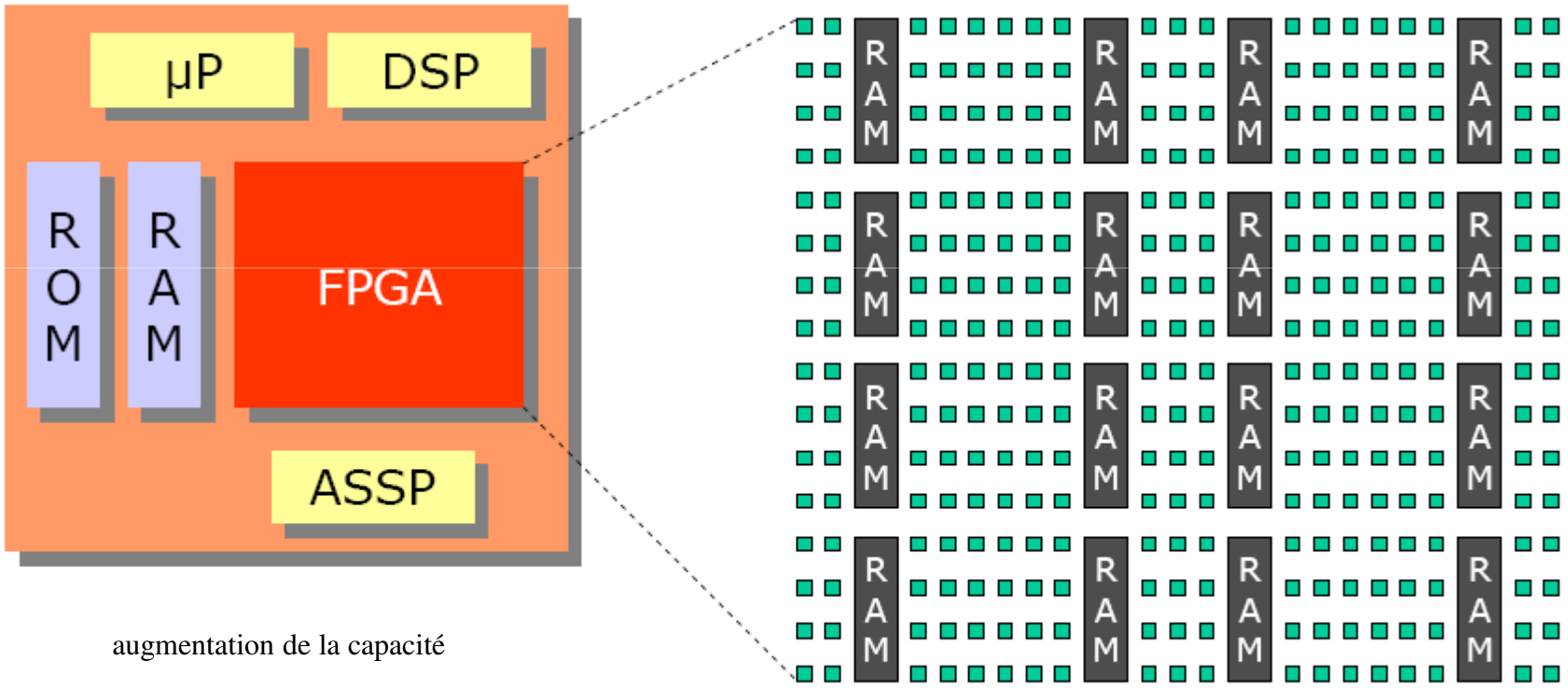


•ASIC :Application Specific Integrated Circuit
•ASSP :Application Specific Standard Product
•FPGA :Field Programmable Gate Array
•RAM :Random Acess Memory
• ROM :Read Only Memory



Système programmable sur puce SoPC

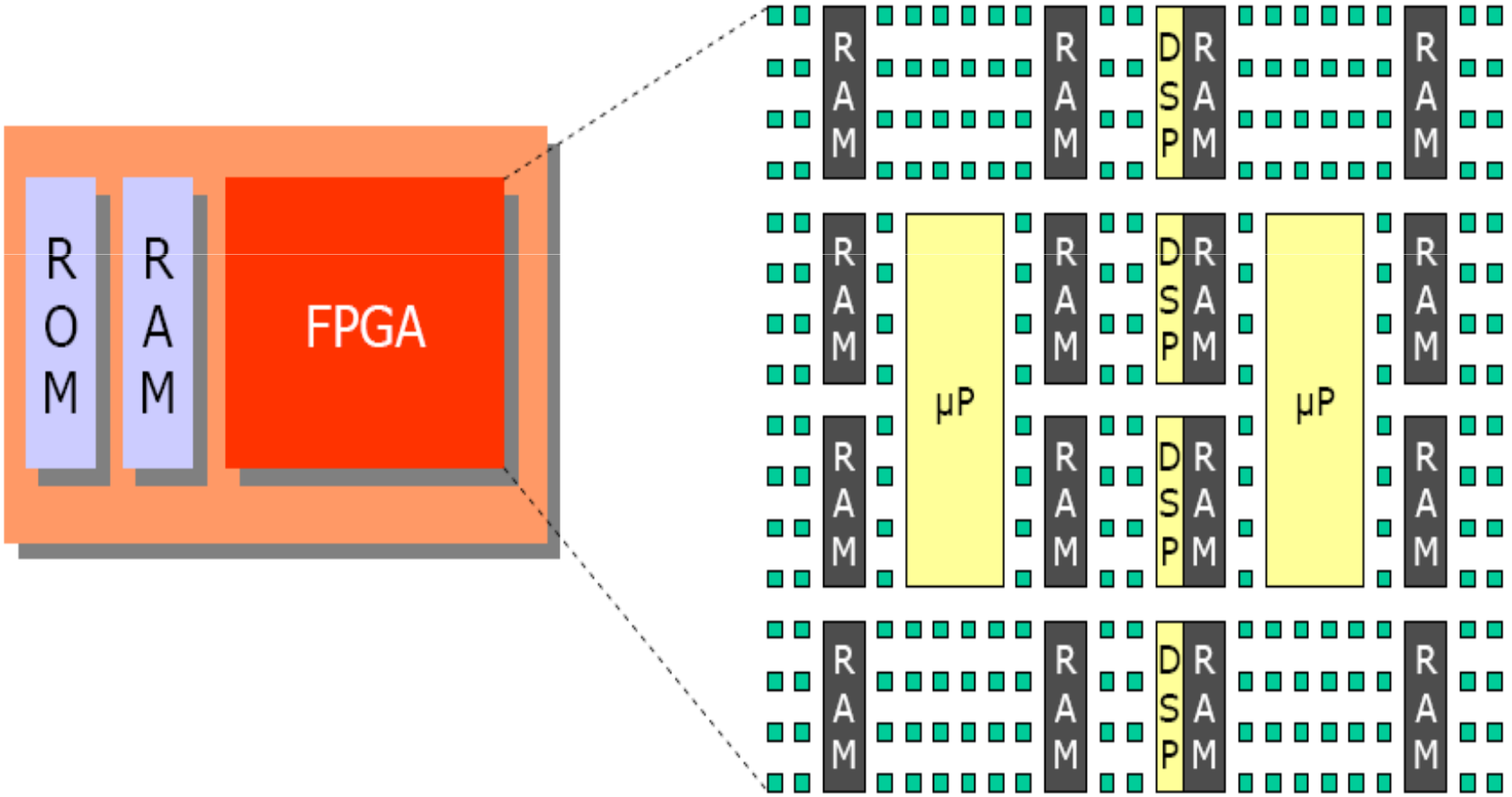
2000 : FPGA+RAM



augmentation de la capacité

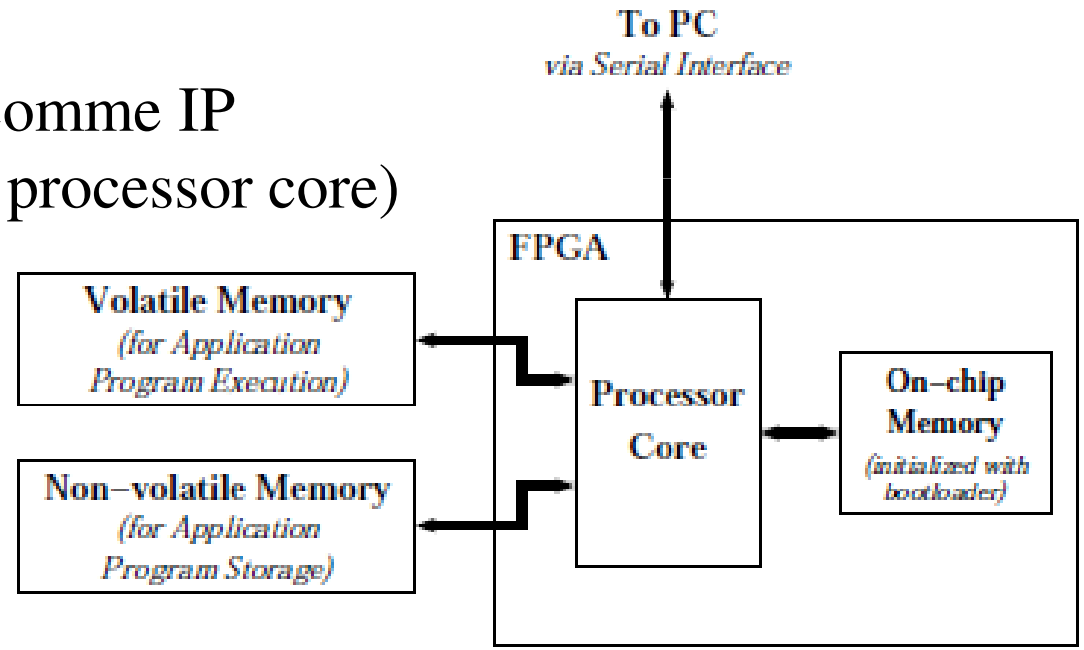
Système programmable sur puce SoPC

2003 : FPGA = FPGA + cœurs de processeurs



System on Programmble Chip (SoPC)

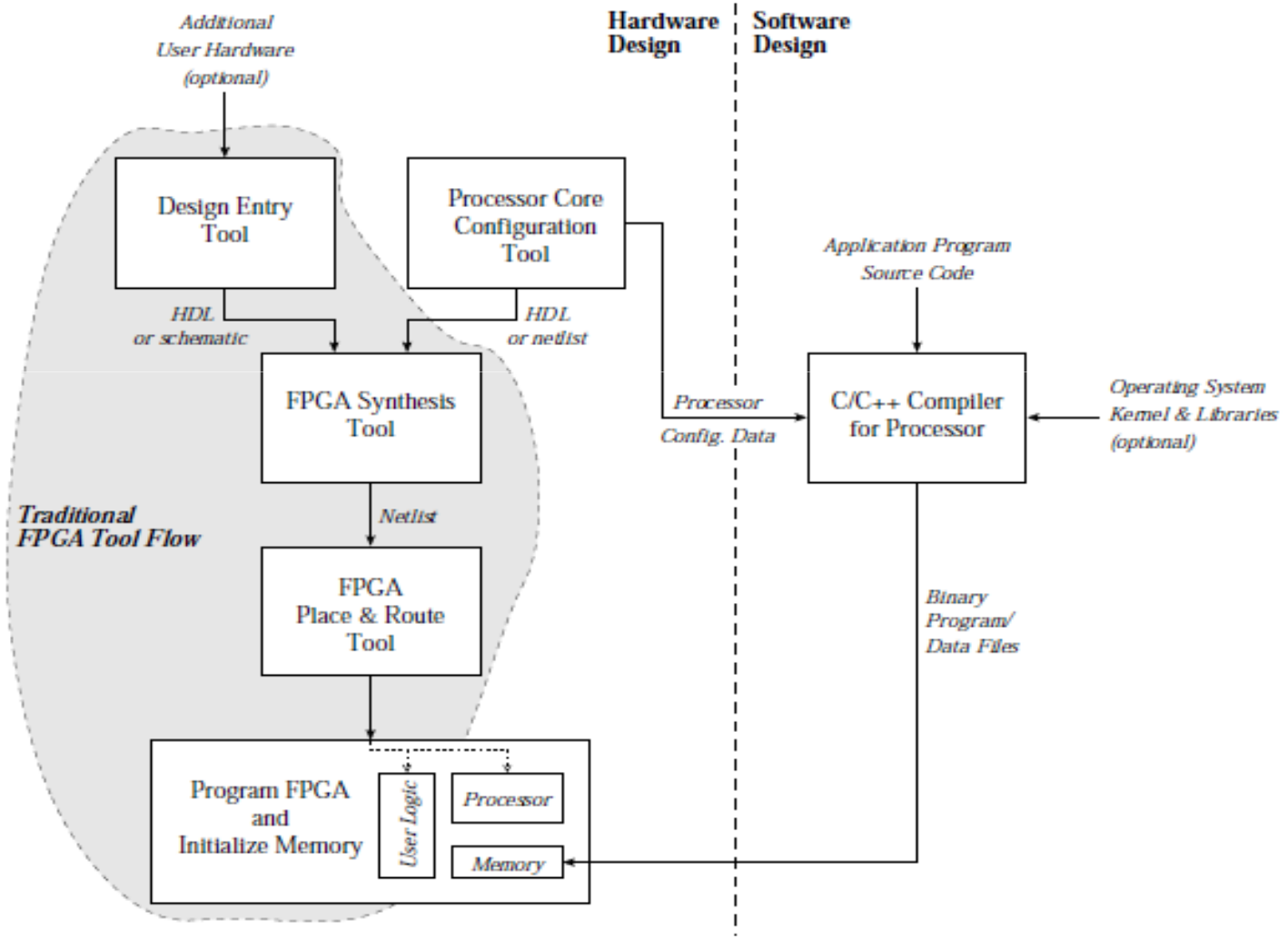
- Utilisation d'un FPGA contenant :
  - Mémoire
  - Eléments logiques
  - Cœur de processeur comme IP (Intellectual property processor core)



Les processeurs pour le SoPC Hardcore/Softcore

- Le choix d'un processeur pour le SoPC peut se faire sur différents critères :
  - Processeur hardcore : implanté dans le circuit électronique en « dur » : on parle de processeur hardcore.
    - Performant mais moins flexible
  - Processeur softcore : implanté dans un FPGA
    - Flexibilité : mise à jour facile
    - Portabilité vers n'importe quel circuit FPGA
    - Migration vers un circuit de type ASIC en cas d'une production en grande série.

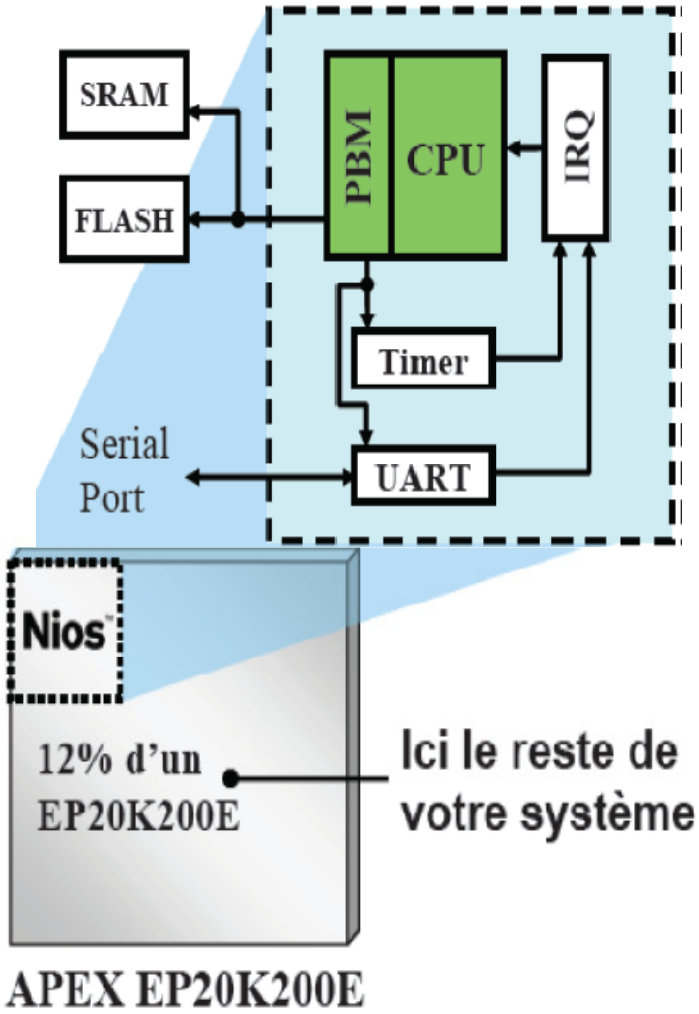
Flot de conception des SoPC



Exemple (1) : approche Altera : IP soft Le ‘NIOS’

- NIOS : cœur de processeur RISC générique optimisé

- Caractéristiques:
  - données sur 16 ou 32 bits, 128, 256 ou 512 registres
  - registres à décalage rapide (1, 3, 7, 15 ou 31 bits/clock)
  - possibilités de lui adjoindre des périphériques (UART, RAM, ROM)



Exemple (2) : Approche Xilinx

- Hardcore : Power PC (Xilinx VIRTEX II PRO (XC2VP):
  - une matrice configurable
    - 1 500 000 de portes
    - De 216 Kbits à 8 Mbits de mémoire
    - De 204 à 1164 I/Os
  - 1, 2 (ou 4) cœurs de processeur PowerPC 405 (32 Bits) à 400MHz (hard)
  - 16 Koctets de cache instructions
  - 16 K octets de cache données
  - Prix: ~ 1 500 $ max
- Softcore
  - MicroBlaze, picoblaze