



Concepts et Pratique des Processeurs Numériques des Signaux (DSP)

Chiheb REBAI, Nadia Khouja

chiheb.rebai@supcom.rnu.tn, nadia.khouja@supcom.rnu.tn



Mise en évidence de:

- Fonctionnalités de base des processeurs numériques des signaux (DSP)
- Architectures des cores et des périphériques des DSPs
- Méthodologies et Outils de développement, de vérification et de mise au point

Compétences à acquérir:

- Maîtrise des architectures matérielles des DSPs
- Maîtrise du jeu d'instructions pour la programmation des DSP
- Spécification des applications d'implantation sur DSP
- Règles de structuration et de développement de codes DSP
- Maîtrise des outils de développement et de test des applications DSP

Programme du module



Séance 1	<i>Concepts de base des DSP: définition et architecture générique</i>	C. Rebai
Séance 2	<i>Étude des DSP AD-Blackfin: architecture et jeu d'instructions</i>	C. Rebai
Séance 3	<i>Initiation à l'environnement de développement intégré Visual DSP++</i>	C. Rebai
Séance 4	<i>Application 1: Familiarisation avec le jeu d'instructions du DSP Blackfin</i>	C. Rebai M. Attia
Séance 5	<i>Application 2: Implantation d'un filtre à réponse impulsionnelle finie (FIR)</i>	C. Rebai M. Attia
Séance 6	<i>Application 3: Chargement des porteuses d'un modulateur DMT</i>	C. Rebai M. Attia
Séance 7	<i>Application 4: Fonctions de traitement d'images</i>	C. Rebai M. Attia



Partie 1

Concepts de base des DSP

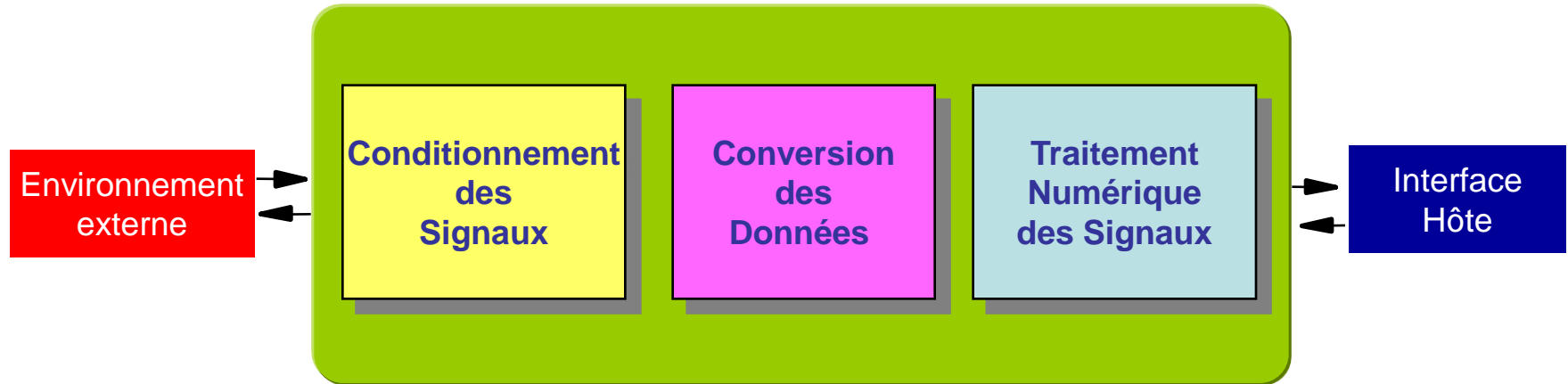
- Définition des DSP
- Applications des DSP
- Représentations et traitements des données
- Fonctionnalités de base des DSP
- Classification des technologies DSP



Définition des DSP



Définition d'un DSP



- Composant électronique utilisant conjointement du matériel (Hard) et du logiciel (Soft) pour mettre en œuvre une application de traitement de signal
- Système DSP: interagit fortement avec son environnement (contraintes de temps réel et de dynamique des phénomènes physiques)
- Composants du DSP (unités de traitement, contrôleurs, registres, bus, mémoire, ports E/S, périphériques,...): fixent les performances expérimentales du système
- Ressources logicielles (jeu d'instructions, moniteur, OS) offrent la flexibilité au système

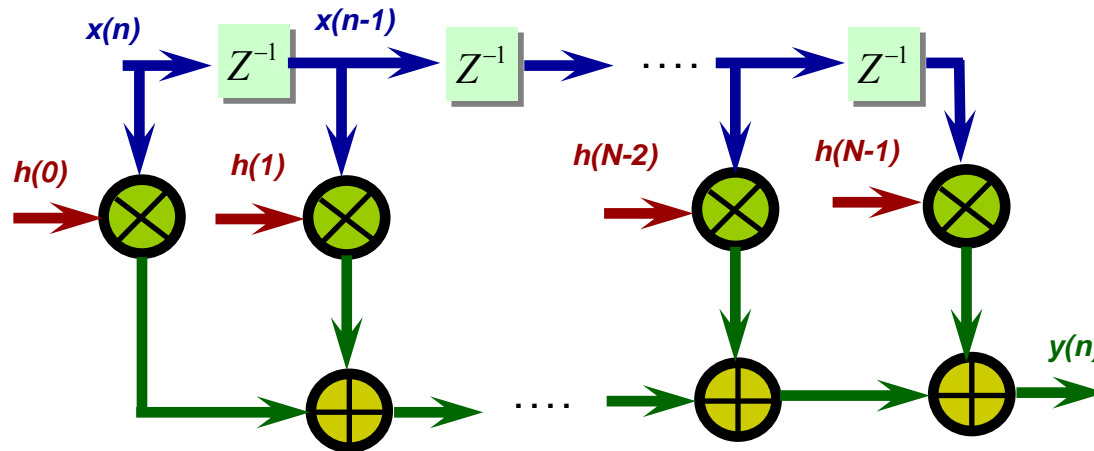
Opération «DSP» fondamentale



Produit de
convolution

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

Filtre FIR



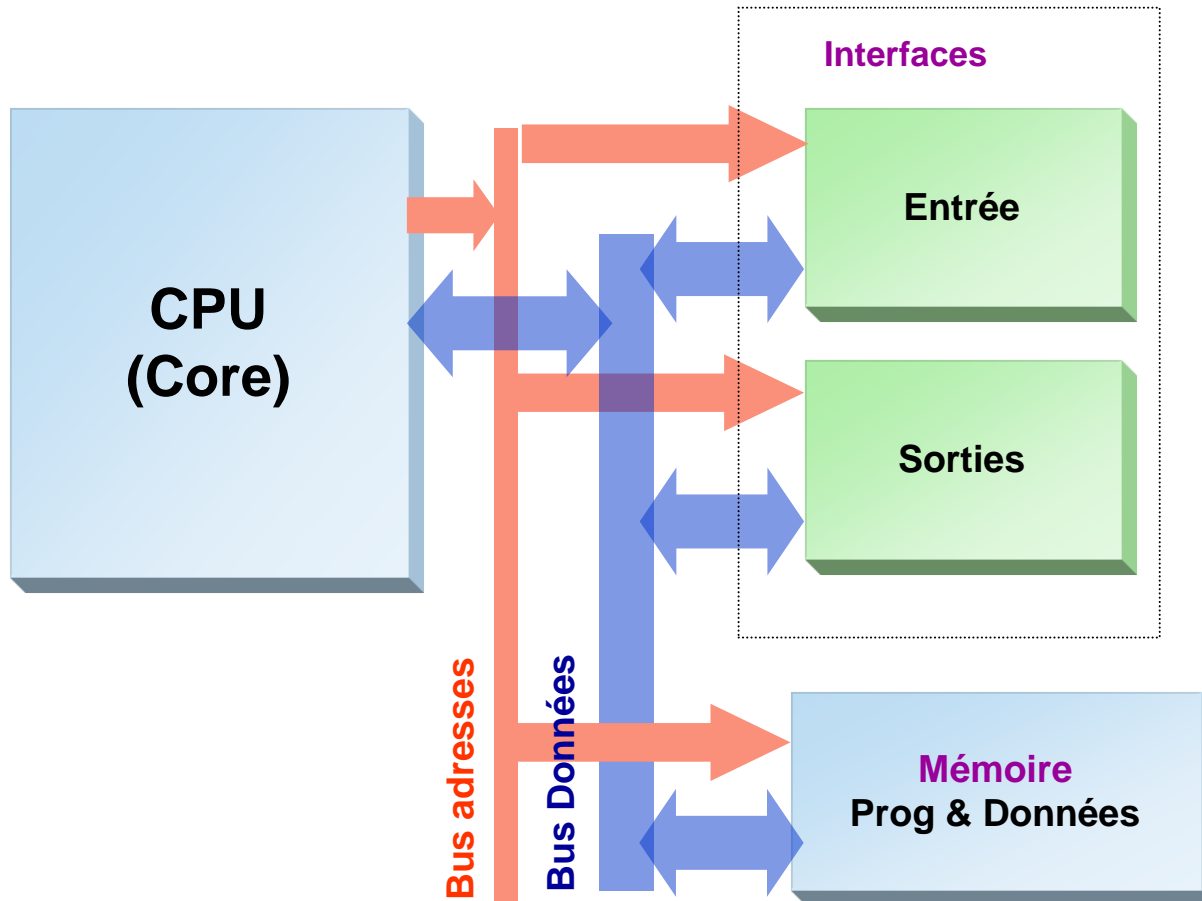
Les opérations pour chaque **TAP** :

- ▶ Two data fetches
- ▶ Multiply
- ▶ Accumulate
- ▶ Memory write-back to update delay line

Objectif

1 FIR Tap / DSP instruction cycle

Architecture Von Neumann



Système à usage général

Une mémoire pour le
code et les données

Double utilisation des bus
données et adresses



1. Instruction fetch
2. Data₁ fetch
3. Data₂ fetch



Plusieurs cycles !

Limitations de l'architecture Von Neumann



$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

Implantation avec un microprocesseur x86 d'Intel

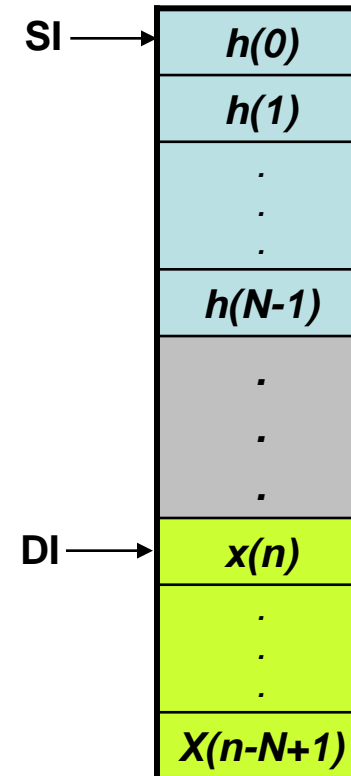
DX ← 0

SI ← @coef

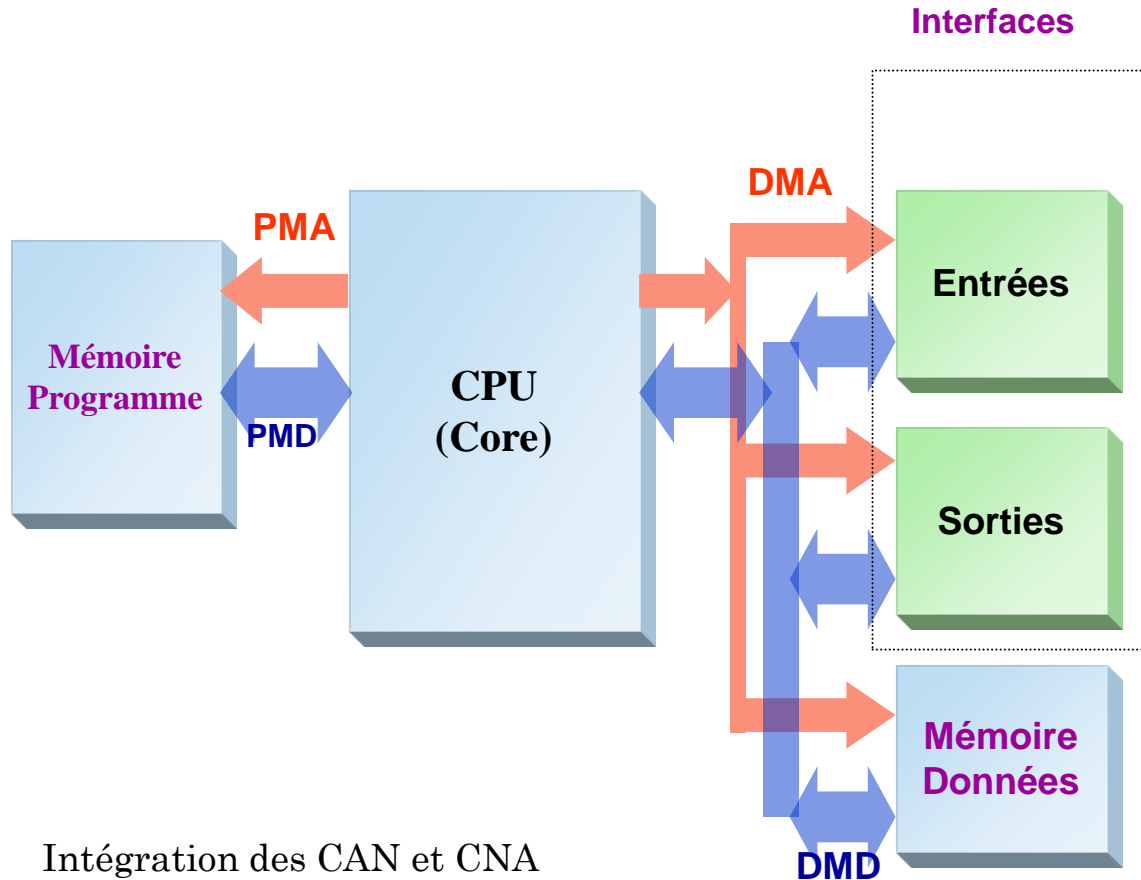
DI ← @samples

CX ← N

tap : MOV AX, [SI] ; lecture coefficients
 MUL [DI] ; multiplication AX*[DI] → AX
 ADD DX, AX ; accumulation
 INR SI ; avancement pointeurs
 INR DI
 LOOP **tap**



Architecture Harvard

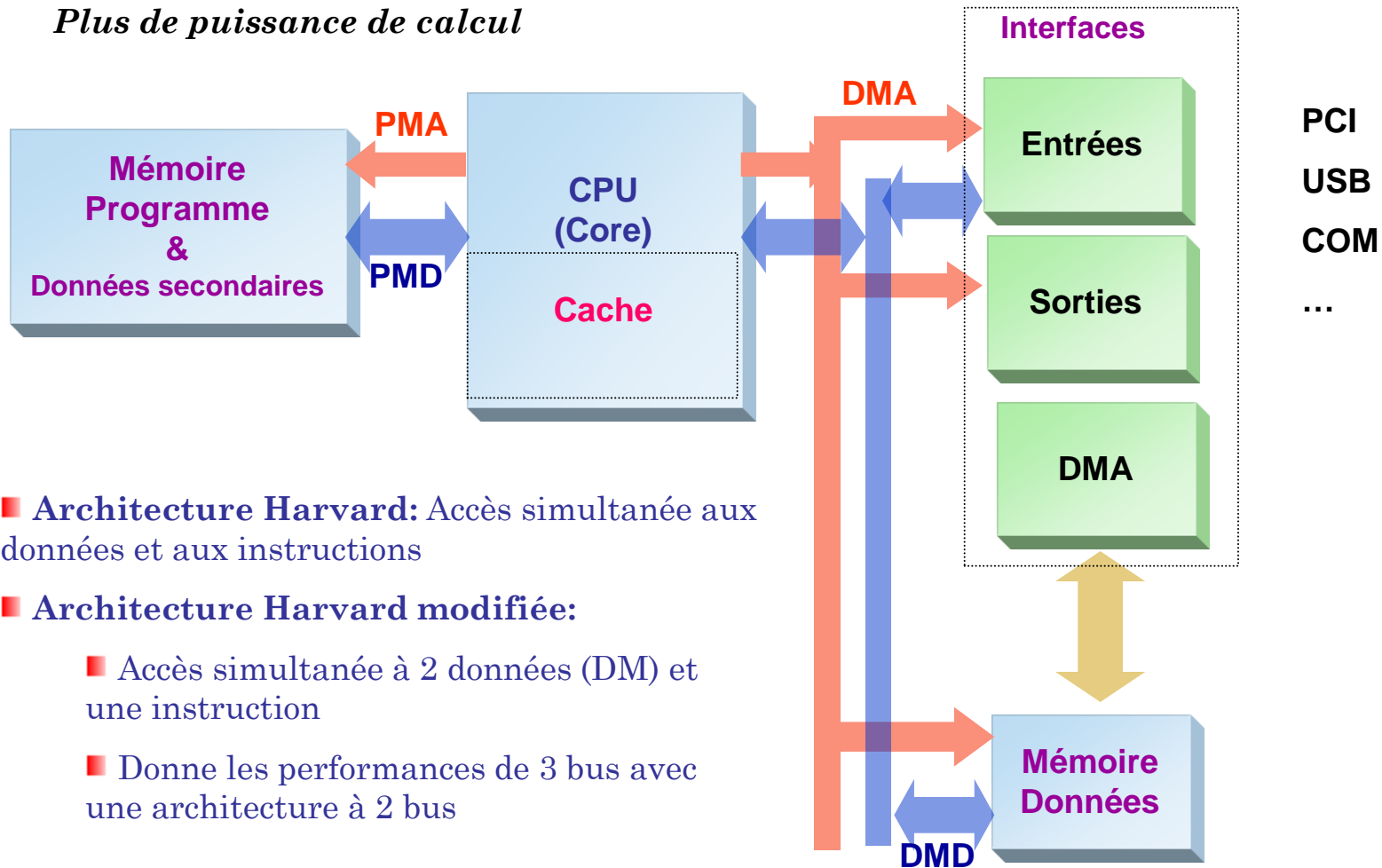


Intégration des CAN et CNA
dans certains DSP

- Mémoire PM et DM séparées
- Bus indépendants pour Chaque mémoire
- **En 1 cycle :**
Instr. Fetch & Data fetch
- Memory Mapped IO
- Core : *Chemins de données optimisés, MAC, adressage approprié des données*
- Software : *instructions dédiées aux applications (FFT, video, parole,...)*



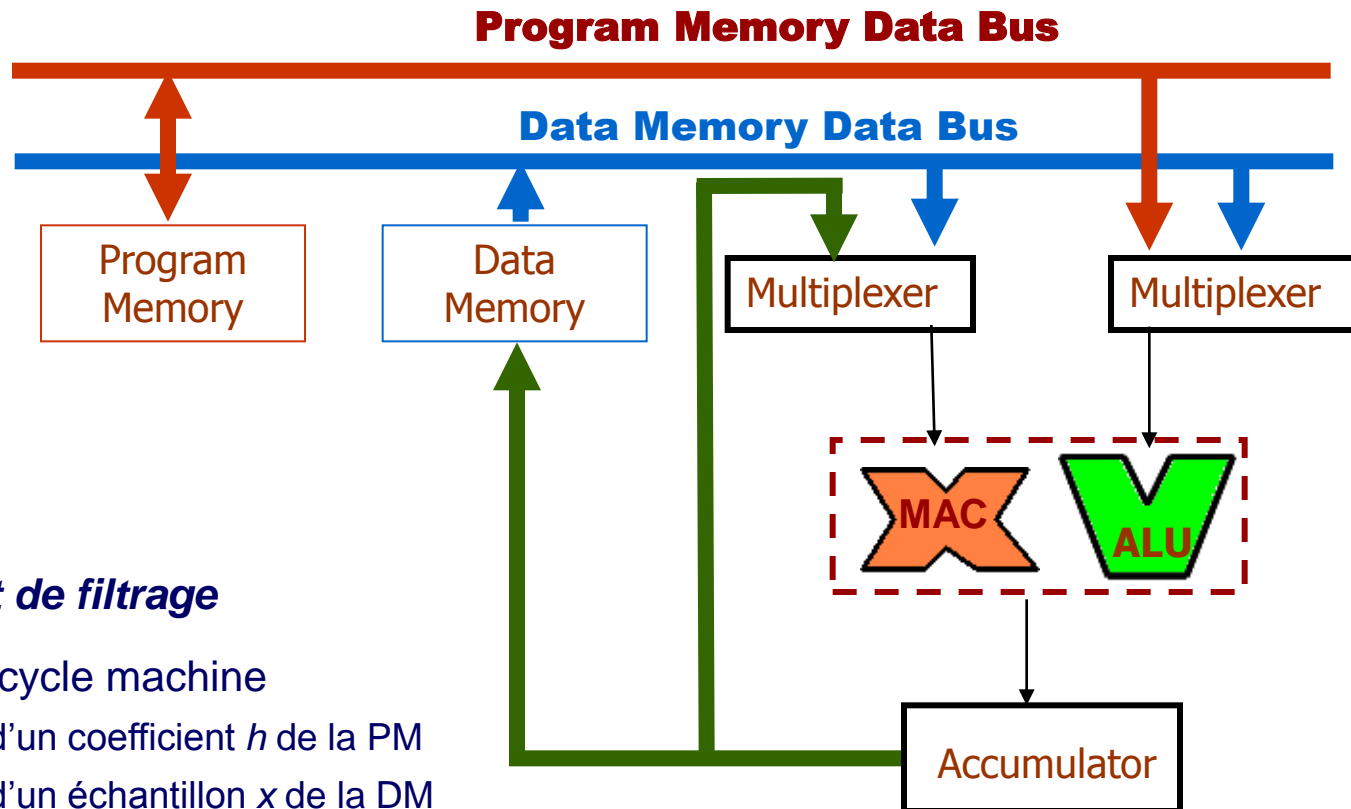
Plus de puissance de calcul



■ **Architecture Harvard:** Accès simultanée aux données et aux instructions

■ **Architecture Harvard modifiée:**

- Accès simultanée à 2 données (DM) et une instruction
- Donne les performances de 3 bus avec une architecture à 2 bus



Traitement de filtrage

En un seul cycle machine

- ➡ Lecture d'un coefficient h de la PM
- ➡ Lecture d'un échantillon x de la DM
- ➡ Multiplication et accumulation



Microcontrôleurs

- **Architecture:** CPU, RAM, ROM, interfaces série / parallèle, timer, circuits d'interruptions
- **Applications:** contrôle / commande de processus
- **Caractéristiques:** pas d'exigence de vitesse, jeu d'instructions compact
- **Exemples:** 8051, 68HC11, PIC,...

Microprocesseurs

- **Architecture:** CPU mono-chip, nécessite des circuits additionnels externes
 - **RISC:** Reduced Instruction Set Computer
 - **CISC:** Complex Instruction Set Computer
- **Exemples:** Pentium-Series, PowerPC, MIPS,...



Processeurs Numériques des Signaux (DSP)

- **Architecture:**
 - CPU optimisée pour traitement mathématique temps réel rapide et répétitif
 - RAM, ROM, interfaces série / parallèle, timer, circuits d'interruptions
- **Exemples:** ADSP-21xx, AD-BF-5xx, AD-TS-xxx, TMS320Cxx,...





Applications des DSP



Domaines d'application



■ Produits électroménagers

Cafetière, machines à laver, fours à micro-onde,...



■ Électronique grand public

Caméras et appareils photos numériques, décodeurs vidéo, téléphones portables, PDA,...



■ Automobile

ABS, GPS, contrôle moteur, informatique de confort,

■ Avionique, spatial, procédés industriels

Systèmes de navigation aérienne et maritime, systèmes de contrôle des procédés industriels

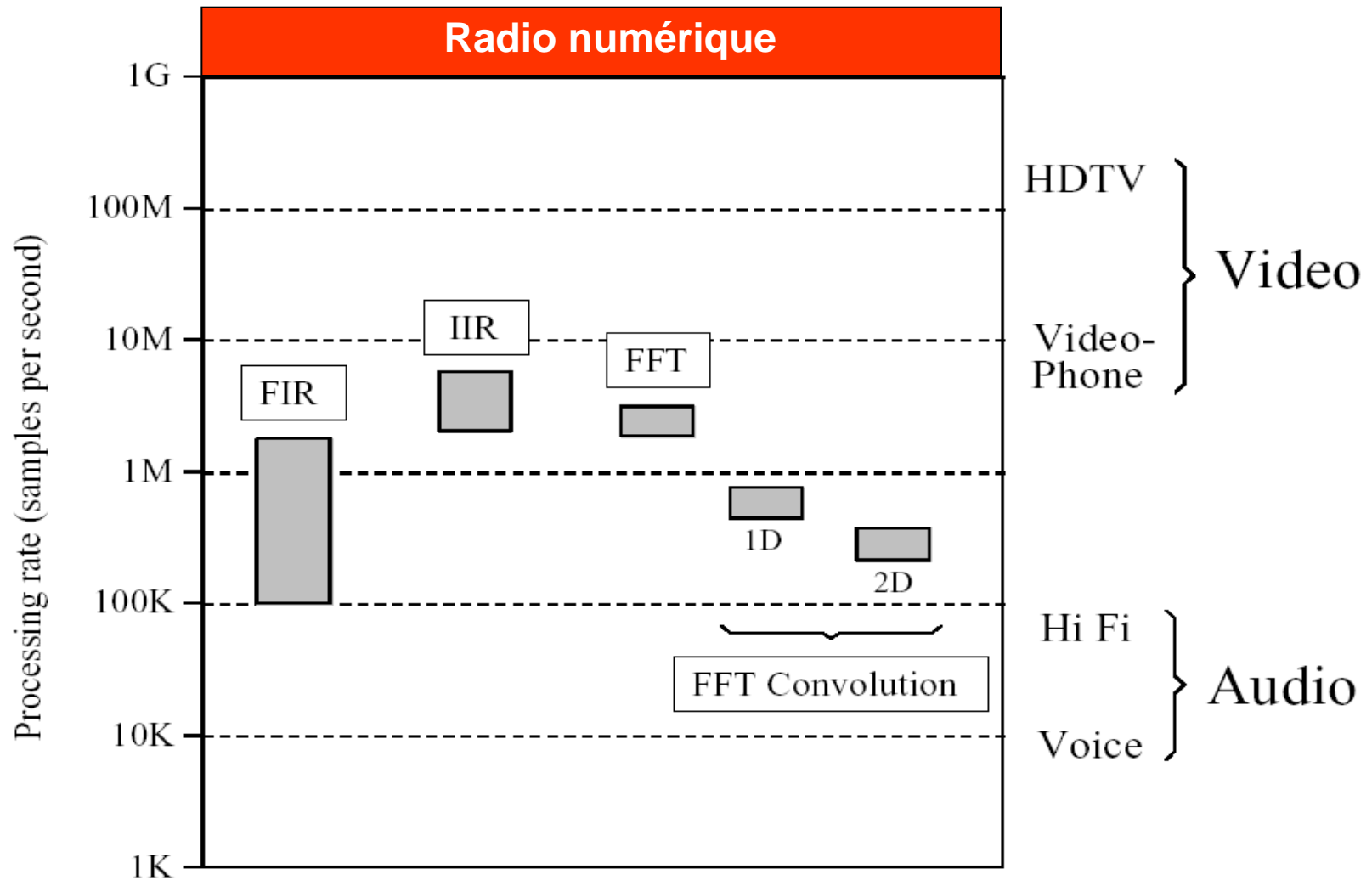


■ Télécommunications et informatique

Terminaux, nœuds de transfert, équipement de transmission, périphériques informatiques



Capacité de traitement DSP





■ LAN câblé

RNIS, xDSL (30 – 500 MIPS), ATM, IP (3000 MIPS),...

■ LAN sans fils

WLL (DECT: 30 MIPS), WLAN (WiFi: 30 – 1000 MIPS),
WiMax,... UWB,...

■ WAN

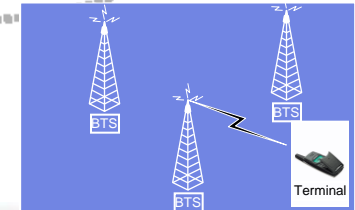
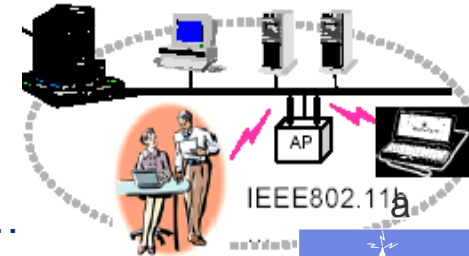
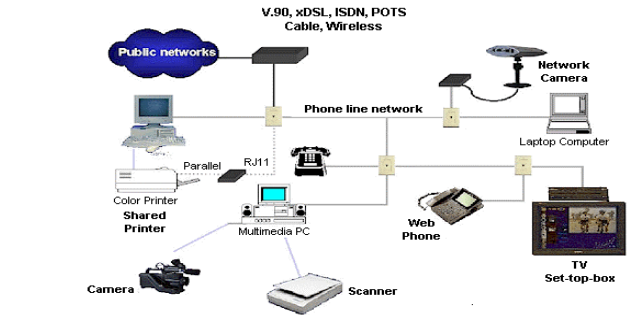
GSM (30 MIPS), UMTS (300 MIPS), MBS (3000 MIPS),...

■ Applications multimédia

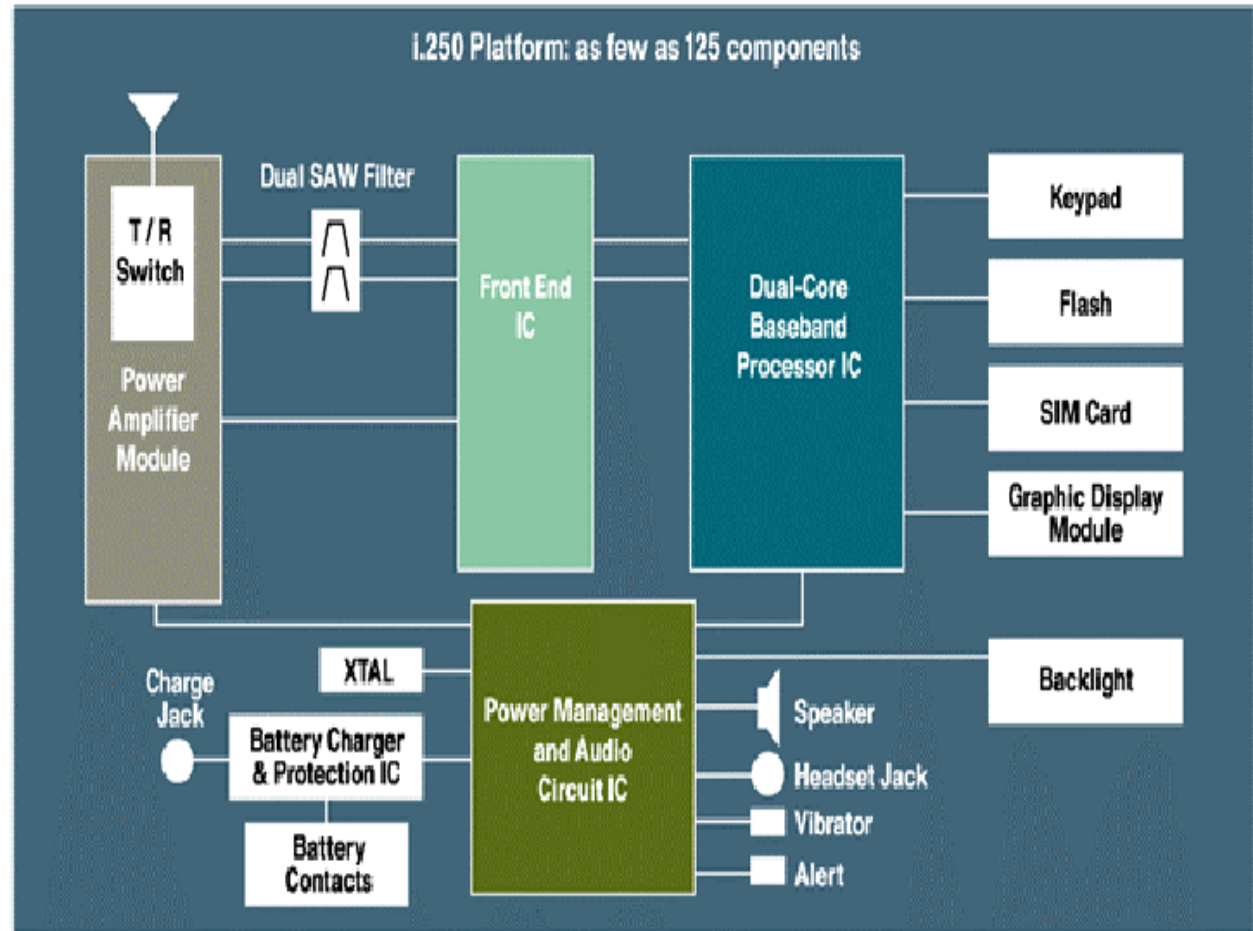
Communication: Visiophone, TV numérique, Web-phone,
vidéosurveillance,...

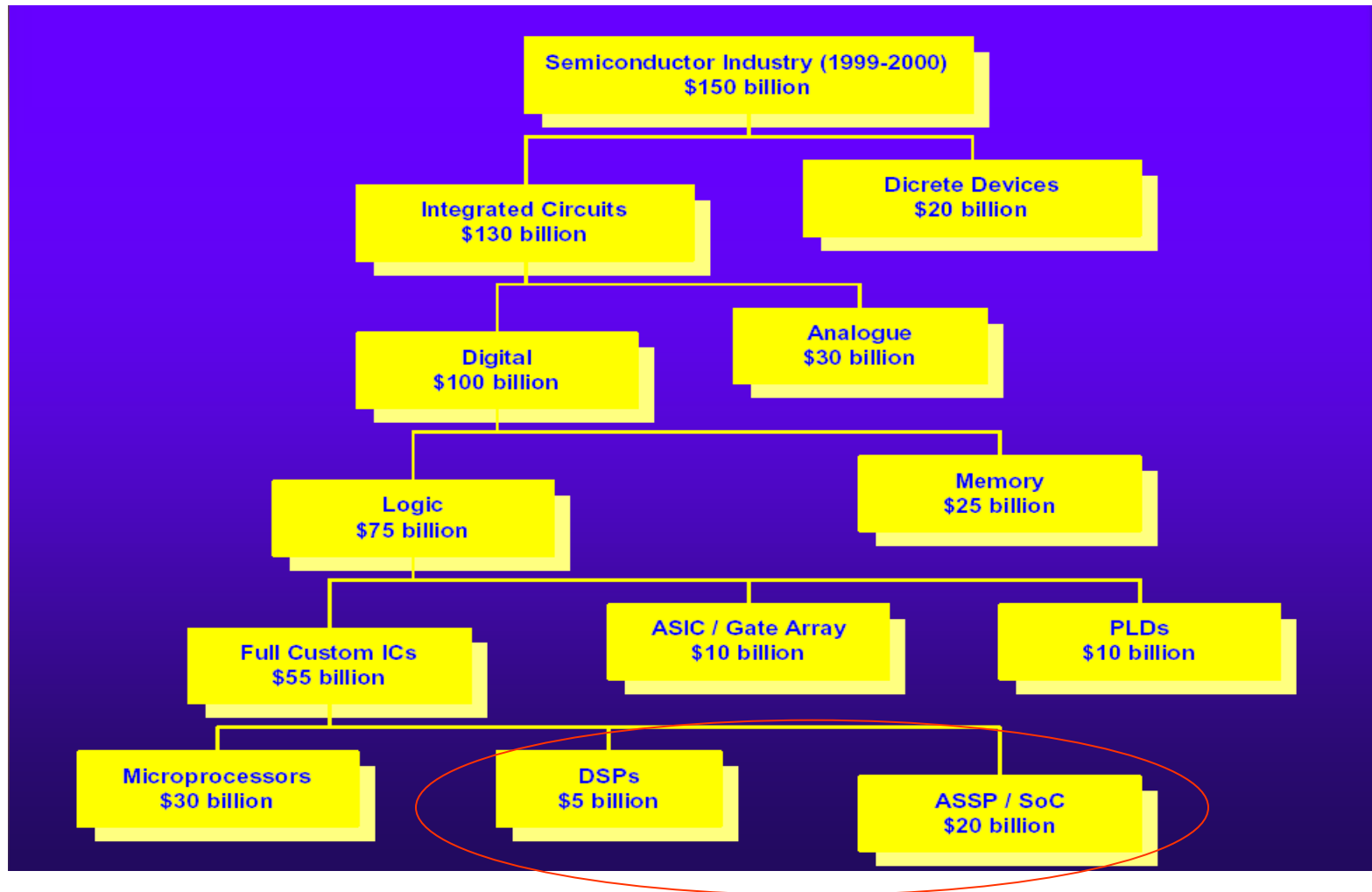
Diffusion: Audio (DAB), Vidéo (DVB),...

Manipulation: services interactifs, gestion de multiples flots
d'information (MPEG4),...



Exemple: téléphone portable







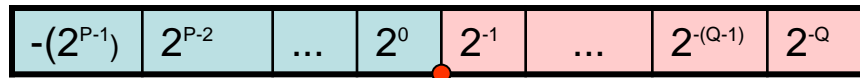
Représentations et traitement des données



Représentation en virgule fixe



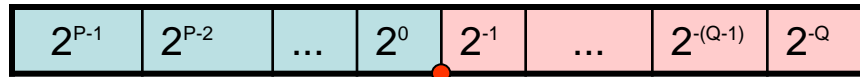
Format P.Q sur (P+Q) bits : *Partie entière sur P bits & Partie fractionnelle sur Q bits*



Signed P.Q (*complément à 2*)

Partie entière

Partie Fractionnelle



Unsigned P.Q

Cas d'un DSP 16 bits

Unsigned Integer 16.0



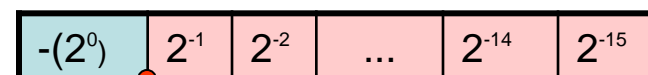
Unsigned Fractional 0.16



Signed Integer 16.0



Signed Fractional 1.15

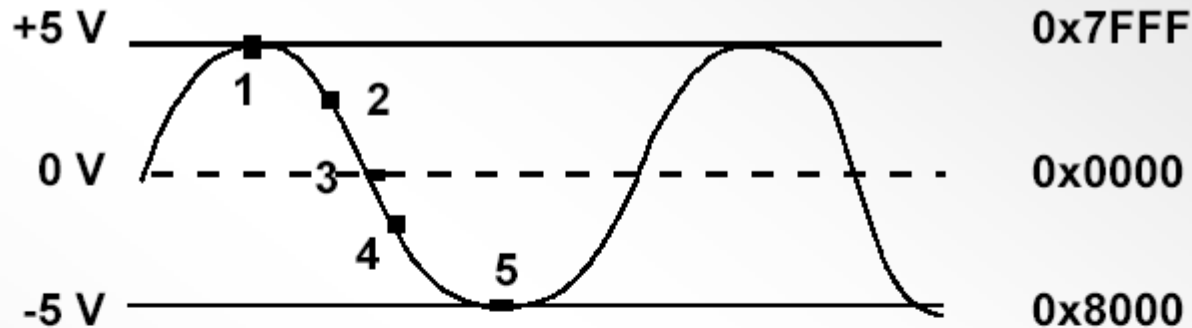


Plages de variation en format 16 bits



FORMAT		Largest Positive Value (0x7FFF) In Decimal	Largest Negative Value (0x8000) In Decimal	Value of 1 LSB (0x0001) In Decimal
1.15	Fractional	0.999969482421875	-1.0	0.000030517578125
2.14		1.999938964843750	-2.0	0.000061035156250
3.13		3.999877929687500	-4.0	0.000122070312500
4.12		7.999755859375000	-8.0	0.000244140625000
5.11		15.999511718750000	-16.0	0.000488281250000
6.10		31.999023437500000	-32.0	0.000976562500000
7.9		63.998046875000000	-64.0	0.001953125000000
8.8		127.996093750000000	-128.0	0.003906250000000
9.7		255.992187500000000	-256.0	0.007812500000000
10.6		511.984375000000000	-512.0	0.015625000000000
11.5		1023.968750000000000	-1024.0	0.031250000000000
12.4		2047.937500000000000	-2048.0	0.062500000000000
13.3		4095.875000000000000	-4096.0	0.125000000000000
14.2		8191.750000000000000	-8192.0	0.250000000000000
15.1		16383.500000000000000	-16384.0	0.500000000000000
16.0	Integer	32767.000000000000000	-32768.0	1.000000000000000

Exemple de numérisation de signal



FORMAT

		16.0	1.15
1)	0x7FFF	= 32767 -> 5 V	0.999969482... -> 5 V
2)	0x3FFF	= 16383 -> 2.5 V	0.499969482... -> 2.5 V
3)	0x0000	= 0 -> 0 V	0.0000000... -> 0 V
4)	0xCCCD	= -13107 -> -2.0 V	-0.399993986... -> -2.0 V
5)	0x8000	= -32768 -> -5.0 V	-1.0000000.... -> -5.0 V



Extension de signe : Augmenter la précision sans perdre la valeur

décimal	4 bits	8 bits
0	0000	00000000
1	0001	00000001
-1	1111	11111111

Overflow

0011 (3 décimal)

+ 0111 (7 décimal)

= 1010 (-6 décimal ??)

cas d'un signal inversion de polarité

Mise à l'échelle (scaling)

faible résolution de représentation et qqes bits fort poids à 0

Bits de garde

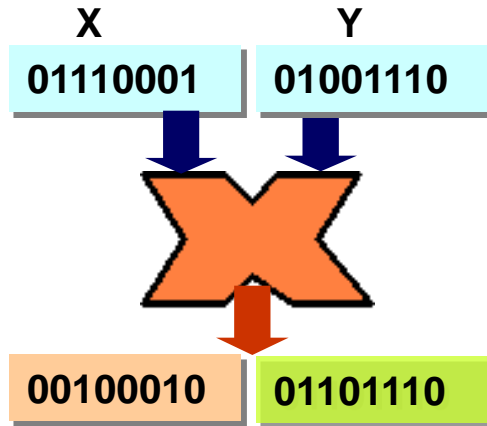
Prévoir les débordements et utiliser l'extension de signe

Saturation :

au maximum négatif ou au maximum positif

Sur 16 bits : 0x7FFF ou 0x8000

Multiplication à virgule fixe



Cas Integer: $(8.0) \times (8.0) = (16.0)$

$78 \times 113 = 8814 = 00100010 \ 01101110$

Aucun ajustement nécessaire. Mais signed ou unsigned ?

Cas Fractional

Règle générale : $(P.Q) \times (P'.Q') \Rightarrow (P+P').(Q+Q')$

Unsigned : $(0.8) \times (0.8) = (0.16)$

$01001110 = 0.3046875$

$01110001 = 0.44140625$

$0010001001101110 = 0.134490966796875$

Aucun ajustement nécessaire

Signed : $(1.7) \times (1.7) = (2.14)$

$01001110 = +0.609375$

$01110001 = +0.8828125$

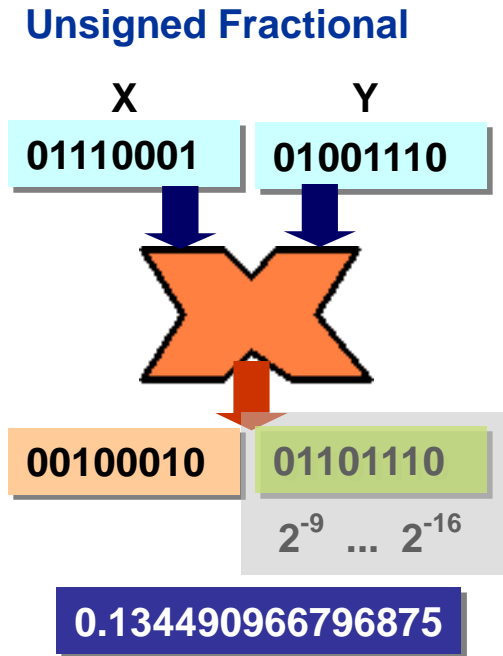
$0010001001101110 = +0.26898193359375$

Décalage à gauche $\Rightarrow (1.15)$

$0100010011011100 = +0.5379638671875$

Il faut spécifier le format pour la Multiplication: IS,IU,FS,FU

Arrondissement et troncature



Troncature



Arrondissement (*Rounding*)



Si $xxxxxxx < (10000000 = 0.5 \cdot 2^{-8})$ → 00100010

0.1328125

Si $xxxxxxx \geq (10000000 = 0.5 \cdot 2^{-8})$ → 00100011

0.13671875

L'arrondissement ou la troncature peuvent être spécifiés dans l'instruction de multiplication



Virgule flottante (floating point) : IEEE Standard 754



Simple précision : n=32 bits

C sur 8 bits

M sur 23 bits

$$N = (-1)^S * (1+M) * 2^{C-127}$$

Double précision : n=64 bits

C sur 11 bits

M sur 52 bits

$$N = (-1)^S * (1+M) * 2^{C-1023}$$

DSP à virgule flottante: *Gestion de la virgule par le Hardware*

DSP à virgule fixe : *des Librairies (Software) pour gérer la virgule flottante*



Fonctionnalités de base des DSP





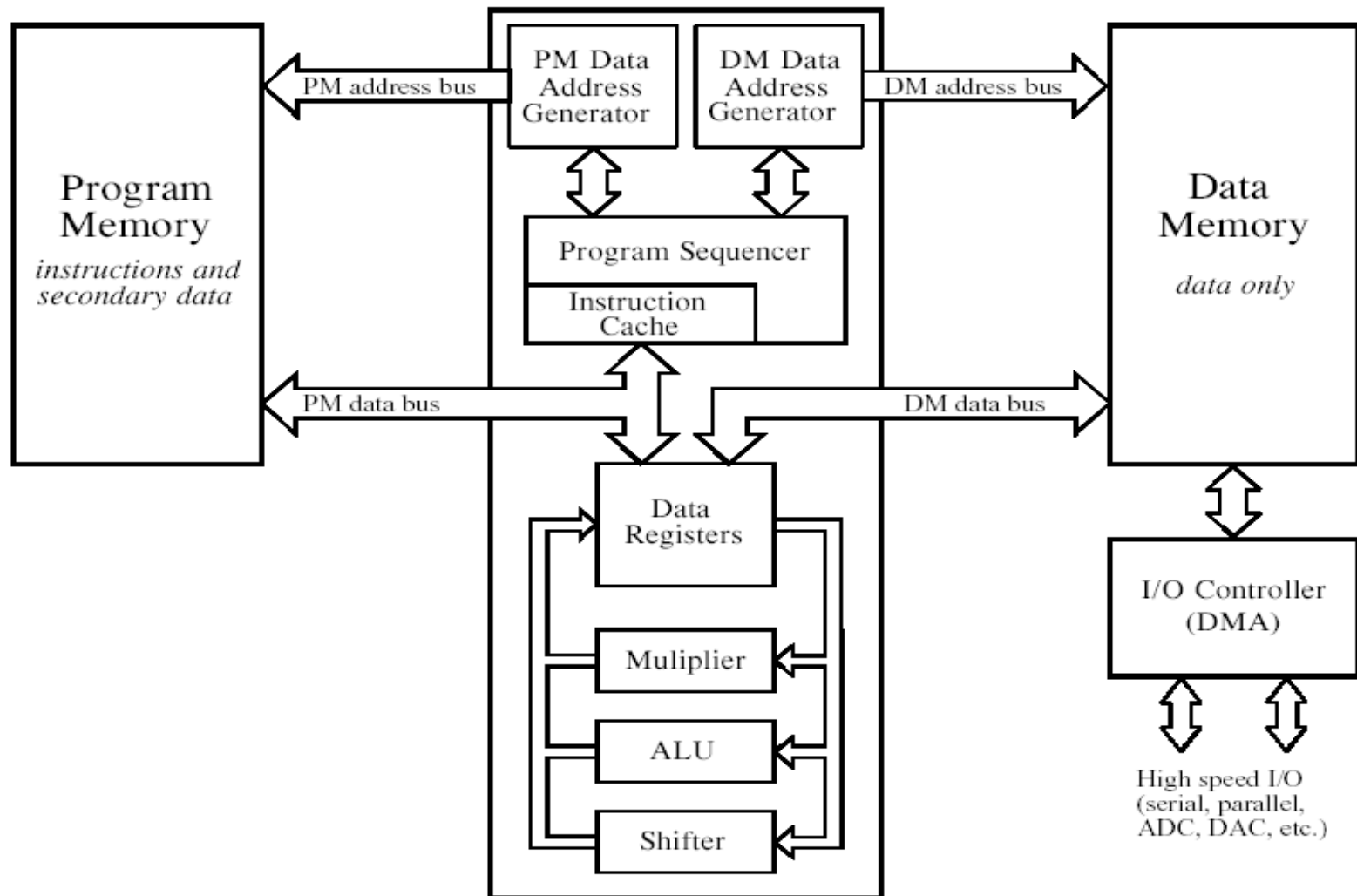
- **Rapidité et flexibilité des unités de traitement arithmétique**
 - Unités de traitement parallèle: *ALU, MAC, SHIFTER*
 - Fonctionnement parallèle du *MAC* et *ALU*
 - Pas de pipeline arithmétique

- **Trafic sans contraintes du flux de données entre UTA et mémoires**
 - *Data Address Generators: DAG*
 - Fonctionnement parallèle: recherche de 2 opérandes pendant 1 seul cycle
 - Supporte matériellement : *circular buffering*

- **Séquencement efficace des tâches**
 - Supporte des cycles singuliers de branchement (delayed branching)
 - “Zero overhead looping” en hard

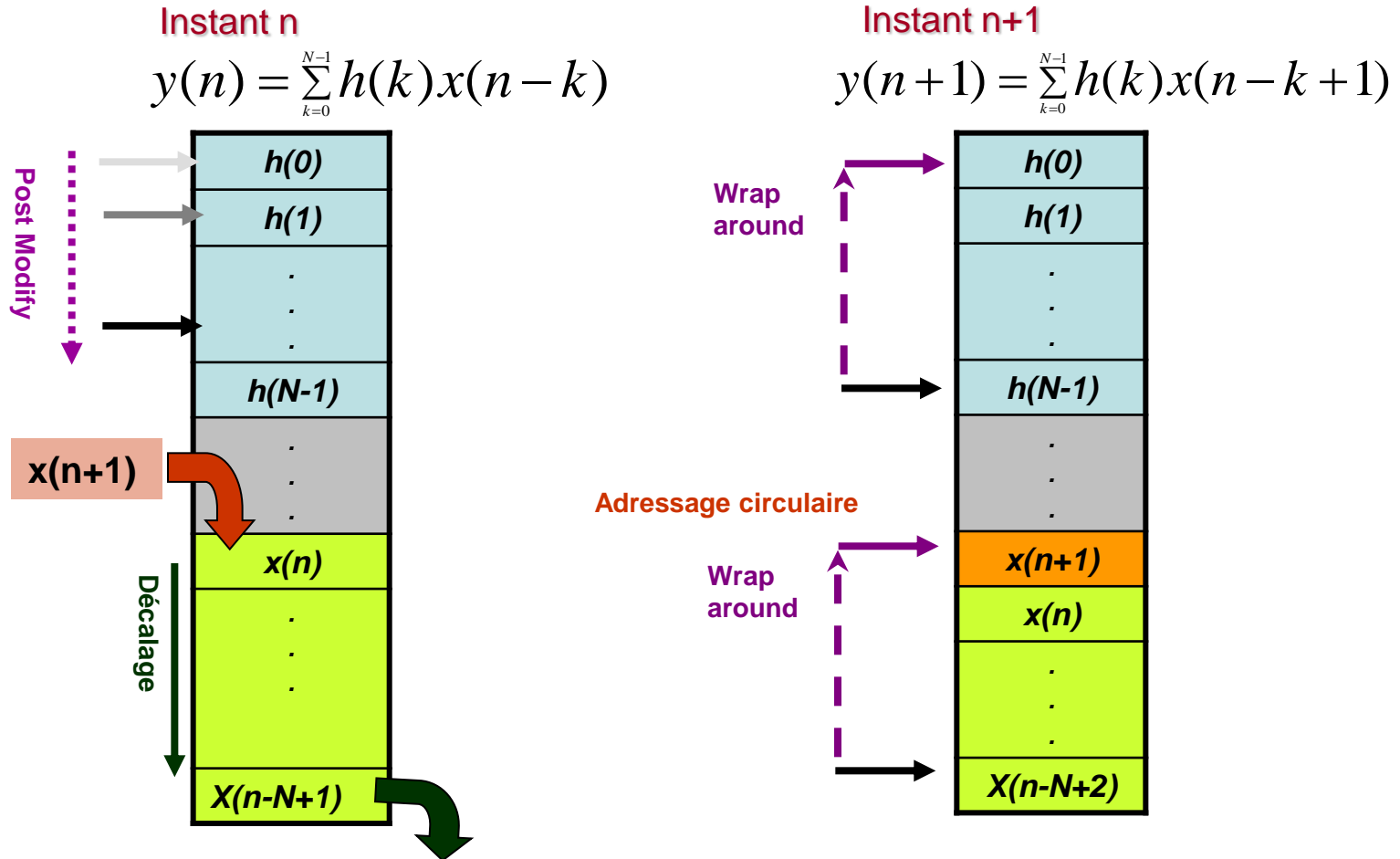
- **Facilité de programmation**
 - Jeu d'instructions en assembleur algébrique + fonctions C
 - Précision et plage dynamique étendue pour les unités de traitement

Architecture DSP de base





Exemple du filtre





Registres du DAG

Index	Modify	Length	Base
I0	M0	L0	B0
I1	M1	L1	B1
⋮	⋮	⋮	⋮

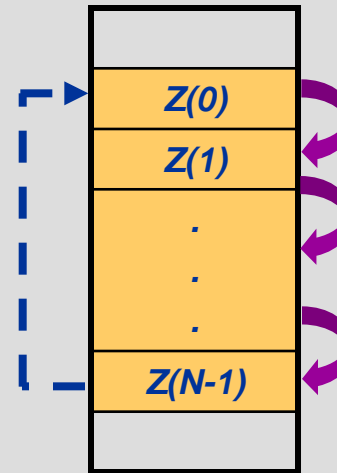
I : pointe sur la case mémoire en cours d'accès

M : contient le pas d'avancement de I après accès

L : spécifie la taille du buffer circulaire (si $L=0$ adressage non circulaire)

B : adresse de base du Buffer

Post Modify & Adressage circulaire

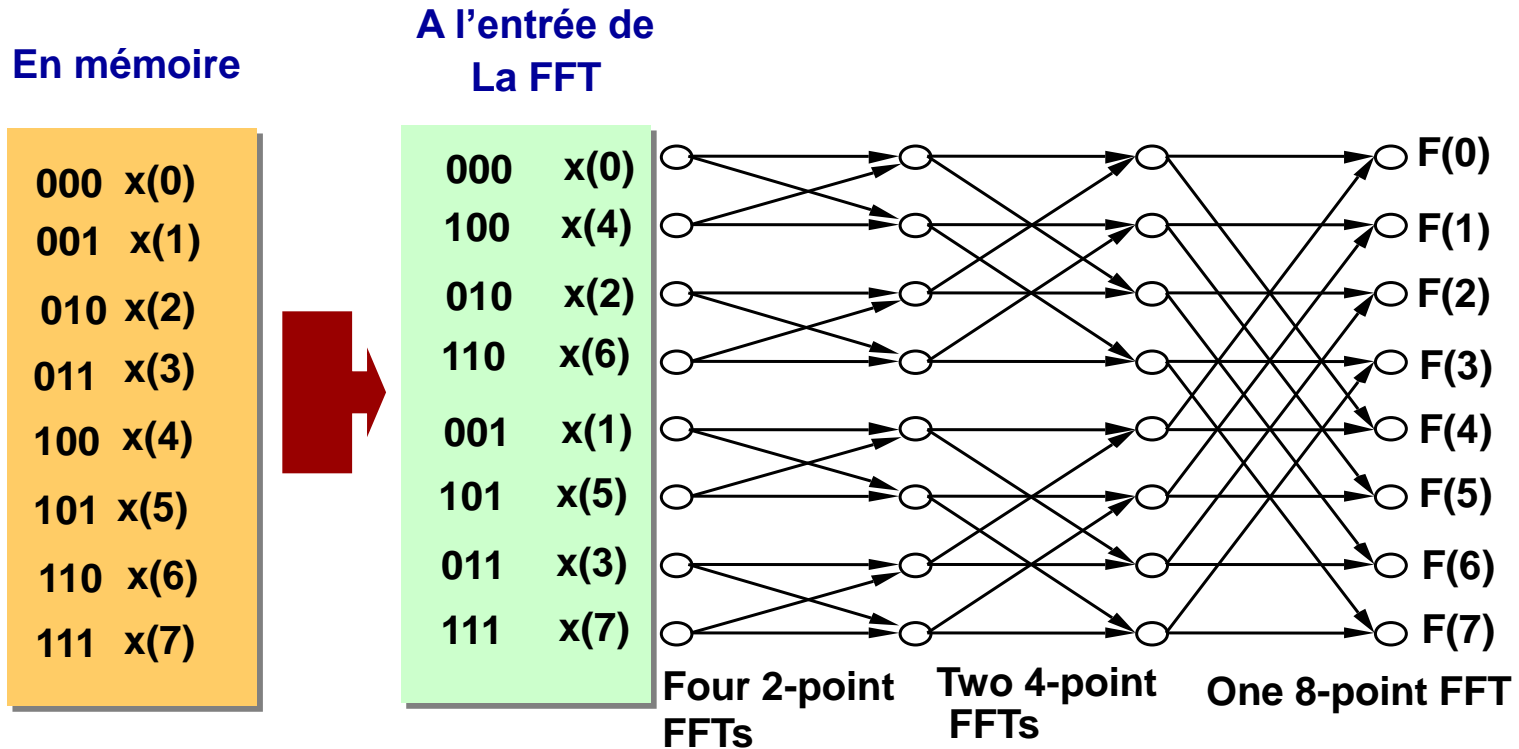


REG=[I0+M0];

REG ← z(k)

I0 ← I0 + M0 - B0 mod (L0)

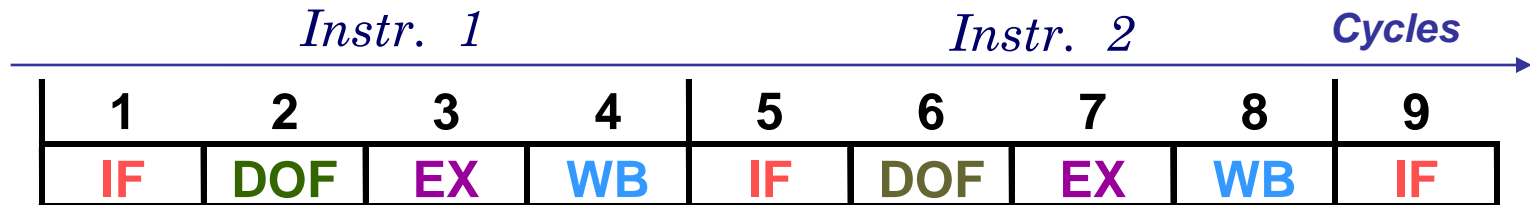
Bit Reversed Addressing



**Bit Reversed Addressing permet une lecture des échantillons
dans l'ordre de traitement par la FFT**



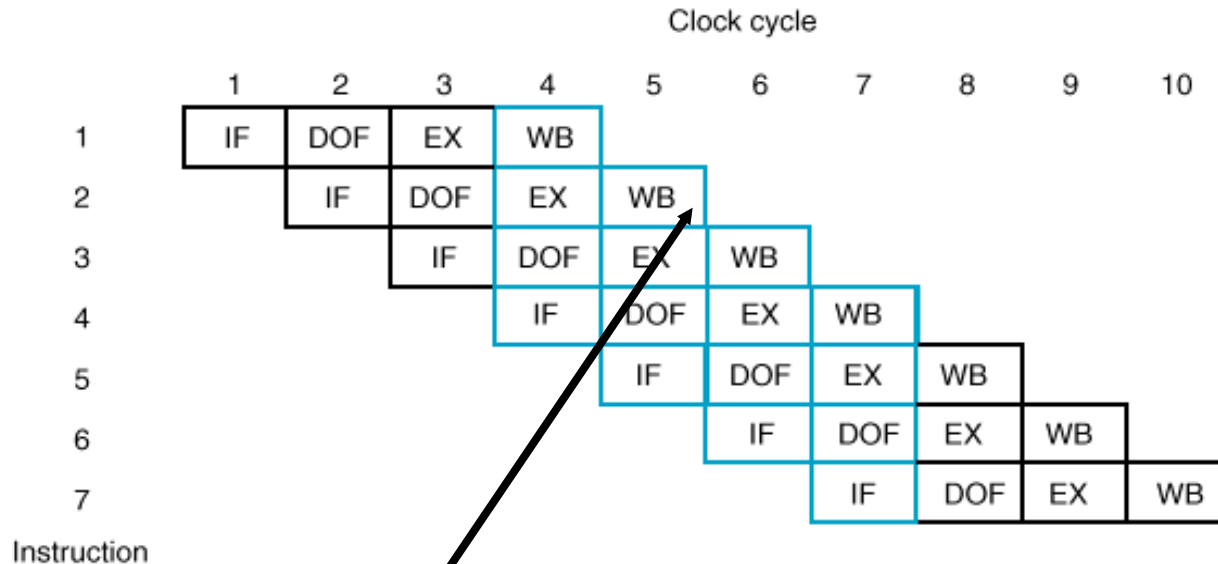
Exécution séquentielle



2 Instructions : 8 cycles

IF Instruction fetch
DOF Decode and operand fetch
EX Execution
WB Write back

Exécution en pipeline



2 Instructions : 5 cycles

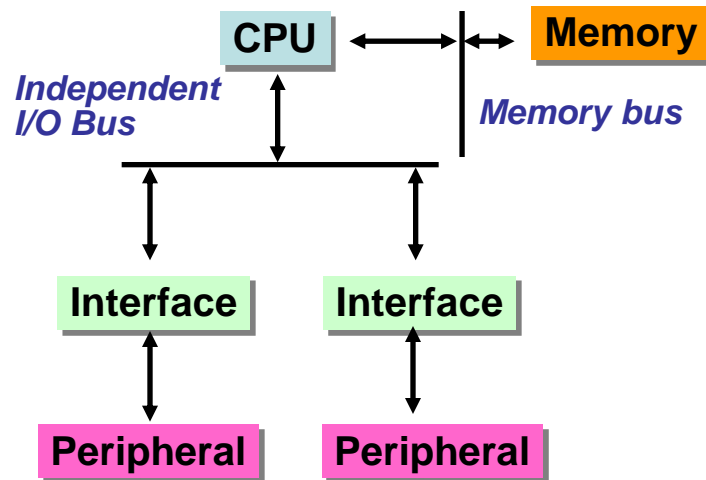
Exceptions : Jump, Call, ...



Pipeline Update

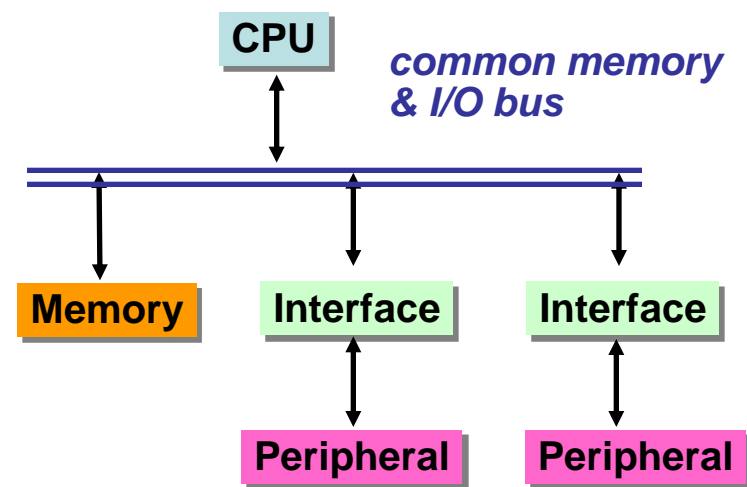


Architecture à Bus indépendants



Instructions dédiées I/O (in,out)

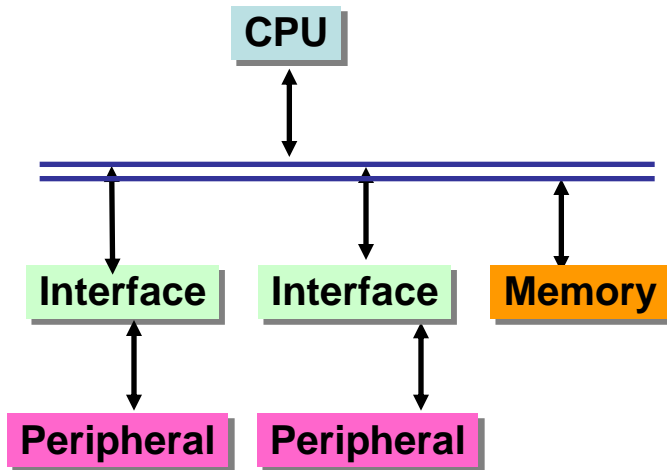
Architecture à Bus commun



Memory Mapped I/O



Architecture à Bus commun



- Configuration de l'interface et association d'un pointeur mémoire à un flux d'entrée
- Auto incrémentation du pointeur à chaque acquisition
- Buffer circulaire
- Génération d'une interruption au moment du *Wrap around*

Acquisition se déroule en **Background** ce qui libère le Processeur

Direct Memory Access: DMA



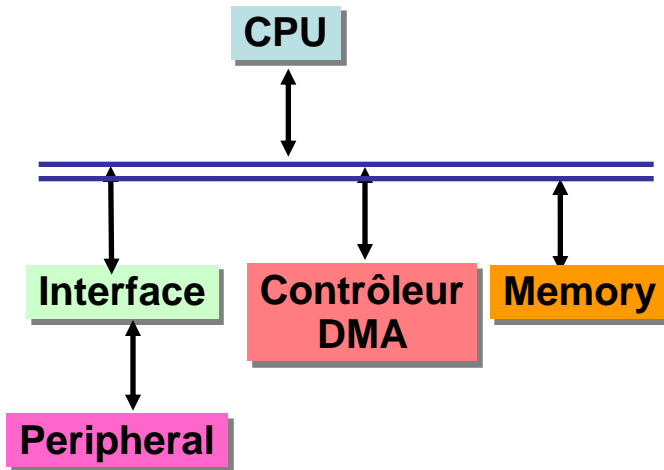
DMA setup :

- Adresse source
- Adresse destination
- Sens du transfert
- Taille des données à transférer

Launch DMA

A la fin du transfert une interruption est générée

Transfert se déroule en
Background ce qui libère le
Processeur





Classification des technologies DSP

d'Analog Devices Inc.





Analog Devices (*www.analog.com/dsp*)

ADSP-21xx 16 bit, fixed point

ADSP-21xxx 32 bit, floating and fixed point

Lucent Technologies (*www.lucent.com*)

DSP16xxx 16 bit fixed point

DSP32xx 32 bit floating point

Motorola (*www.mot.com*)

DSP561xx 16 bit fixed point

DSP560xx 24 bit, fixed point

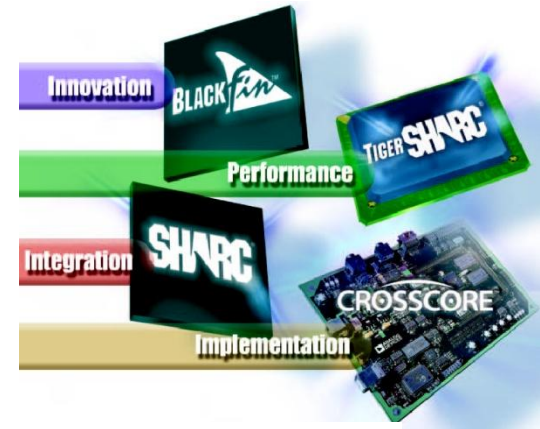
DSP96002 32 bit, floating point

Texas Instruments (*www.ti.com*)

TMS320Cxx 16 bit fixed point

TMS320Cxx 32 bit floating point

Technologie DSP d'ADI...

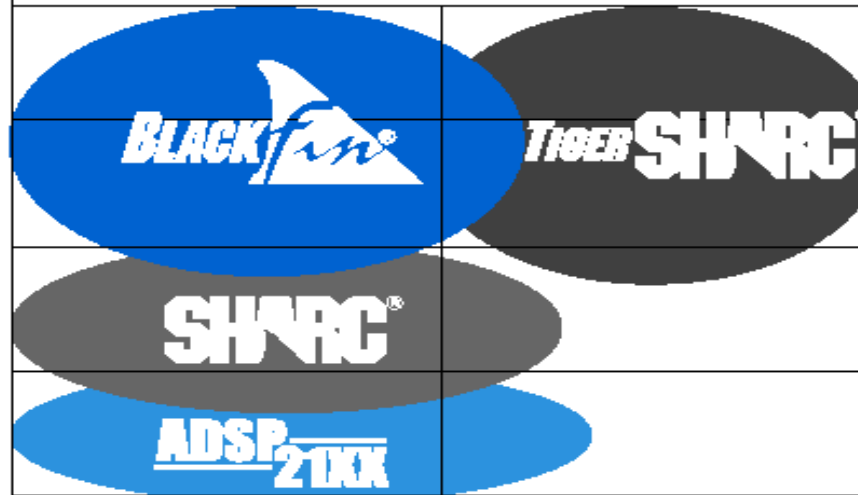


Media
Processing

Baseband
Processing

Audio
Processing

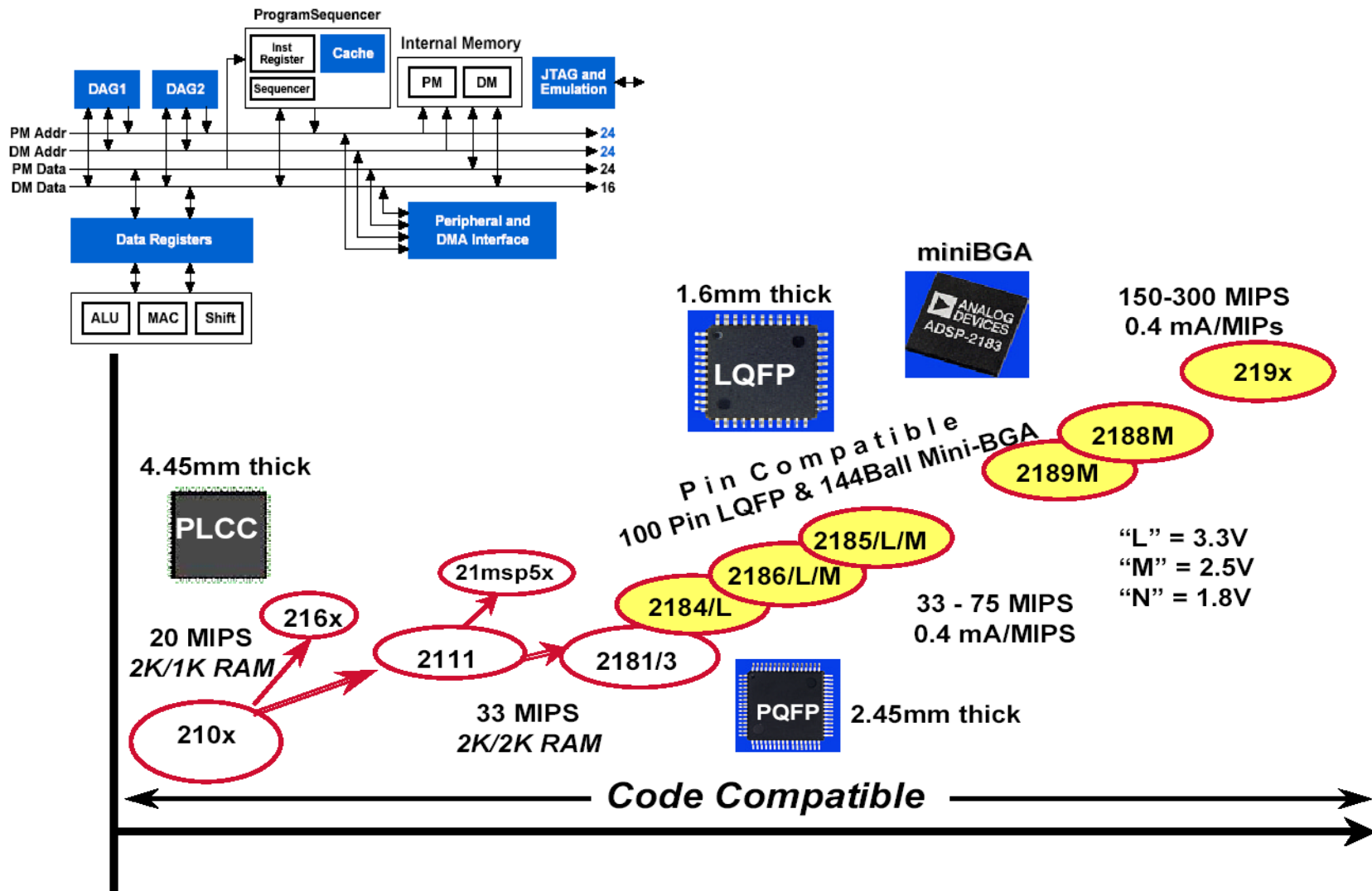
Speech
Processing



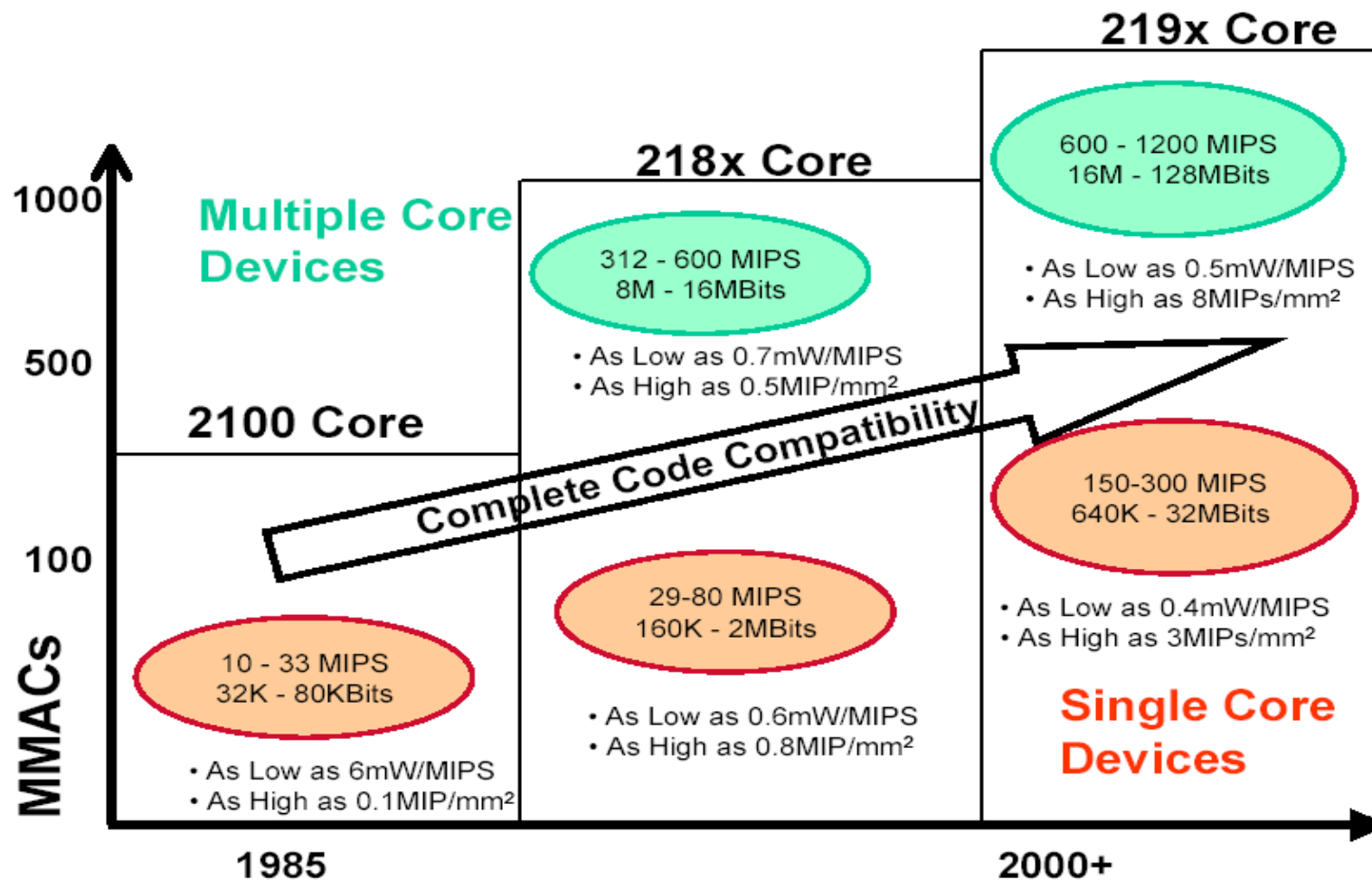
Consumer Devices

Infrastructure

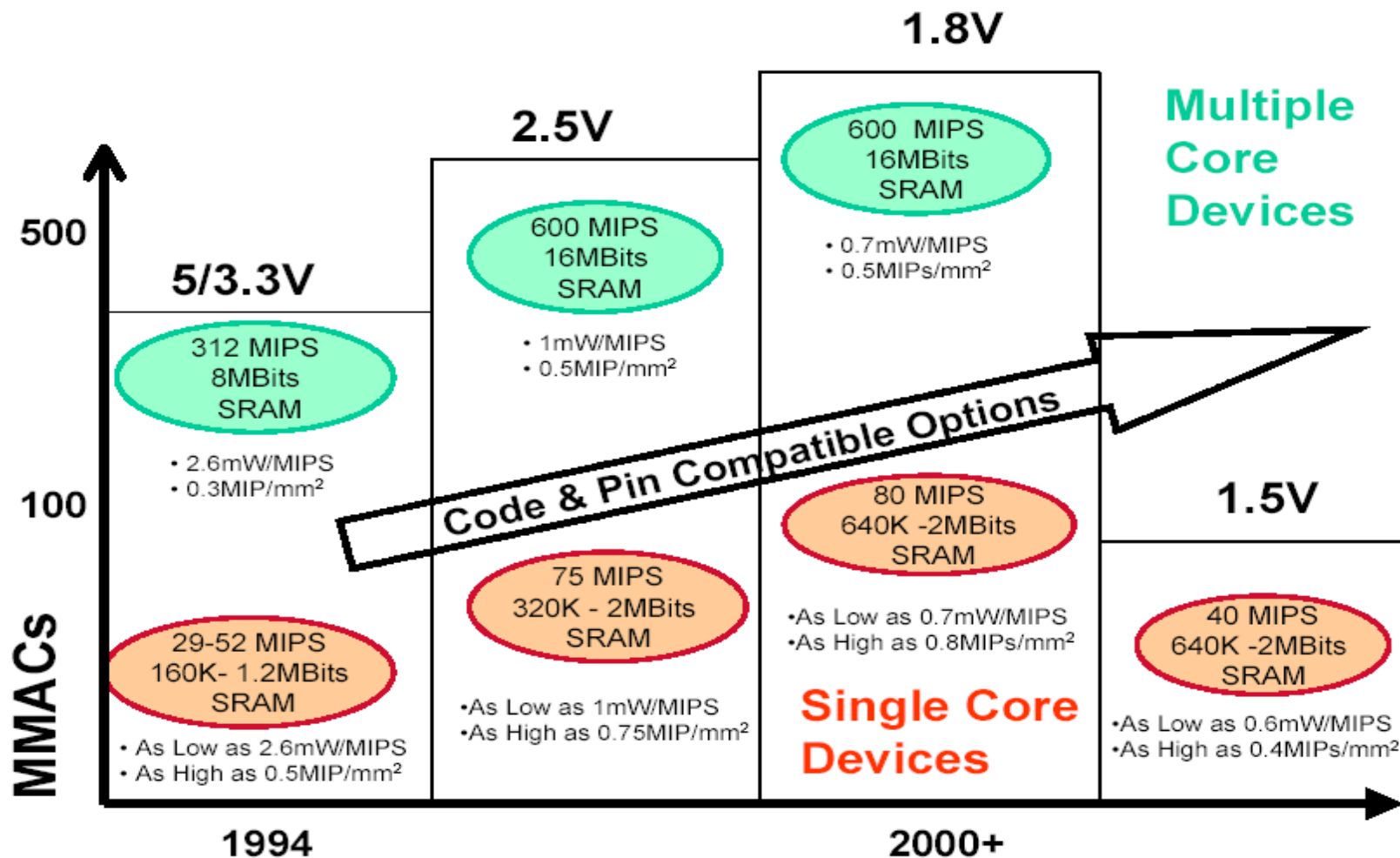
Classification des DSP 16 bits



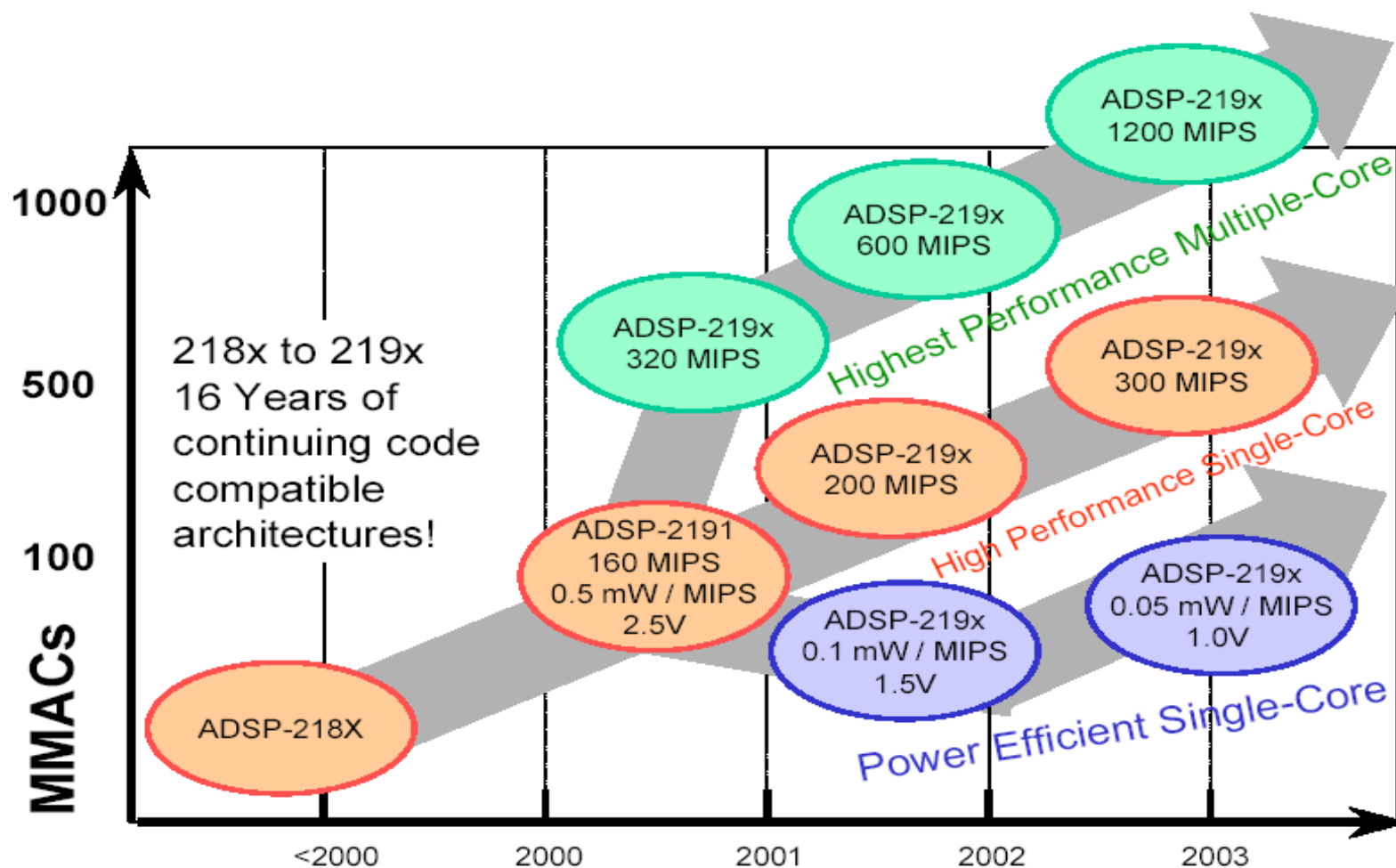
Caractéristiques - ADSP21xx



Caractéristiques - ADSP218x



Caractéristiques - ADSP219x



DSP 32 bits virgule flottante



MULTIPROCESSING

ADSP-21060

ADSP-21160M

MP
HP SHARC

- 10 GFLOPs
- 64 Mbits



ADSP-21062

- 120-198 MFLOPs
- 0.5 - 4 Mbits Memory

ADSP-21061

ADSP-21065

ADSP-21161N

Low Cost
HP SHARC

- 1200 MFLOPs
- <<\$10 SHARC

MASS MARKET

DSP Tiger SHARC



Core

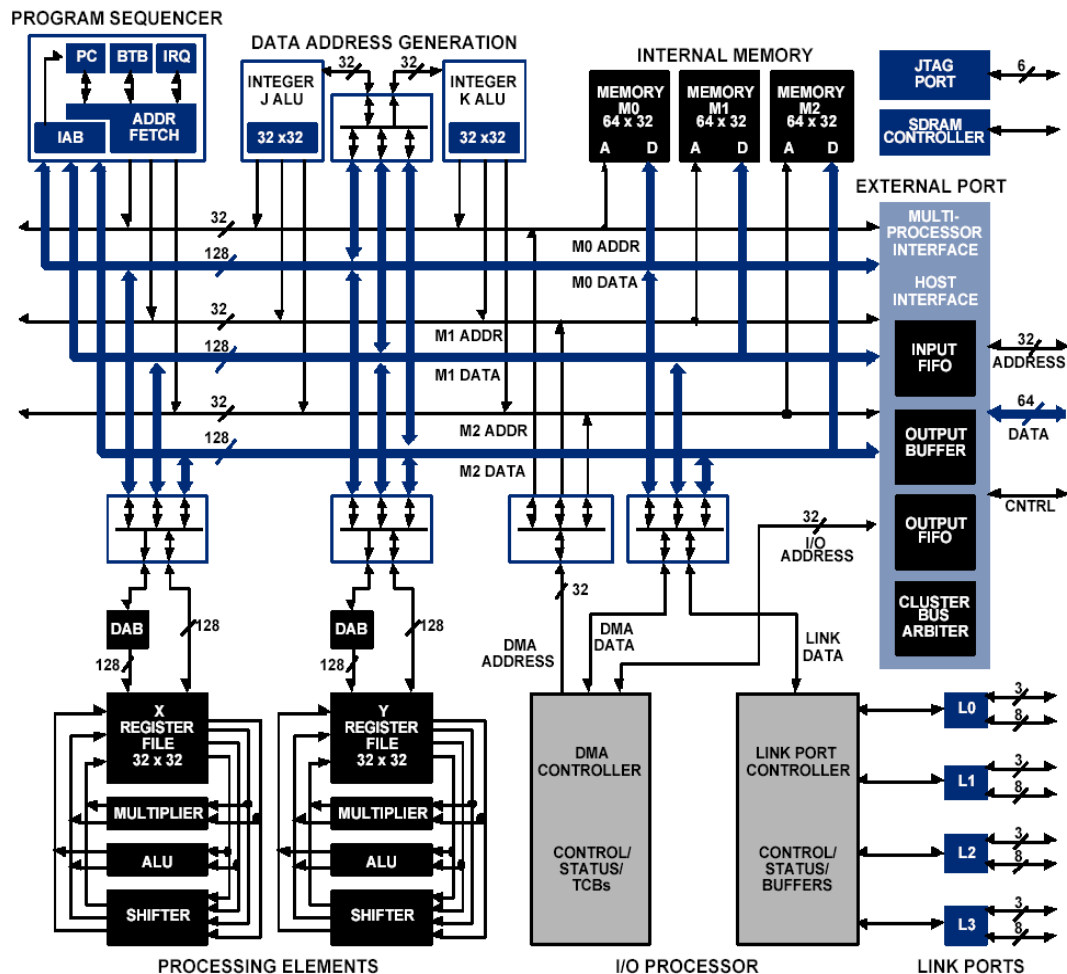
- 1200 MMACs/s @ 150 MHz -- 16-Bit Fixed Point
- 300 MMACs/s @ 150 MHz -- 32-Bit Floating Point
- 900 MFLOPS -- 32-Bit Floating Point

Memory

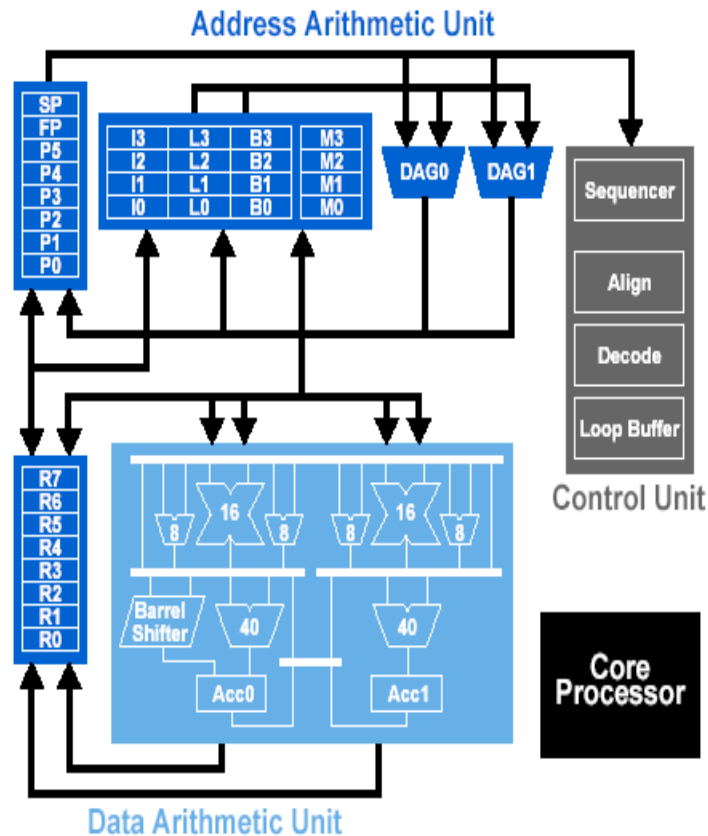
- 6 Mbits of on-chip SRAM organized in a unified memory map as opposed to the traditional Harvard architecture.

I/O, Peripherals, & Package

- 600 Mbytes/s transfer rate through external bus.
- 600 Mbytes/s aggregate transfer rate through 4 Link Ports
- Glueless multiprocessor cluster support for up to 8 ADSP-TS001s
- 4 General Purpose I/O Ports
- SDRAM Controller
- 360 Ball, SBGA Package 35x35mm

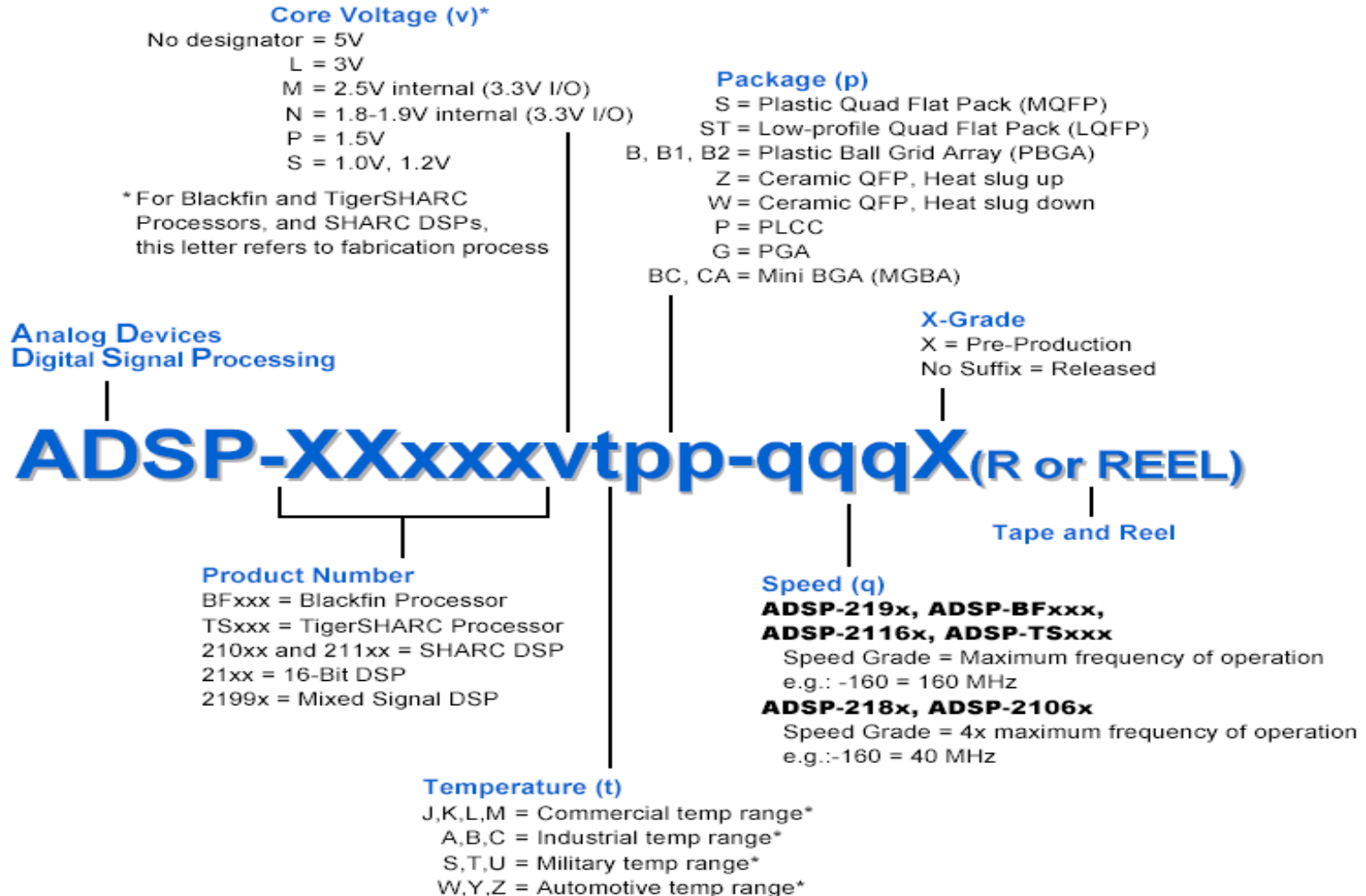


DSP de la famille Blackfin



Model	Max MMACS	L1 Memory Bytes	L2 Memory Bytes	Operating Voltage Core, I/O	Pin/Pkg	Price/1K*
ADSP-BF533SKBC-600	1200	148K	-	0.7-1.2V/3.3V	160-MBGA	\$23.50
ADSP-BF533SBBC-500	1000	148K	-	0.7-1.2V/3.3V	160-MBGA	\$20.00
ADSP-BF532SBBC-400	800	116K**	-	0.7-1.2V/3.3V	160-MBGA	\$11.50
ADSP-BF532SBST-300	600	116K**	-	0.7-1.2V/3.3V	176-LQFP	\$11.50
ADSP-BF531SBBC-400	800	84K**	-	0.7-1.2V/3.3V	160-MBGA	\$8.00
ADSP-BF531SBST-300	600	84K**	-	0.7-1.2V/3.3V	176-LQFP	\$7.00
ADSP-BF535PKB-350	700	52K	256K	1.0-1.6V/3.3V	260-PBGA	\$44.80
ADSP-BF535PBB-300	600	52K	256K	1.0-1.5V/3.3V	260-PBGA	\$35.20
ADSP-BF535PKB-300	600	52K	256K	1.0-1.5V/3.3V	260-PBGA	\$32.00
ADSP-BF535PBB-200	400	52K	256K	1.0-1.5V/3.3V	260-PBGA	\$30.00

Repérage des ADSP-XXxx





***Fin
de la 1^{ère} partie...***