



Institut  
Mines-Télécom

# Architecture matérielle des systèmes embarqués

...ou comment on fait les SE ?

Alexis Polti



# Licence de droits d'usage



Contexte académique } sans modification

***Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.***

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après, et à l'exclusion de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage dans un cadre académique, par un utilisateur donnant des cours dans un établissement d'enseignement secondaire ou supérieur et à l'exclusion expresse des formations commerciales et notamment de formation continue. Ce droit comprend :

- le droit de reproduire tout ou partie du document sur support informatique ou papier,
- le droit de diffuser tout ou partie du document à destination des élèves ou étudiants.

Aucune modification du document dans son contenu, sa forme ou sa présentation n'est autorisée.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel, non exclusif et non transmissible.

Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur :

[alexis.polti@telecom-paristech.fr](mailto:alexis.polti@telecom-paristech.fr)

# Plan



- • Architecture générale
- Composants discrets usuels
- Capteurs / effecteurs
- Flashs
- RAM
- CPU

## Brève histoire des SE

- 1967 : Apollo Guidance Computer, premier système embarqué. Environ un millier de circuits intégrés identiques (portes NAND).
- 1960-1970 : Missile Minuteman, guidé par des circuits intégrés.
- 1971 : Intel produit le 4004, premier microprocesseur, à la demande de Busicom. Premier circuit générique, personnalisable par logiciel.
- 1972 : lancement de l'Intel 8008, premier microprocesseur 8 bits (48 instructions, 800kHz)
- 1974 : lancement du 8080, premier microprocesseur largement diffusé. 8 bits, 64KB d'espace adressable, 2MHz – 3MHz.
- 1978 : création du Z80, processeur 8 bits
- 1979 : création du MC68000, processeur 16/32 bits

## Systeme embarqué : définition

- **Un système embarqué est un système :**
  - qui contient un processeur,
  - dont la fonction première n'est pas "d'être un ordinateur",
  - souvent créé pour accomplir une tâche spécifique,
  - en cachant la présence de processeur / logiciel à l'utilisateur.

# Système embarqué : définition

Système embarqué	Ordinateur
programmes souvent exécutés à partir d'une mémoire non volatile	programmes exécutés à partir de la RAM
"embarqué" = enfoui, auto-contenu	architecture ouverte, extensible
spécialisé	non spécialisé, standardisé
tolérant aux fautes, fiable	les fautes sont tolérées (!)

# Systèmes embarqués : exemples

- Exemples de systèmes embarqués :
  - Distributeur de billets
  - Carte à puce
  - Système de contrôle de vol d'avion
  - Console de jeux
  - Robot
  - Four micro-ondes
  - Disque dur

# Caractéristiques des SE

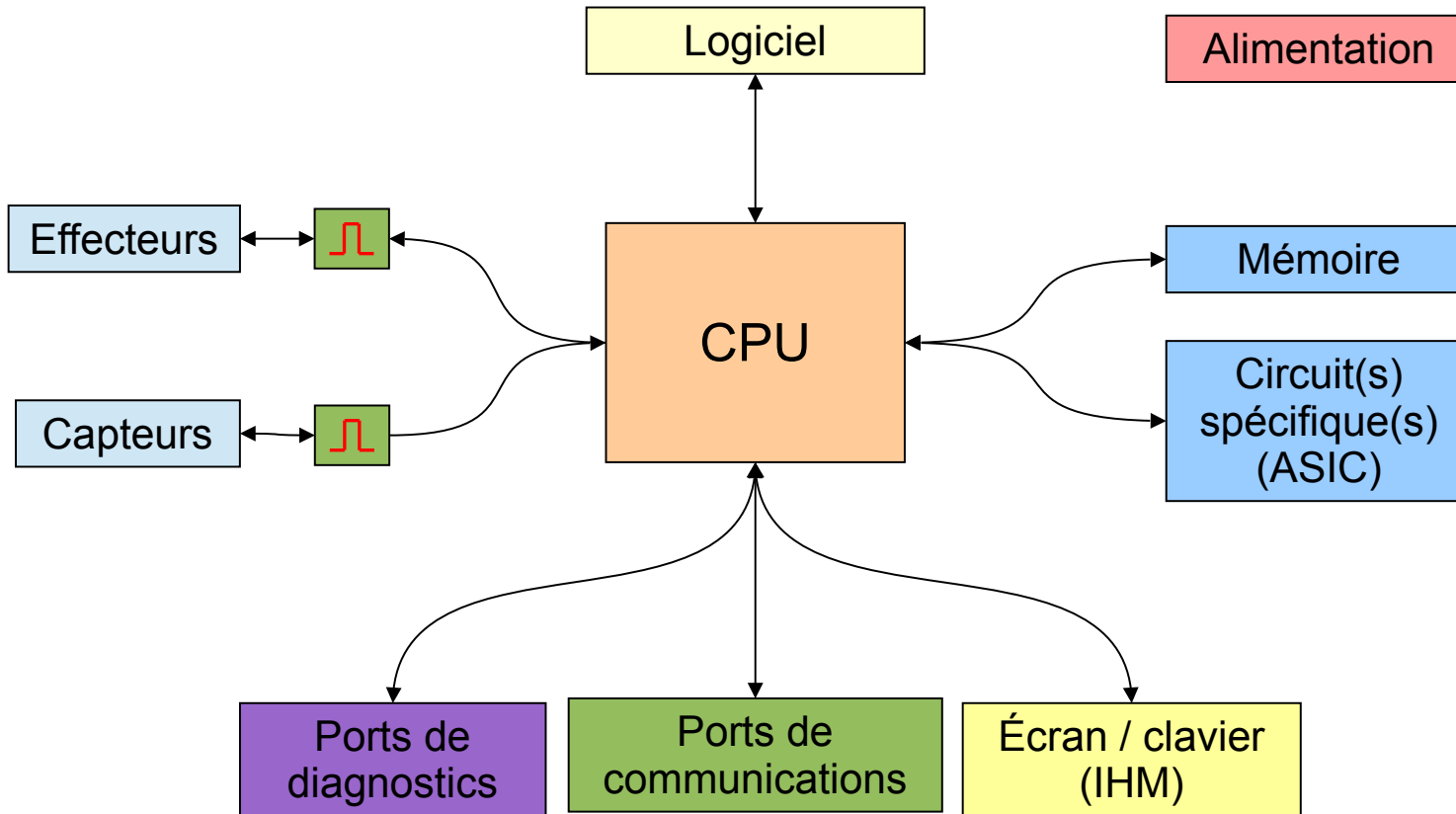
- Un système embarqué:
  - récupère éventuellement des stimuli extérieurs,
  - prend des décisions en fonction de ces stimuli et de son état interne,
  - altère son état interne et agit de manière perceptible par l'extérieur en fonction des décisions prises.
- Un système embarqué peut être :
  - Atomique : une centrale inertielle.
  - Hiérarchique : formé de plusieurs sous systèmes embarqués.



# Caractéristiques des SE

- Contraintes exercées sur un système embarqué :
  - encombrement
  - poids / portabilité
  - consommation d'énergie
  - coût
  - robustesse
  - sécurité
  - sûreté
  - maniabilité
  - performance
  - réactivité
  - déterminisme

# Architecture matérielle des SE



# Plan



- Architecture générale
- • Composants discrets usuels
- Capteurs / effecteurs
- Flashs
- RAM
- CPU

# Résistances

• Caractéristiques : 
$$\begin{cases} Z = R \\ U = R \cdot I \end{cases}$$

• Utilisations :

- Limitation de courant
- Tirage (pull-up / pull-down)

• Boîtiers CMS usuels ([JEDEC JEP95](#)) :

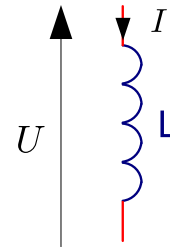
- 0201, 0402, 0603, 0805, 1206, 1210, 2512, ...
- Influent sur la chaleur pouvant être dissipée.



Code		L		W		H		P
Imperial	Metric	inch	mm	inch	mm	inch	mm	Watt
0201	0603	0.024	0.6	0.012	0.3	0.01	0.25	1/20 (0.05)
0402	1005	0.04	1.0	0.02	0.5	0.014	0.35	1/16 (0.062)
0603	1608	0.06	1.55	0.03	0.85	0.018	0.45	1/10 (0.10)
0805	2012	0.08	2.0	0.05	1.2	0.018	0.45	1/8 (0.125)
1206	3216	0.12	3.2	0.06	1.6	0.022	0.55	1/4 (0.25)
1210	3225	0.12	3.2	0.10	2.5	0.022	0.55	1/2 (0.50)
1218	3246	0.12	3.2	0.18	4.6	0.022	0.55	1
2010	5025	0.20	5.0	0.10	2.5	0.024	0.6	3/4 (0.75)
2512	6332	0.25	6.3	0.12	3.2	0.024	0.6	1

# Inductances

- Caractéristiques : 
$$\begin{cases} Z = j \cdot L \cdot \omega \\ U = L \cdot \frac{dI}{dt} \end{cases}$$



- Utilisations :

- Filtrage des hautes-fréquences.
- Stockage d'énergie.
- Adaptation d'impédance.

- Boîtiers CMS usuels :

- Rectangulaires : 0201, 0402, 0603, 0805, ...



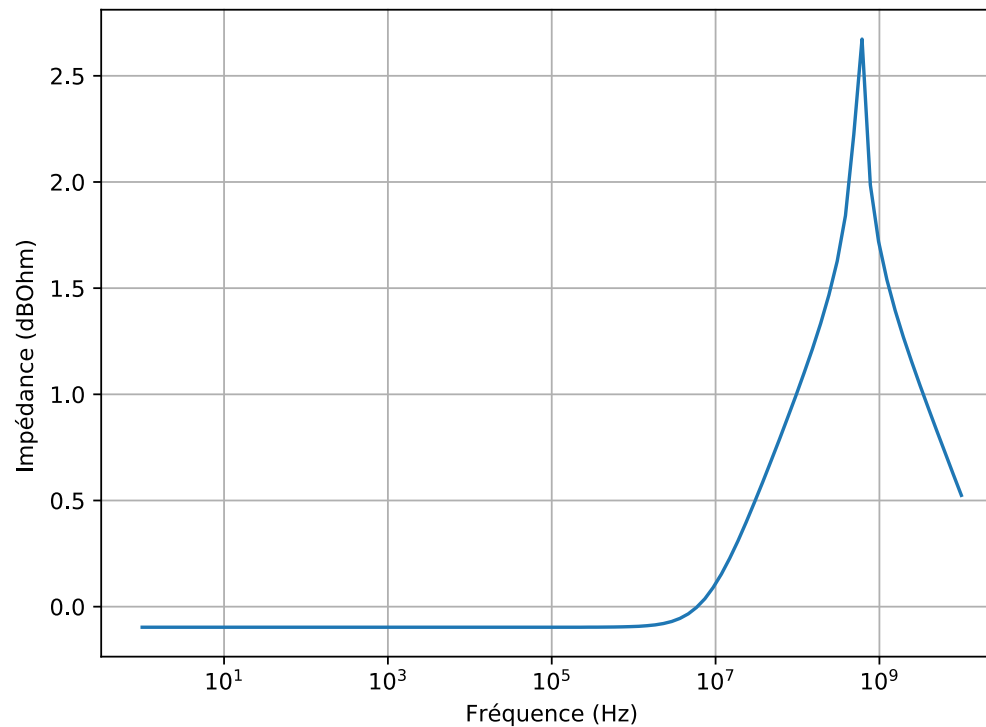
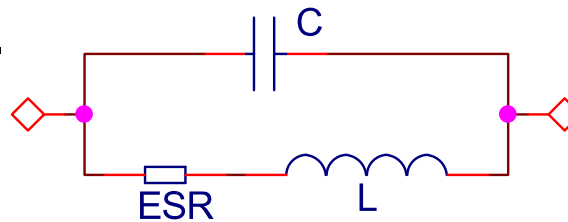
- Autres : noyau ouvert / fermé / blindées.



# Inductances

## • Problèmes :

- Résistance interne, capacité parasite.
- Courant de saturation.
- Modèle plus réaliste :



# Capacités

● Caractéristiques :

$$\begin{cases} Z = \frac{1}{j \cdot C \cdot \omega} \\ I = \frac{1}{C} \cdot \frac{dU}{dt} \end{cases}$$

● Utilisations :

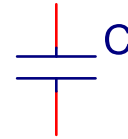
- Filtrage des hautes-fréquences.
- Stockage d'énergie.
- Adaptation d'impédance.

● Boîtiers CMS usuels :

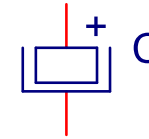
- Rectangulaires : 0201, 0402, 0603, 0805, ...



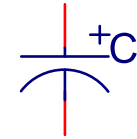
- Autres : A, B, C, ... // W, X, Y, Z // Radiaux ronds.



Non polarisée  
(céramique)



Polarisée  
électrolytique

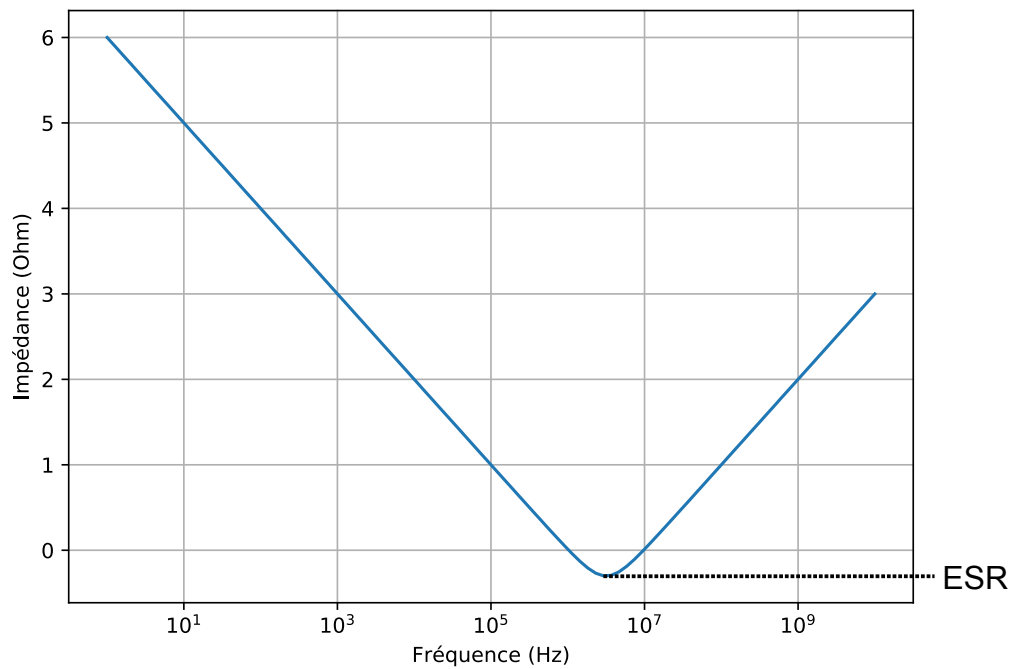
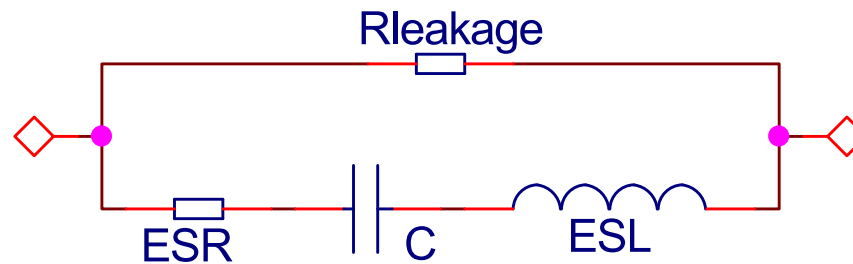


Polarisée  
tantale ou polymère

# Capacités

## • Problèmes :

- Tension de fonctionnement.
- Ripple admissible.
- ESR, ESL.
- Modèle plus réaliste :





- Différents types de diélectriques

- Électrolytiques :  $1\mu\text{F}$  à  $1\text{mF}$ ,  $6.3\text{V}$  –  $400\text{V}$ ,  $\text{ESR} \sim 10\ \Omega$
- Tantale :  $1\mu\text{F}$  à  $1000\mu\text{F}$ ,  $6.3\text{V}$  –  $50\text{V}$ ,  $\text{ESR} \sim 1 - 10\ \Omega$
- Polymère :  $1\mu\text{F}$  à  $470\mu\text{F}$ ,  $6.3\text{V}$  –  $50\text{V}$ ,  $\text{ESR} \sim 0.1 - 1\ \Omega$
- Céramique :  $0.1\text{pF}$  à  $10\mu\text{F}$ ,  $6.3\text{V}$  –  $200\text{V}$ ,  $\text{ESR} \sim 0.01\ \Omega$ 
  - Y5V, Z5U, ...
  - X5R, X7R, ...
  - C0G, N0P, ...

- À faire chez vous :

- Regarder la vidéo <https://www.baldengineer.com/slides/hddg11>
- La comprendre !
  - Si nécessaire, posez des questions sur la liste.
- À partir de maintenant, vous êtes sensés savoir quel type de capacité utiliser et quand.

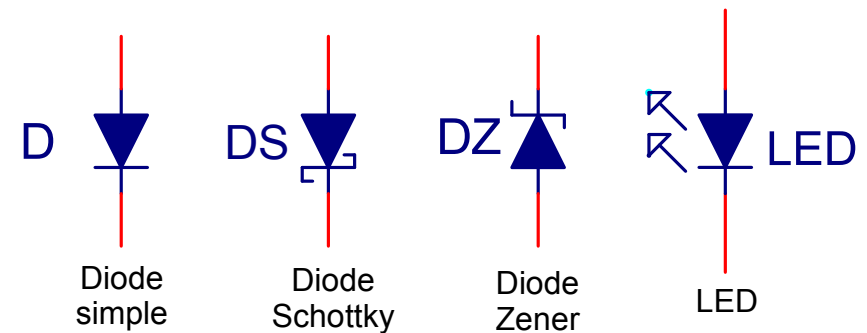
# Diodes

## Caractéristiques :

- Ne conduit que dans un sens, dès que la tension à ses bornes dépasse une tension seuil.
- Conduit assez de courant pour que la tension à ses bornes reste égale à la tension seuil.
  - Ou que quelque chose crame...

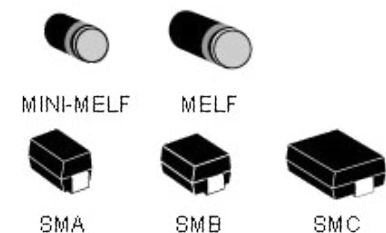
## Utilisations :

- Redressement du courant.
- Protection contre les surtensions.
- Référence de tension (Zener).



## Boîtiers CMS usuels :

- Rectangulaires : 0402, 0603, 0805, 1210, ...
  - Surtout pour les LED
- Autres : SOT523, SOT323, SOT223, SOT23 // SOD523, SOD323, SOD123 // SMA, SMB, SMC



# Diodes

## • Problèmes :

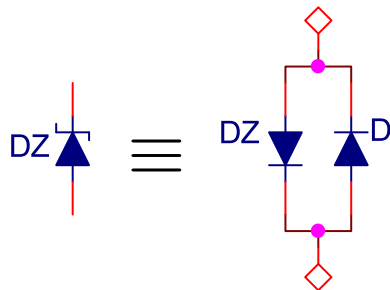
- Tension de seuil : 0.6 typique, 1.7 à 3.7V pour les LED.
- Résistance interne.
- Capacité de recouvrement (sauf Schottky).

## • Diode Schottky

- Très faible tension de seuil
- Pas de capacité de recouvrement → rapide.

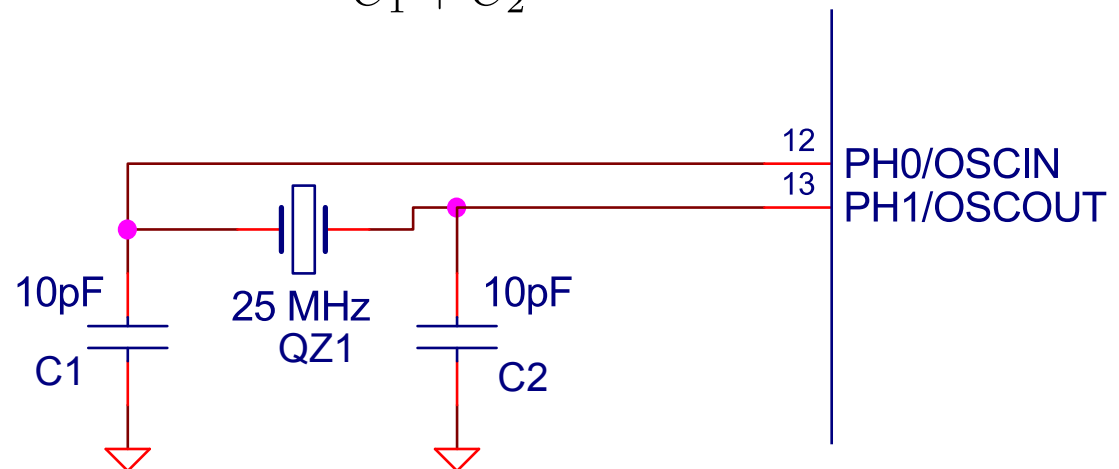
## • Diode Zener

- Deux diodes en inverse, dont une à tension de seuil élevée et au choix de l'utilisateur.
- Nécessite un courant minimal pour qu'elle conduise.
- Utilisée comme référence de tension.



## • Caractéristiques :

- Permet de générer une horloge « relativement » précise.
- Précision typique : 20ppm. Ne prend pas en compte l'influence de la température.
- Nécessite une capacité de charge totale  $C_L$  spécifiée par le constructeur pour fonctionner.
  - $C_s$  = capacité parasite des pistes et des broches des composants.
  - $C_1$  et  $C_2$  doivent respecter :  $C_L = \frac{C_1 \cdot C_2}{C_1 + C_2} + C_s$



# Transistors MOS

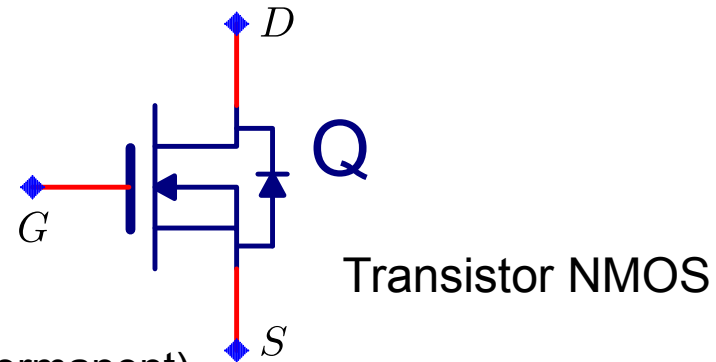
## • Caractéristiques :

### • Contrôlés en tension

- Le courant de grille est nul (en régime permanent).
- La grille se comporte comme une capacité (typiquement quelques nF).
  - MOS N : le transistor conduit si  $V_{GS} > V_{seuil}$ . Bloque sinon.
  - MOS P : le transistor conduit si  $V_{GS} < V_{seuil}$ . Bloque sinon.

### • Le transistor possède une diode parasite intrinsèque.

- Parfois utile, parfois pas...



## • Boîtiers CMS usuels :

- SOT523, SOT323, SOT223, SOT23 : petits signaux



- D2PACK, ... : grands courants.

# Circuits intégrés

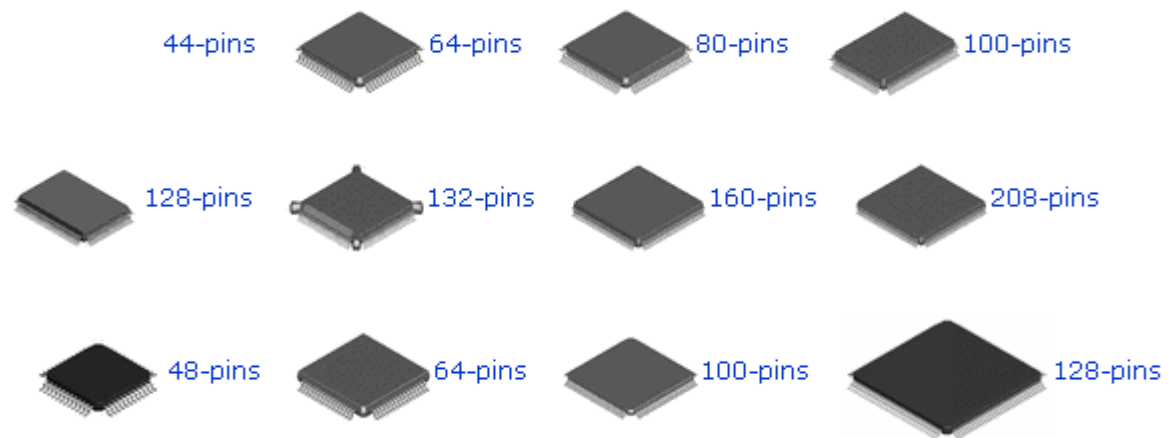
## • Les boîtiers :

### • CI carrés

- QFP (Quad Flat Pack)
- PQFP (Plastic Quad Flat Pack)
- TQFP (Thin Quad Flat Pack)

### • Espacement

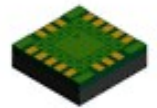
- 0,4mm
- 0,5mm
- 0,65mm
- 0,8mm
- 1mm



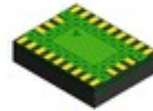
# Circuits intégrés

- Les boîtiers :

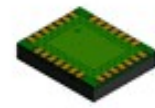
- QFN / LGA / CFP :



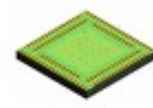
16-pins



24-pins



28-pins



128-pins

- PLCC :



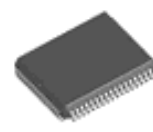
20-pins



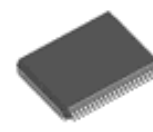
28-pins



44-pins



68-pins



84-pins

- Les boîtiers :

- Ball Grid Array

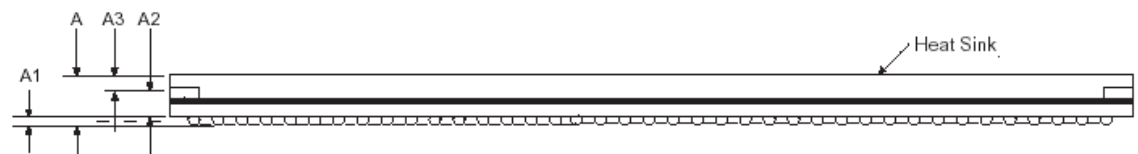
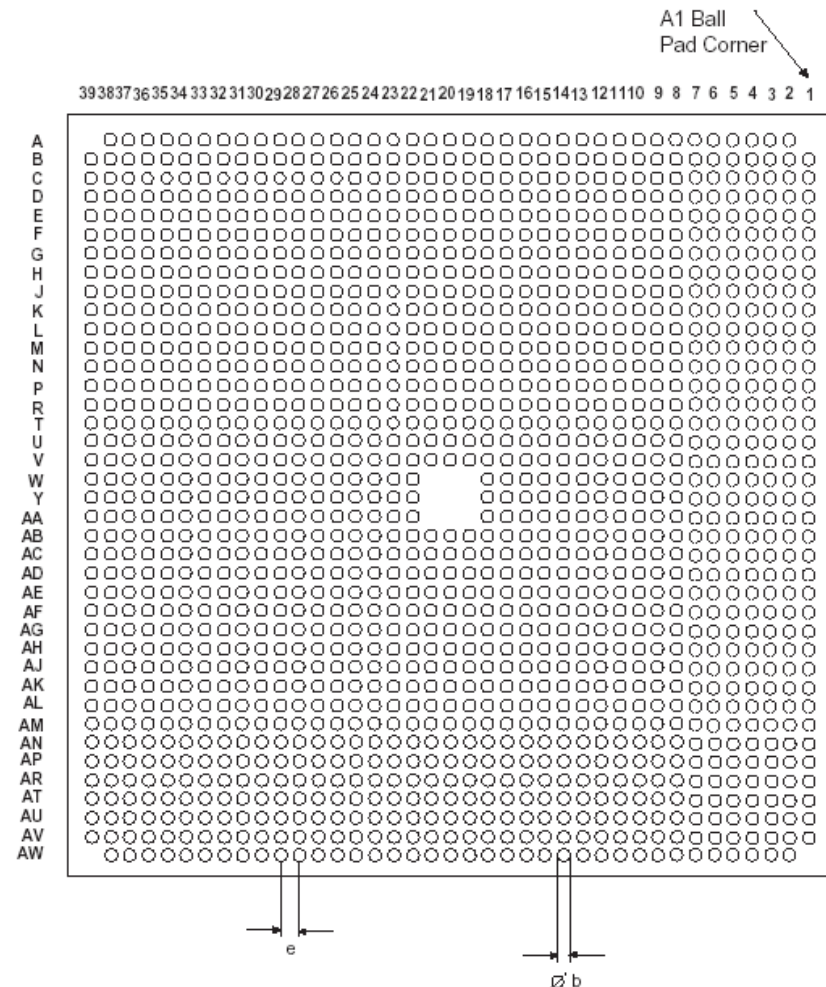
- BGA
    - LBGA
    - FBGA
    - uFBGA

- espacement

- 0.4mm
    - 0.6mm
    - 0.8mm
    - 1mm
    - 1.27mm

- broches

- 12
    - ...
    - >2100





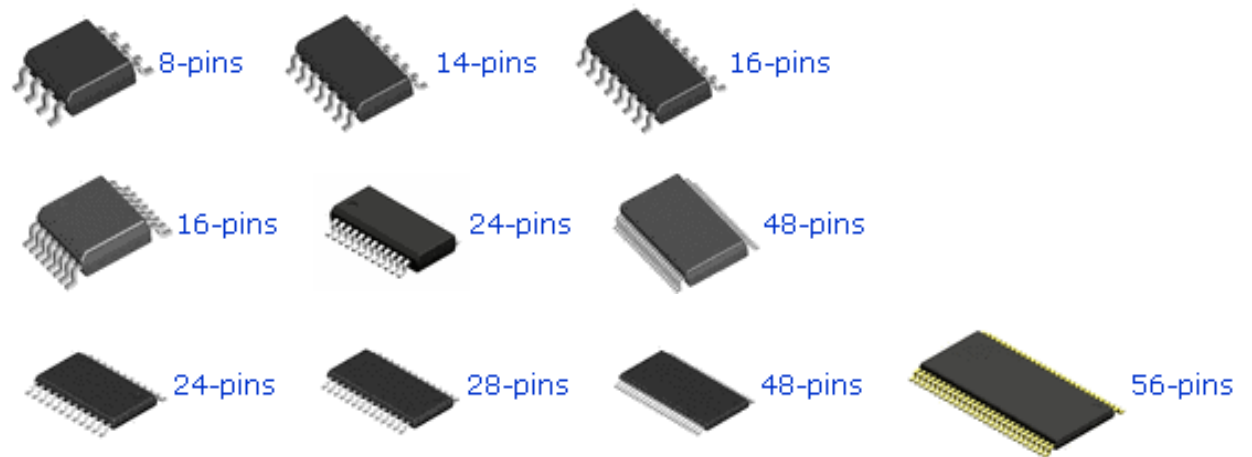
## • Les boîtiers :

### • CI en ligne (CMS)

- SO
- SSOP
- TSOP
- TSSOP

### • espacement

- 0,4mm
- 0,5mm
- 0,65mm



- Exercice 1 :
  - On veut piloter une LED à l'intensité lumineuse la plus forte possible par une GPIO d'un microcontrôleur.
    - Courant max dans la LED : 10mA
    - Tension de seuil de la LED : 2.3V
    - Tension de sortie de la GPIO : 0 / 3.3V

Comment brancher tout ça, quelles valeurs de composants ?

- Même question si :
  - le courant maximum admissible dans la GPIO est de 6mA (source) et -15mA (sink),
  - avec une tension d'alimentation du circuit de 3.3V.
- Même question si le courant maximum admissible dans la GPIO est de 4mA, et que la LED demande 500mA de courant pour fonctionner (LED de puissance) ?

## • Exercice 2 :

- On veut piloter la broche de  $\overline{\text{RESET}}$  d'un microcontrôleur à l'aide d'un bouton poussoir.
  - Cette broche est active à l'état bas.

Tracez le schéma du circuit.

## • Exercice 3 :

- Un microcontrôleur dispose d'une GPIO qui a une double fonction :
  - Au reset et pendant les 100 cycles après, c'est une entrée :
    - GND : le processeur boote normalement,
    - VCC : le processeur boote sur un bootloader de secours.
  - Après ça, c'est une GPIO normale.

On ne souhaite pas l'utiliser en mode GPIO. Comment la câbler de façon à booter normalement la plupart du temps et se réserver la possibilité d'utiliser le bootloader de secours en appuyant sur un bouton ?

# Plan



- Architecture générale
- Composants discrets usuels
- • Capteurs / effecteurs
- Flashs
- RAM
- CPU

## • Capteurs :

- Récupèrent des informations physiques de l'environnement.
- Virtuellement, tous les stimuli physiques peuvent être captés :
  - température,
  - lumière, couleur,
  - son,
  - vitesse (souvent angulaire),
  - accélération (souvent linéaire),
  - champ magnétique,
  - pression,
  - tension, courant, capacité...

- Interfaces pour ces capteurs
  - Alimentation
  - Isolation
  - Amplification
  - Multiplexage
  - Filtrage
  - Conversion analogique-numérique (ADC)
  - FPAA

### • **Effecteurs : influent sur l'environnement**

- LED, afficheurs, écrans, ...
- Moteurs
  - Usuels : à balais, brushless.
  - Pas à pas.
  - Servomoteurs.
- Haut-parleurs, buzzers, ...
- Émetteurs / absorbeurs de chaleur
- Stimulateurs tactiles / cardiaques
- Antennes
- Valves
- Relais

## • Interfaces des effecteurs

- Conversion numérique-analogique (DAC)
  - Convertisseur « du pauvre » : générateur PWM
- Amplification
- Interrupteurs de puissance (MOS, IGBT, ...)
- FPAA



# Plan



- Architecture générale
- Composants discrets usuels
- Capteurs / effecteurs
- • Flashes
- RAM
- CPU

## • Mémoire non volatiles

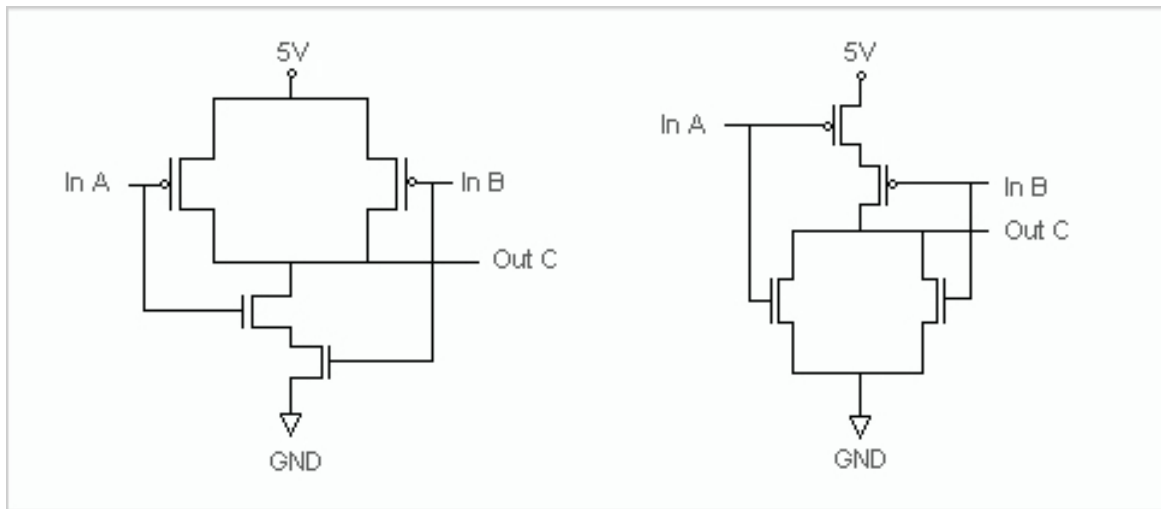
- Flash / MRAM / EEPROM
  - Stockage du programme (firmware)
  - Stockage de données de calibration ou de configuration
- Horloges temps réel (RTC) sauvegardées par batteries
- Types
  - mask-ROM / OTP-ROM
  - EPROM
  - EEPROM
  - Flash
  - RAM sauvegardées par batteries
  - MRAM

## • Flash

- Non volatiles
- Alimentation simple (5V, 3.3V, 1.8V)
- Cellule de base : transistor MOS avec une porte isolée (entre la grille de commande et le canal MOSFET)
- Les cellules peuvent être arrangées en
  - série, formant des portes NAND implicites
  - parallèle, formant des portes NOR implicites
- La taille des contacts (bit line, source line) est le facteur limitant pour la réduction de la taille des cellules.

# Flashes

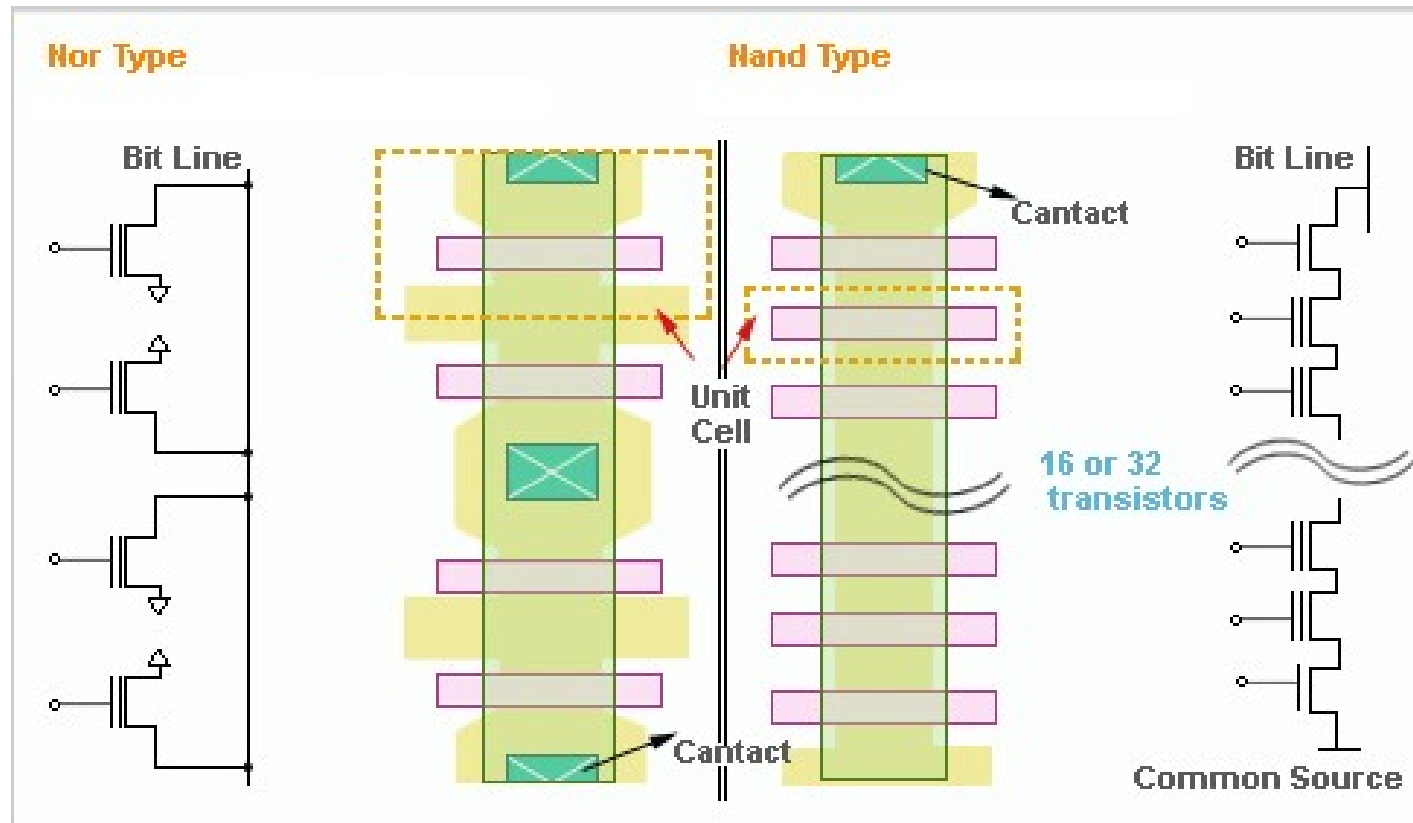
- Rappel :
  - Portes NAND / NOR à base de transistors MOS.



source : Samsung

# Flashes

- Flash NOR vs NAND

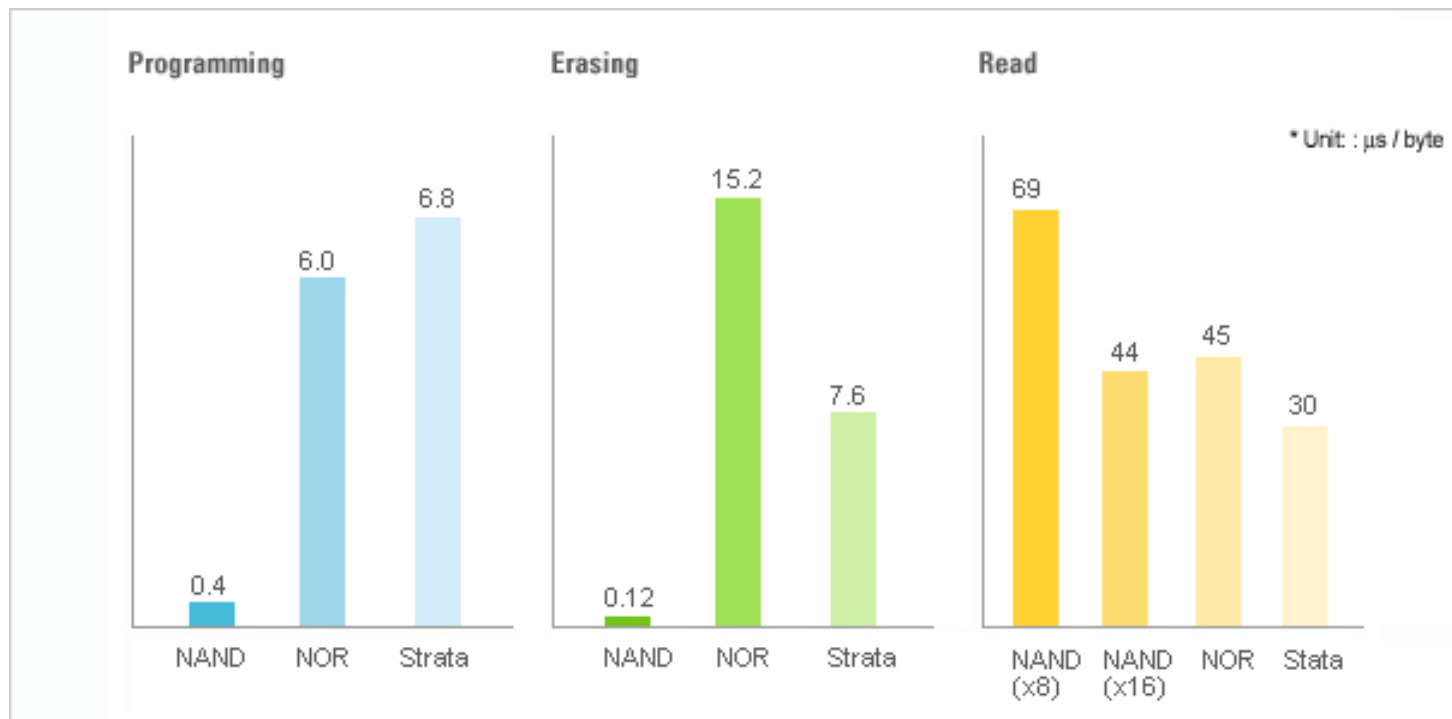


source : Samsung

# Flashes

	NOR	NAND
capacité	1MB - 16MB	8MB - 16GB
interface	type SRAM	IO : nécessite du matériel supplémentaire pour exécuter directement un code de boot
lecture	rapide, aléatoire	moins rapide, séquentielle par bloc
écriture	lente (6µs)	rapide (0.4µs)
effacement (bloc)	TRÈS lent : 0.5s	lent (2ms)
cycles	10 000 – 100 000	100 000 – 1 000 000
fiabilité	pas d'ECC	ECC (16B pour 512B de données)
utilisation	code	données
coût	>2x	x

## Flash NOR / NAND



- Problème de la production
  - Programmation avant montage sur outils dédiés.
  - Programmation in situ
    - JTAG / SWD (coûts fixes faibles)
    - Lit a clous (rapide, coûts fixes « élevés », possibilité de le faire lors du test électrique de la carte)
    - Doit être anticipée lors de la création des schémas.



# Plan



- Architecture générale
- Composants discrets usuels
- Capteurs / effecteurs
- Flashes
- • RAM
  - Dynamiques
  - Statiques
- CPU

## • Mémoire vive

### • Types

- statique : SRAM, SSRAM, ZBT-RAM, NoBL-RAM, ...
- dynamique : DRAM, SDRAM, DDRAM, ...

### • Intégrée au processeur ou non

- cache, générique ou mixte
- généralement de petite taille

### • Fonction

- fonctionnement du CPU
- acquisition / restitution de données à haut débit (tampons)

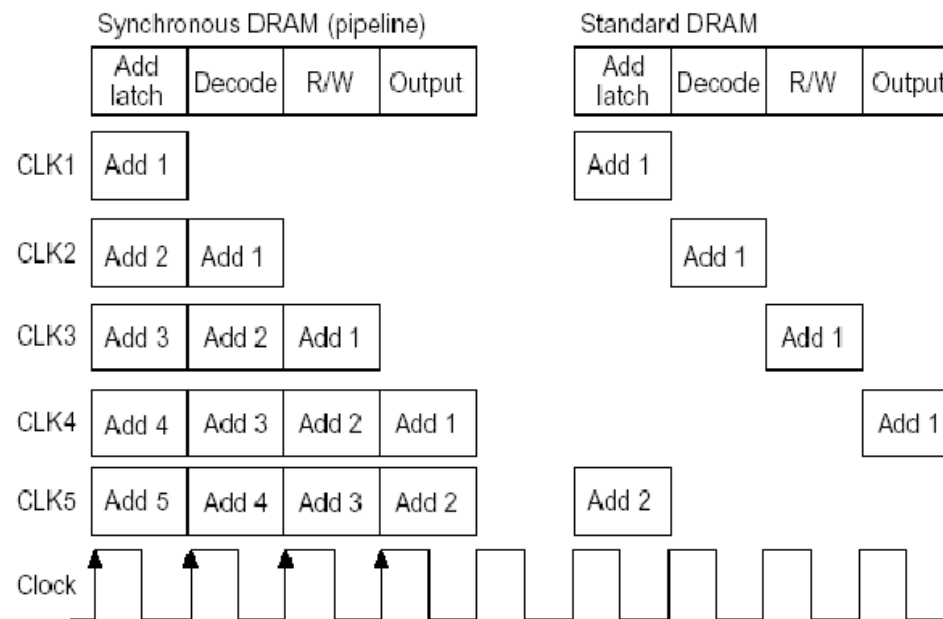
### • Le choix du type de RAM est souvent dicté par le type d'interface disponible sur le processeur

## • Organisation

- Une DRAM est organisée en
  - Matrice de rangées colonnes
  - Chaque point de la matrice correspond à  $w$  bits (largeur de bus de la mémoire)
- Capacité =  $N_{rangées} * N_{colonnes} * largeur$
- Profondeur =  $N_{rangées} * N_{colonnes}$
- Une capacité est souvent donnée sous la forme "*taille matrice \* largeur*"

## • SDRAM et DRAM

- Synchronisation des transactions sur l'horloge microprocesseur. Tous les signaux de contrôle sont échantillonnés sur front d'horloge.
- RAS et CAS n'ont besoin d'être positionnés que pendant un cycle d'horloge : pipeline des opérations possible.



## ● Organisation

- Une SDRAM est organisée en
  - Bancs
  - Chaque banc en matrice de rangées colonnes
  - Chaque point de la matrice correspond à  $w$  bits (largeur de bus de la mémoire)
- Capacité =  $N_{\text{bancs}} * N_{\text{rangées}} * N_{\text{colonnes}} * \text{largeur}$
- Profondeur =  $N_{\text{bancs}} * N_{\text{rangées}} * N_{\text{colonnes}}$
- Une capacité est souvent donnée sous la forme "*taille matrice \* largeur \* nombre de bancs*"
- Exemple : SDRAM 64Mb en x32 (512k x 32 x 4)
  - profondeur : 2M
  - largeur : 32 bits
  - 4 bancs
  - capacité : 64Mb

## • Organisation (suite)

- La sélection des bancs se fait par les bits d'adresse de poids fort.
- Chaque banc peut être adressé de façon indépendante.
- Une fois qu'une rangée d'un banc a été sélectionnée, une nouvelle colonne peut être sélectionnée à chaque cycle d'horloge.
- De plus, si différentes rangées sont requises et se situent dans des bancs différents, la plupart des temps de précharge peuvent être éliminés. Un banc peut être utilisé pendant que les autres sont en précharge.

## ● **Rafraîchissement**

- Principe : sélection d'une rangée, toutes les colonnes sont ensuite rafraîchies.
- Taux de rafraîchissement standard
  - nombre de rangées à rafraîchir pour rafraîchir toute la (S)DRAM.
  - généralement 1k, 2k, 4k ou 8k.
- Le circuit de sélection de colonne consomme plus de puissance que celui de sélection de rangée,
- Un rafraîchissement sélectionnant plus de colonnes par rangée consomme donc plus : à taille égale, 1k consomme plus (en pointe) que 8k...
- Compromis entre vitesse de rafraîchissement et consommation pic

- **Fréquence des cycles de rafraîchissement**
  - Distribués : cycles régulièrement espacés,
  - Burst : rafraîchissement de toute la mémoire en bloc.
- Cycles standards : tous les environ  $15.6\mu\text{s}$  au minimum



## • **Rafraîchissement des SDRAM**

### • Auto

- CAS before RAS
- Le contrôleur initie le cycle
- L'adresse de la rangée a rafraîchir est interne à la RAM, et incrémentée à chaque cycle
- Pas de lignes d'adresse a piloter : économise de l'énergie

### • Self refresh

- Rafraîchissement même quand le système est en veille (pas d'horloge transmise) : une horloge interne est produite.
- Besoin de commandes spéciales pour sortir de ce mode.

## • Câblage

- Les contrôleurs (S)DRAM ne supportent pas tous toutes les configurations possibles de (S)DRAM !!!
- Vérifier que le contrôleur (S)DRAM supporte bien la mémoire choisie
  - largeur du bus
  - multiplexage adresse colonne
- À partir de 80MHz environ, les contraintes d'intégrité du signal deviennent critiques pour les SDRAM
- cf. le cours / les TP d'intégrité du signal

# Plan



- Architecture générale
- Composants discrets usuels
- Capteurs / effecteurs
- Flashs
- RAM
  - Dynamiques
  - Statiques
- CPU



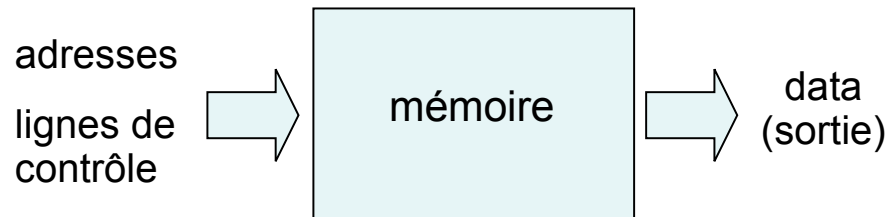
## • SRAM

- Pas besoin de rafraîchissement
- Cellule de base
  - SRAM : 6 transistors
  - DRAM : 1 transistor
- Une SRAM est plus rapide qu'une DRAM
- Consommation pic plus faible, consommation moyenne plus élevée

## • Types

- Asynchrones
- Synchrones standard
  - flow through
  - pipelined
- Synchrones améliorées
  - MRAM / FRAM
  - ZBT-RAM (NoBL, NT-RAM)
  - QDR-SRAM
  - DDR-SRAM
  - SigmaRAM, ...
- Simple port / multi ports

## • Architecture



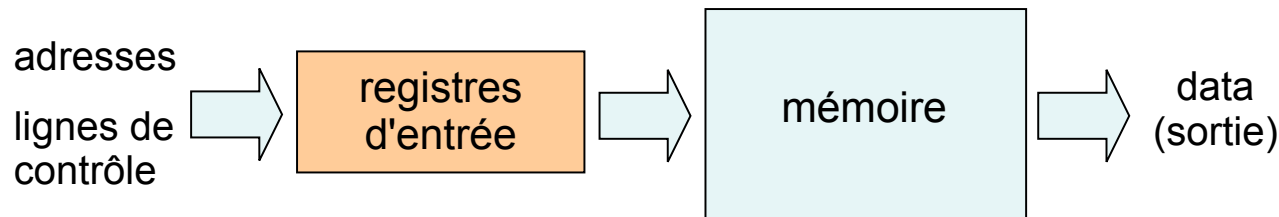
## • Asynchrones :

- Nécessité de maintenir les adresses / signaux de contrôle pendant toute la durée du cycle
- Nécessité de définir des temps d'accès minimum aux ressources
- Synchrone → asynchrone : perte de temps

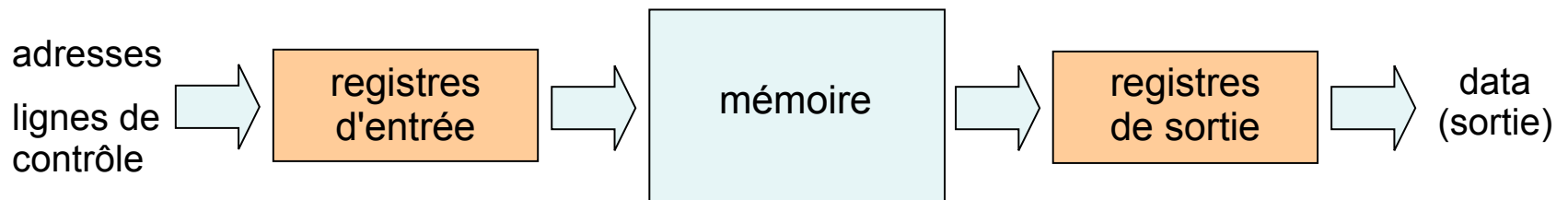
# SRAM synchrones

## • 2 architectures

### • Flow through



### • Pipelined



### • Séparation du chemin mémoire – pin – registre d'entrée du processeur en deux

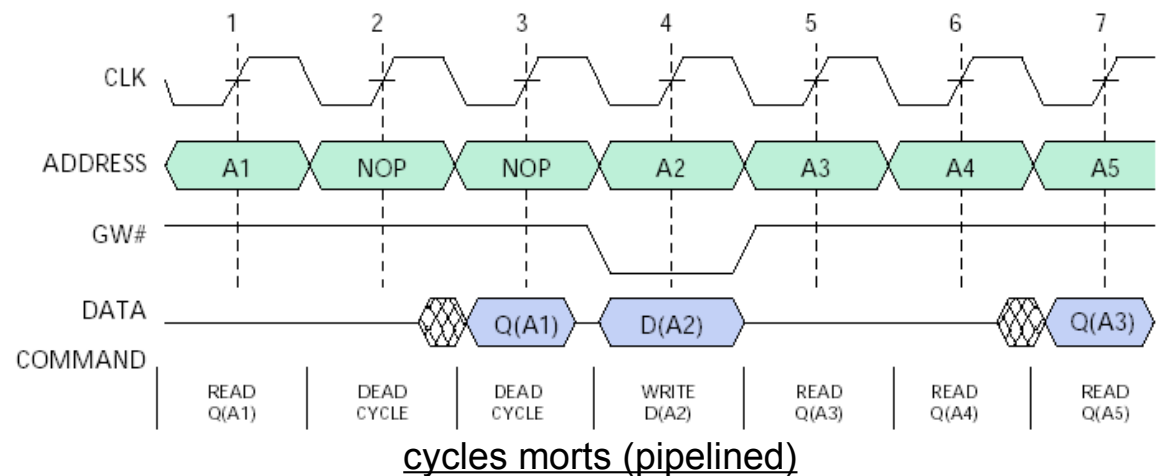
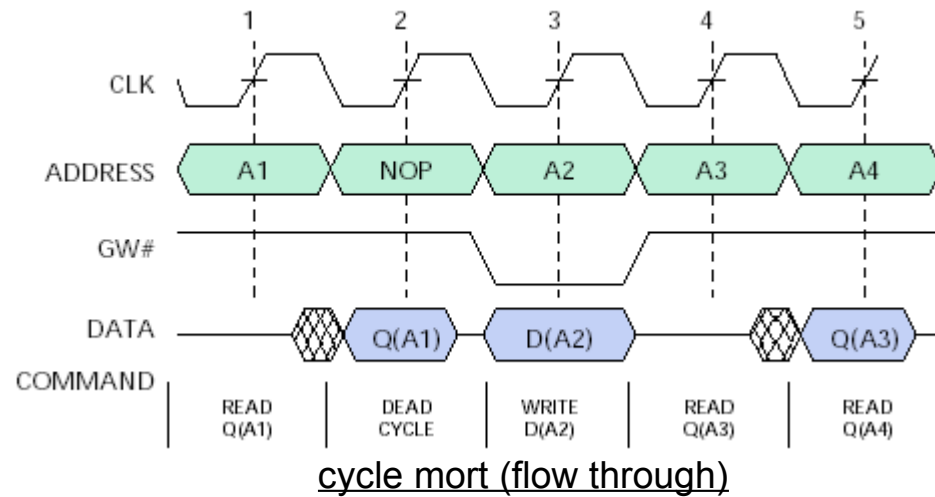
- augmentation de la fréquence de travail (débit)
- introduction d'une latence d'un ou deux cycles en lecture

## • Retournement de bus

- La longueur de pipeline n'est pas la même en lecture et écriture
  - lecture : donnée 1 ou 2 cycles après l'adresse
  - écriture : donnée en même temps que l'adresse.
- Donc insertion de cycle(s) mort(s) quand on passe d'un cycle de lecture à un cycle d'écriture.
- → Adaptées à des bus dont le sens d'utilisation est globalement constant.



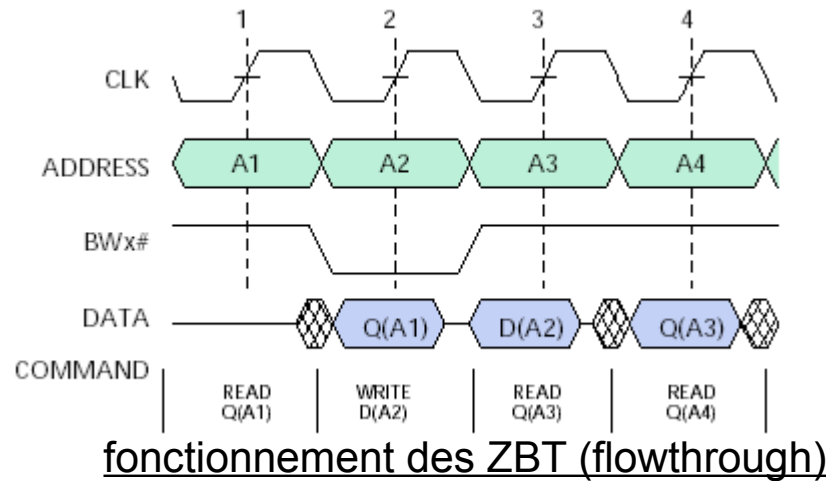
# SRAM synchrones



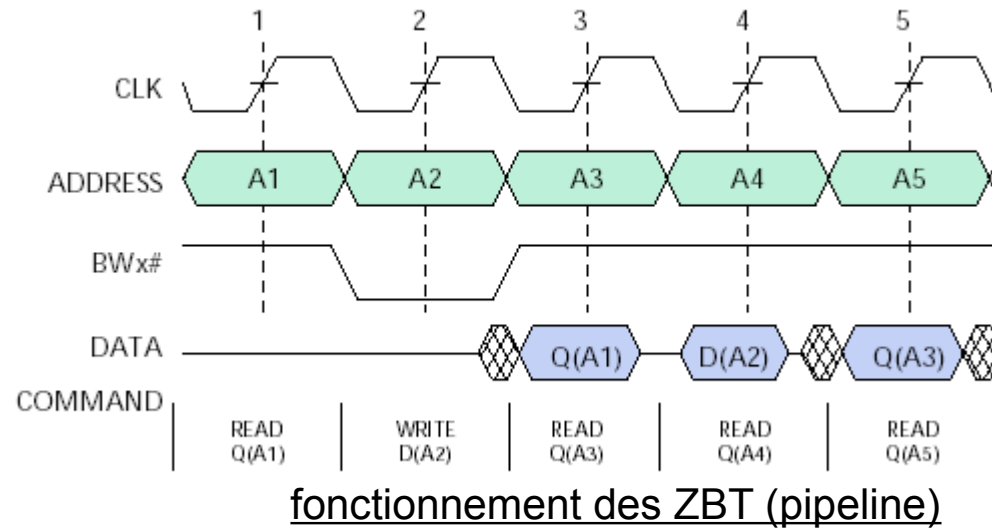
## • ZBT (Zero Bus Turnaround)

- Créées par Micron, pour éviter l'insertion de cycles morts lors des retournement de bus.
- Reprises par d'autres fabricants (NoBL, NT-RAM).
- Principe : équilibrage des pipelines de lecture et écriture.
- Adaptées aux changement fréquents de direction de bus.
- Nécessitent un contrôleur spécial.
- Le signal R/W ne peut plus servir d'indicateur de direction de bus : possibilité de contention de bus.

# SRAM synchrones ZBT



source : Micron



## • Limitation des ZBT

- Les temps de commutations (turn on/off) des drivers sont fixes.
- Pour éviter la contention de bus, les fréquences d'utilisation doivent rester dans la zone 50 – 166MHz.
- Au-delà, des cycles d'attente doivent être insérés...

- Les systèmes embarqués pour les réseaux demandent des RAM toujours plus puissantes
- Facteurs clefs :
  - Ratio lecture / écriture
    - application de look-up : lectures fréquentes, écritures occasionnelles
    - applications buffer : autant de lectures que d'écritures
    - applications QoS : ratio dépendant du trafic
  - Possibilité ou non d'augmentation de la bande passante par augmentation de la largeur de bus.
  - Latence acceptable.
- En fonction de ces différents facteurs, trois nouvelles sortes de SRAM synchrones ont été créées :
  - QDR
  - DDR IO séparées
  - DDR IO communes

## • QDR

- Bus de données lecture et d'écriture séparés.
- Chaque bus fonctionne en DDR : transferts sur chaque front d'horloge.
- Les deux bus fonctionnent de façon concurrente (cohérence assurée).
- Optimisée pour :
  - un ratio RW proche de 1 (balancé à court terme)
  - packet buffering (petits paquets)
- Facteur limitant : une adresse à chaque front d'horloge
  - le bus d'adresse a souvent un fan out élevé : bus lent,
  - la matrice interne doit être rapide (deux accès par cycle d'horloge).

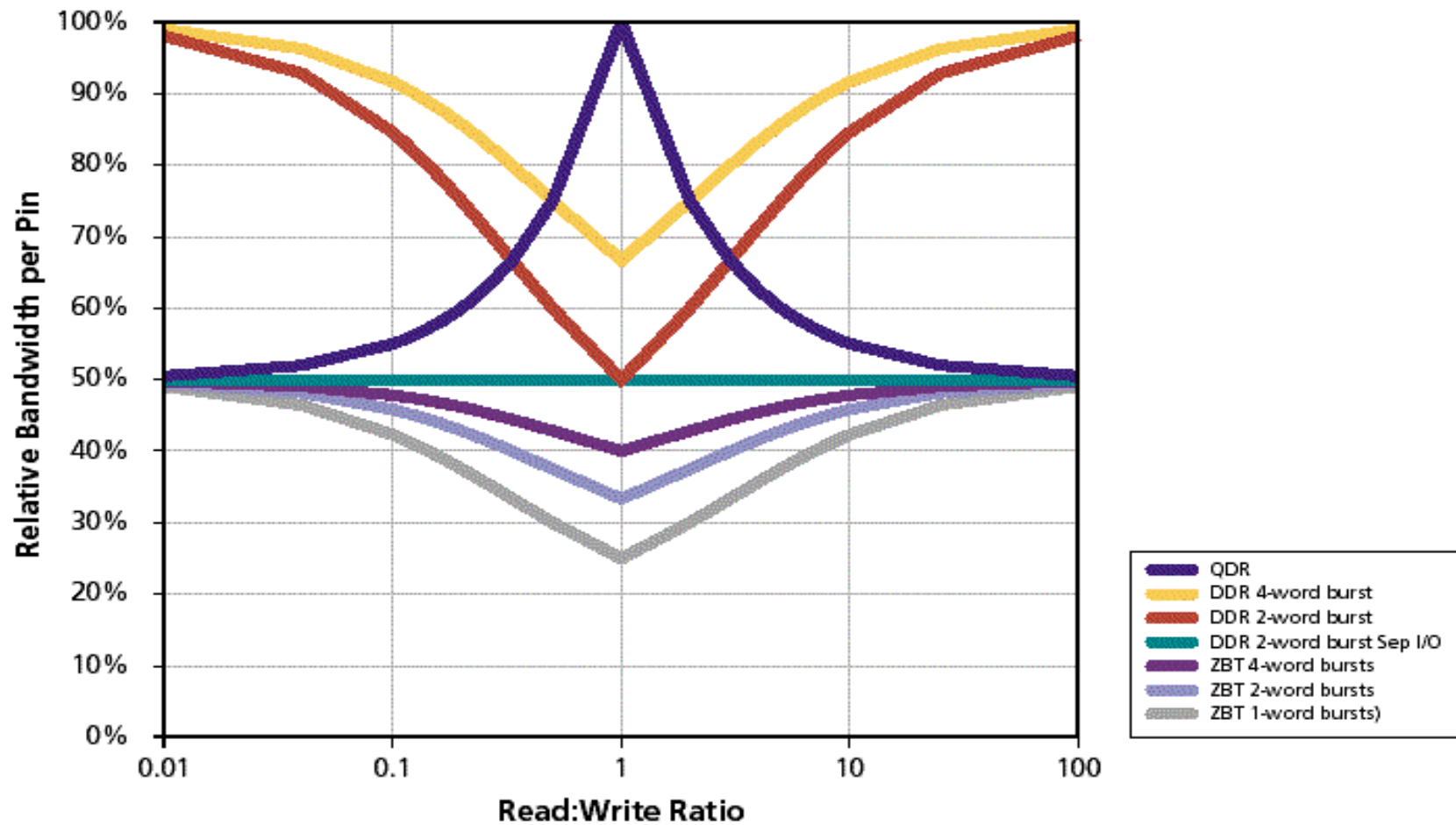
## • DDR CIO

- Un seul bus, DDR
- Un seul type d'accès à la fois
- Optimisée pour :
  - des bus maintenus dans une même direction (ratio RW = 0 ou  $\infty$ )
  - lookup, data streaming (gros paquets)

## • DDR SIO

- Mélange de QDR et DDR CIO, pour remédier aux facteurs limitant fréquentiels de la QDR et fonctionnels de la DDR CIO.
- Pour certaines applications,
  - les cycles morts de turn-around ne sont pas tolérables,
  - et un débit constant d'adresses doit être maintenu.
- Donc
  - bus de données lecture et écriture séparés,
  - un seul type d'accès à la fois.
- Tout se paye : chaque broche n'est occupée que 50% du temps...

# QDR et DDR



Sustainable SRAM Bandwidth at 250 MHz

source : Micron



# SSRAM : bilan

## • Bilan (ouf...)

- Choix des SRAM dicté par
  - Le CPU
  - Le type d'application
  - Le type d'accès le plus souvent effectué dans les applications cibles

Device	QDR	DDR SRAM
2-word burst	R/W < 3:1 2 Addr/Clock	R/W > 3:1 1 Addr/Clock
4-word burst	R/W < 2:1 1 Addr/Clock	R/W > 2 1 Addr/2 Clock

source : [www.qdrsram.com](http://www.qdrsram.com)

Note: R/W means ratio of READ cycles to WRITE cycles, or WRITE cycles to READ cycles

- Et si ça ne suffit pas ?
  - QDR II : 333MHz (48Gb/s sur une x36)
  - DDR II : 333MHz...
  - QDRIII, DDRIII, etc.

# Plan



- Architecture générale
- Composants discrets usuels
- Capteurs / effecteurs
- Flashes
- RAM
- CPU



- Grandes familles
- Répartition
- Comment le choisir ?

### • **Unité centrale :**

- microprocesseur
- microcontrôleur
- DSP
- circuit dédié (SoC)
- système complet préexistant (PC104, VME...)
- souvent équipée de :
  - protection contre les crashes (watchdog)
  - référence temporelle (timer)

## CPU pour systèmes embarqués

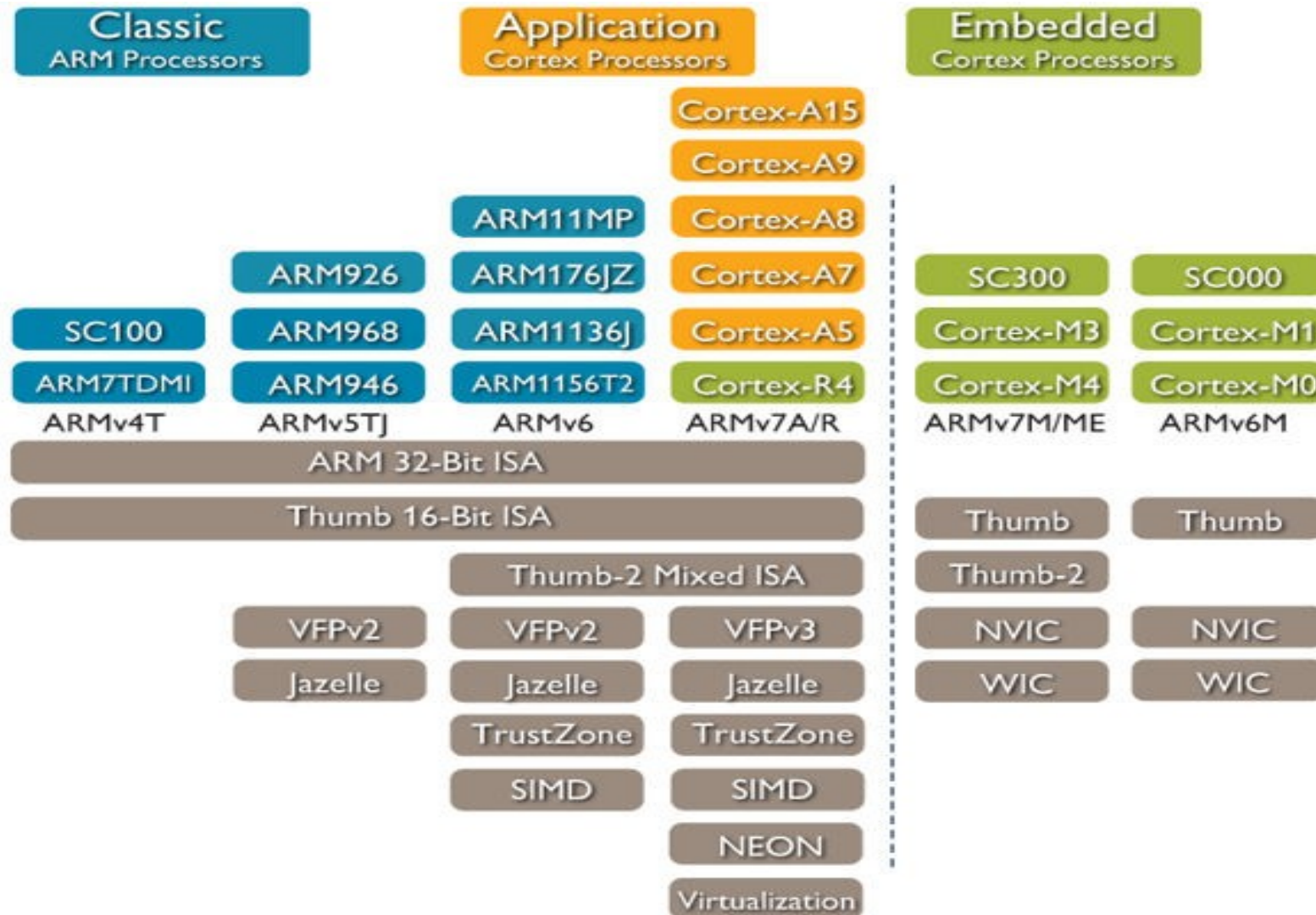
- Entre 40 et 90 processeurs par habitation aux États-Unis, plus 10 processeurs par PC.
- Une voiture moderne comporte une vingtaine de processeurs (130 pour une BMW série 7, 220 pour une Mercedes classe S).
- Dans un PC :
  - processeur central (Intel, AMD, ...)
  - processeurs 8/16/32 bits
    - clavier (Philips ou Intel 8048), souris
    - disque dur, graveur DVD/BR (Z80, H8S, 8051, SuperH, ...), clefs USB
    - écran, HP
    - alimentations

## • **Grandes familles de processeurs pour l'embarqué :**

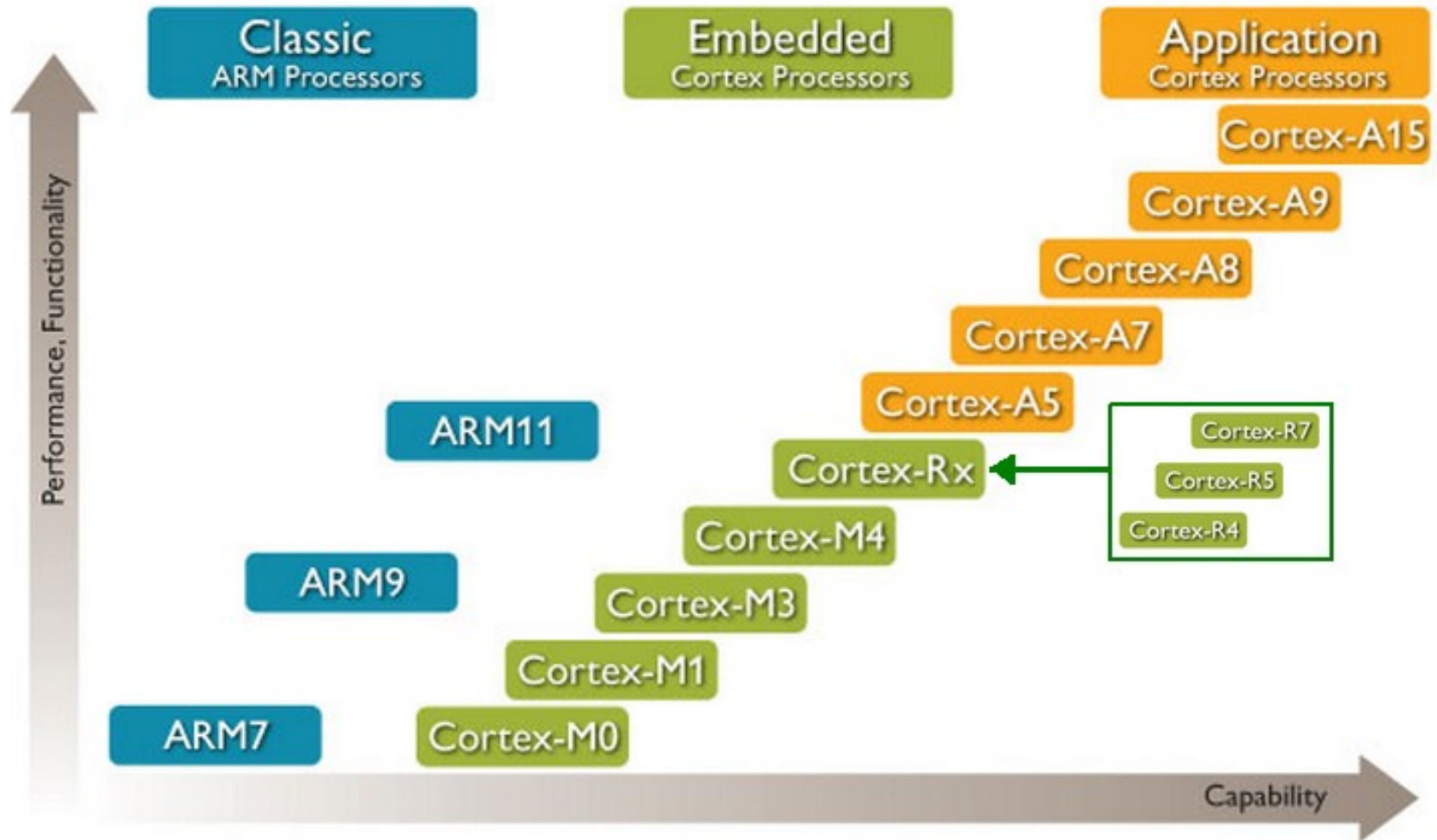
- ARM (Advanced RISC Machines)
  - Le cœur 32/64 bits le plus vendu actuellement dans les portables.
  - Équipe la majorité des téléphones portables.
  - Équipe 20% des notebook en 2015.
  - ARM ne vend que des licences de cœurs de processeurs.
  - Plus de 500 processeurs utilisent des cœurs d'ARM
    - ARM7, StrongArm / Xscale.
    - Cortex.

# CPU pour systèmes embarqués

- ARM (suite)



# CPU pour systèmes embarqués



## • Cortex A

- Deux familles :
  - ARMv7-A : 32 bits (AArch32), supportent aussi le thumb2 (16/32 bits),
  - ARMv8-A : 64 bits (AArch64), supportent un mode d'exécution 32 bits.
- Caractéristiques principales :
  - très hautes performances, haute vitesse (2GHz),
  - simple cœur à 4 cœurs,
  - instructions SIMD et Advanced SIMD (NEON),
  - FPU avancé.
- Utilisés dans :
  - Smartphones, Netbooks, eBooks
  - TV
  - Home Gateway



## • Cortex R

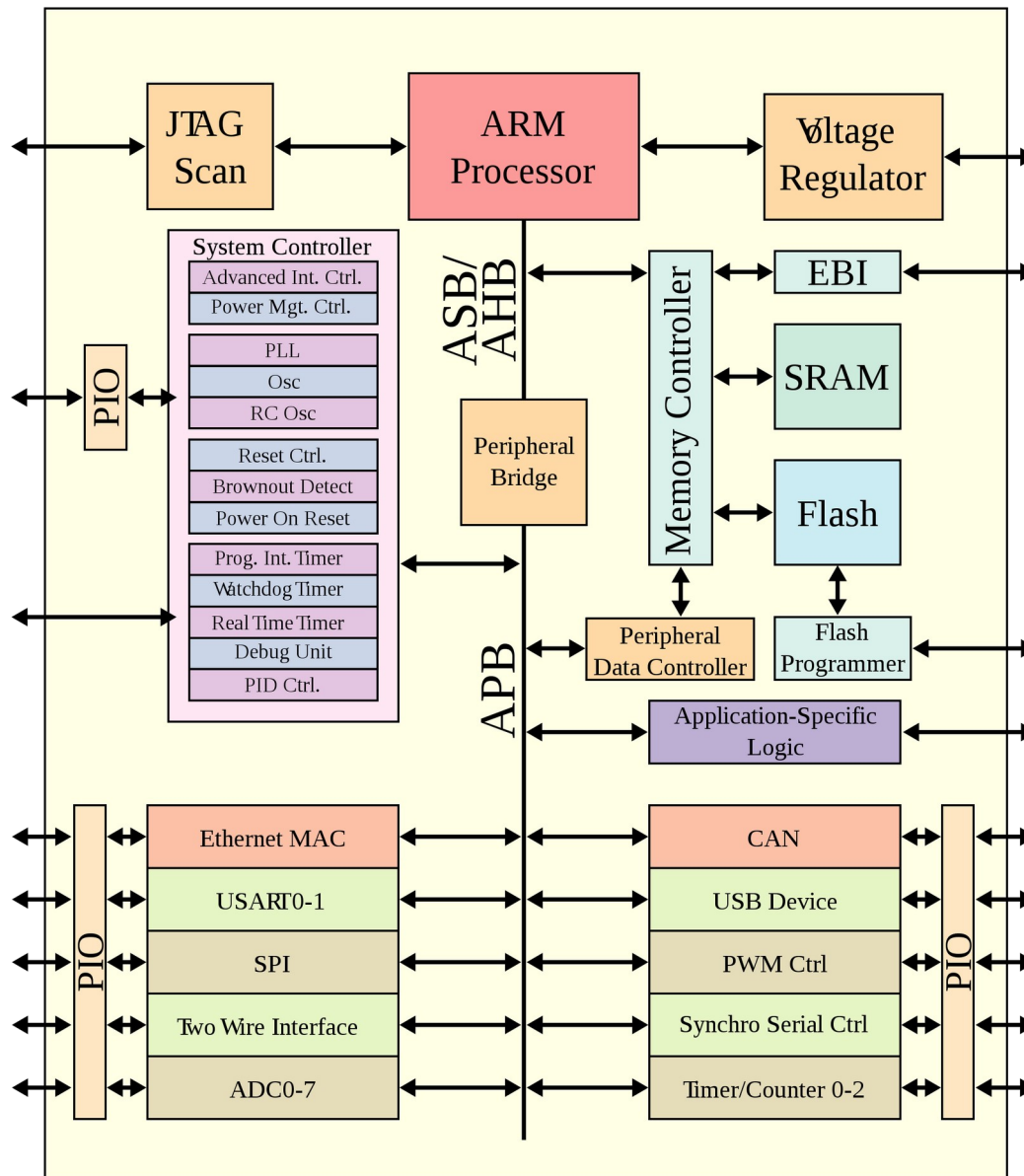
- Faits spécialement pour les applications critiques (temps-réel, haute-disponibilité, résistance aux fautes) : médical et automobile.
- Caractéristiques principales :
  - hautes performances et hautes vitesses,
  - temps-réel : IRQ rapides, bornées et déterministes en temps,
  - fiabilité : MPU, ECC (bus et cache L1), deux cœurs redondants.
- Utilisés dans :
  - médical
  - systèmes de transport (freinage, conduite, ...)
  - contrôleurs de disques
  - réseaux et imprimantes

## • Cortex M

- Faits spécialement pour les applications embarquées faible coût et/ou basse consommation.
- Caractéristiques principales :
  - 6 gammes : M0, M0+, M1 (ARMv6-M) et M3, M4, M7 (ARMv7-M)
  - IRQ rapides
  - haute densité de code (thumb2)
  - facilité de développement
  - Instructions DSP et coprocesseur flottant pour M4 et M7
- Standard industriel.
- Utilisés dans
  - IoT
  - systèmes embarqués non critiques

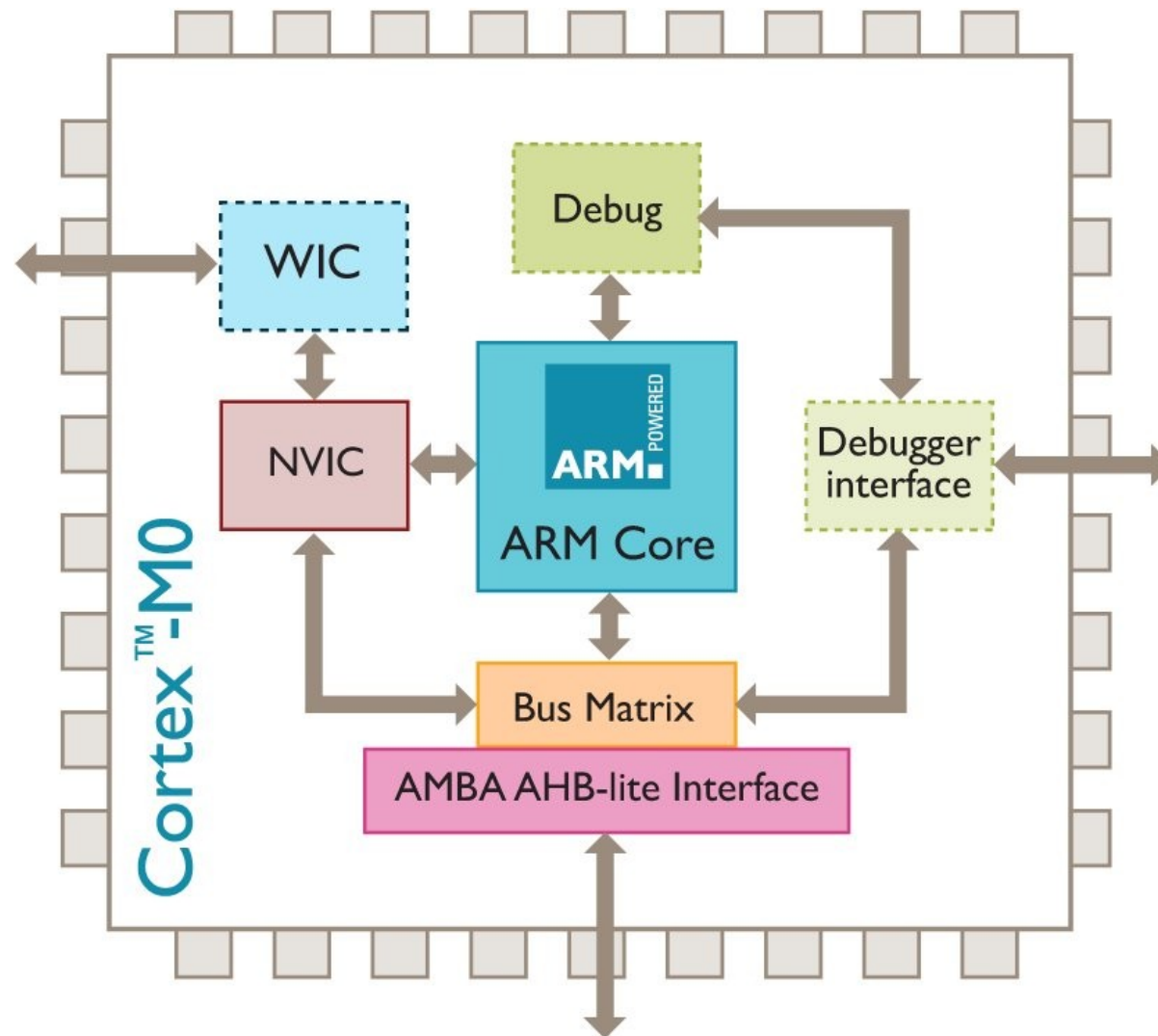
# CPU pour systèmes embarqués

- Architecture des SoC basés sur des Cortex



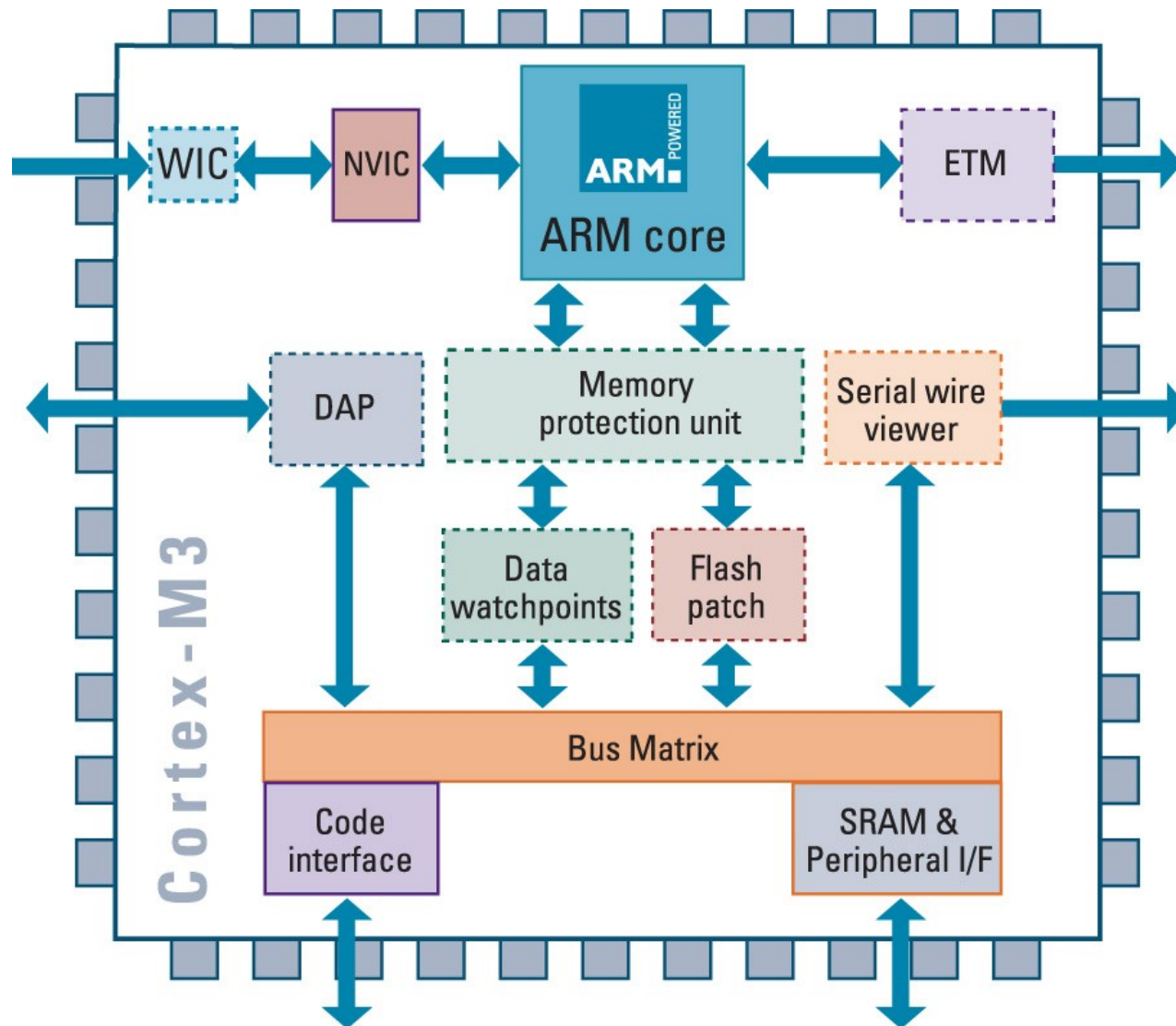
# CPU pour systèmes embarqués

- Architecture des Cortex M0 / M1



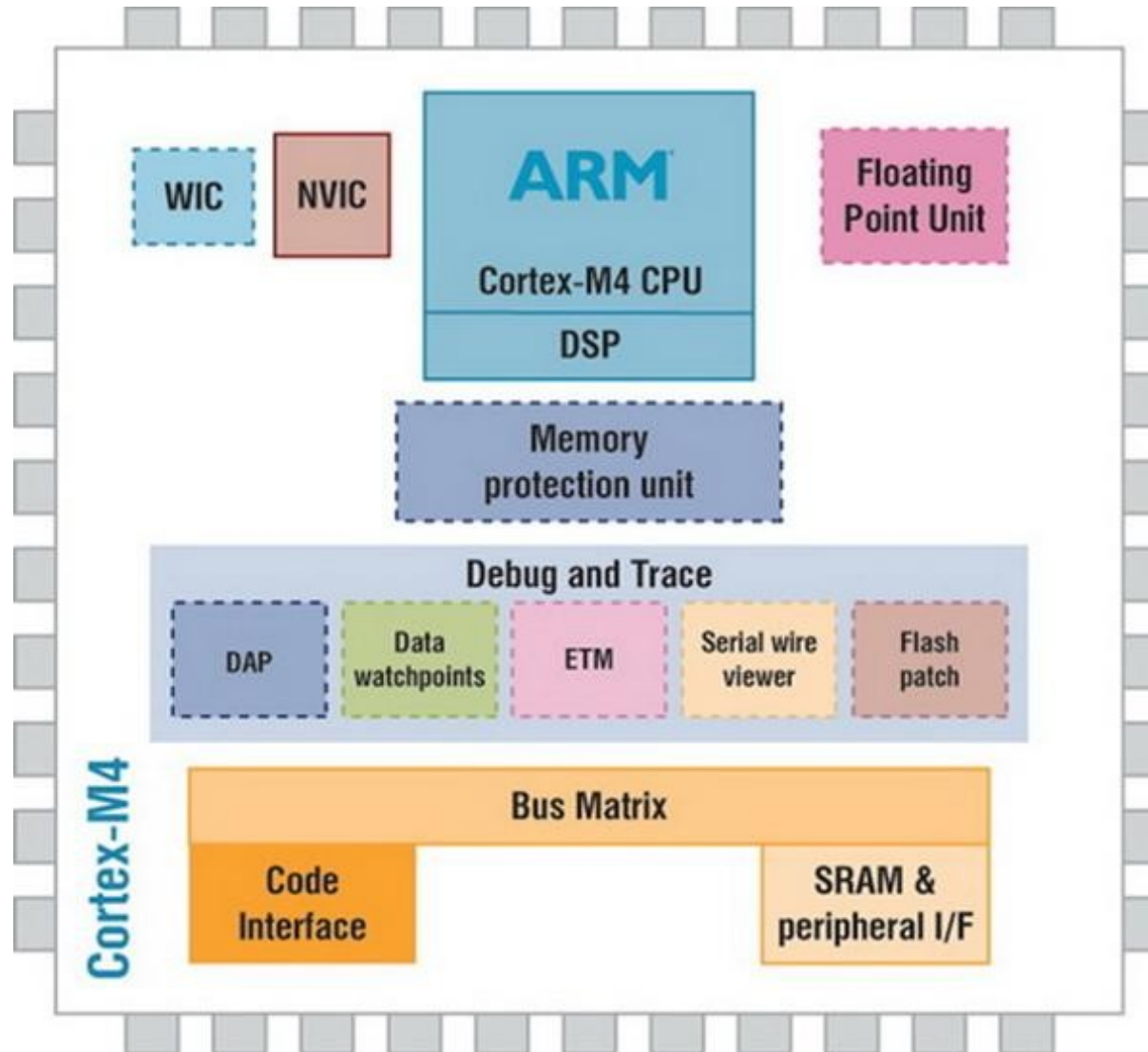
# CPU pour systèmes embarqués

- Architecture des Cortex M3



# CPU pour systèmes embarqués

- Architecture des Cortex M4





# ISA des Cortex M

FPU

DSP / SIMD / Fast MAC

Advanced data processing  
Bit field manipulation

General data processing  
IO control tasks

VABS	VADD	VCMP	VCMPPE	VCVT	VCVTR	VDIV	VLDH
VLDH	VMLA	VMLS	VMOV	VMRS	VMSR	VMUL	VNEG
VNMLA	VNMLS	VNMUL	VPOP	VPUSH	VSQRT	VSTM	VSTR
VSUB	VFMA	VFMS	VFNMA	VFNMS			

**Cortex-M4 FPU**

PKH	QADD	QADD16	QADD8	QASX	QDADD	QDSUB	QSAX
QSUB	QSUB16	QSUB8	SADD16	SADD8	SASX	SEL	SHADD16
SHADD8	SHASX	SHSAX	SHSUB16	SHSUB8	SMLABB	SMLABT	SMLATB
SMLATT	SMLAD	SMLALBB	SMLALBT	SMLALTB	SMLALTT	SMLALD	SMLAWB
SMLAWT	SMLSDB	SMLSDB	SMMLA	SMMLS	SMMUL	SMUAD	SMULBB
						SMULBT	SMULTT
						SMULTB	SMULWT
						SMULWB	SMUSD
						SSAT16	SSAX
						SSUB16	SSUB8
						SXTAB	SXTAB16
						SXTAH	SXTB16
						UADD16	UADDB
						UASX	UHADD16
						UHADDB	UHASX
						UHSAX	UHSUB16
						UHSUB8	UMAAL
						UQADD16	UQADDB
						UQASX	UQSAX
						UQSUB16	UQSUB8
						USAD8	USADA8
						USAT16	USAX
						USUB16	USUB8
						UXTAB	UXTAB16
						UXTAH	UXTB16

ADC	ADD	ADR	AND	ASR	B
CLZ	BFC	BFI	BIC	CDP	CLREX
CBNZ	CBZ	CMN	CMP	DBG	EOR
LDMA	LDMD	LDR	LDRB	LDRBT	LDRD
LDREX	LDREXB	LDREXH	LDRH	LDRHT	LDRSB
LDRSBT	LDRSHT	LDRSH	LDRT	MCR	LSL
LSR	MCR	MLS	MLA	MOV	MOVT
MRC	MRR	MUL	MVN	NOP	ORN
ORR	PLD	PLDW	PLI	POP	PUSH
RBIT	REV	REV16	REVSH	ROR	RRX
			RSB	SBC	SBC
			SDIV	SEV	SMLAL
			SMULL	SSAT	STC
			STMIA	STMDB	STR
			STRB	STRBT	STRD
			STREX	STREXB	STREXH
			STRH	STRHT	STRT
			SUB	SXTB	SXTH
			TBB	TBH	TEQ
			TST	UBFX	UDIV
			UMLAL	UMULL	USAT
			UXTB	UXTH	WFE
			WFI	YIELD	IT

**Cortex-M0/M1**

**Cortex-M3**

**Cortex-M4**

# CPU pour systèmes embarqués

- ARM (suite)
  - Architecture load/store.
  - Jeu de registre quasi-uniforme / jeu d'instruction quasi-orthogonal.
  - Modes d'adressage simples.
  - Bit banding (utile pour les mutex / sémaphores).
- Attention : tous les processeurs ne supportent pas toutes les instructions (M0, M0+, M1).
  - Important de préciser au compilateur / linker le processeur utilisé.
- Couvre du cœur simple de petite taille et très faible consommation (Cortex M0) au multicœur à très hautes performances (Cortex A73).
- Nous utiliserons en TP un STM32F405 (Cortex M4)



# CPU pour systèmes embarqués

- 68k / 68HCxxx (Motorola)
  - Architecture CISC
  - Historiquement, une des plus grandes familles
  - 68k : l'un des processeurs 32 bits les plus utilisés
  - 68HCxx : famille de micro-contrôleurs 8 / 16 / 32 bits
  - Environ 80 millions de pièces de 68k vendues par an
  - Toute la famille 68k est compatible (logiciel)
  - Utilisés dans :
    - automobiles
    - TV
    - électroménager
    - distributeurs de boisson
    - ...

# CPU pour systèmes embarqués

- MIPS (Microprocessor without Interlocked Pipeline Stages)
  - Architecture RISC 32 et 64bits, couvrant une large gamme :
    - ultra faible consommation : SmartMIPS MIPS32 4KSc pour cartes à puce, MIPS 1004Kf (4 cœurs MIPS32) : 0.17mW/MHz !
    - très haute puissance (MIPS64 20Kc, MIPS32 1074K)
  - Licence d'architectures et de cœurs
  - La société MIPS a été rachetée en 2013 par Imagination Technologies
- Équipe la majeure partie des
  - set top box numériques (31%)
  - routeurs
  - imprimantes laser couleur
  - supercalculateurs
- Exemples :
  - R3000 / R4000 / R5000
  - Sigma Designs SMP8652
  - Loogson

## • MIPS

- Architecture load / store
- Jeu d'instruction orthogonal
- Processeur modèle
- Deux familles de jeux d'instruction : MIPS32 et MIPS64
- MIPS32 : deux sous-jeux d'instructions réduits « à la thumb »
  - MIPS16
  - MicroMIPS
- MIPS32 sont répartis en deux familles :
  - 4k pour les micro-contrôleurs
  - 20k pour les hautes performances
- MIPS64 : hautes performances. Exemple : OCTEON III de Cavium (1 à 48 cœurs de MIPS64 à 2.5GHz).

# CPU pour systèmes embarqués

## • x86

- 8086, 286, 386, 486, Pentium, Core, ...
- architecture RISC
- Peu utilisé dans l'embarqué, derrière ARM, 68K, MIPS et SuperH.
- Architecture mal adaptée à l'embarqué :
  - Rapport performance / consommation très bas
  - Consommation de toutes façons élevée
  - Difficiles à programmer
  - Jeu d'instruction totalement non orthogonal
    - EAX - Accumulator Register
    - EBX - Base Register
    - ECX - Counter Register
    - EDX - Data Register
    - ESI - Source Index
    - EDI - Destination Index
    - EBP - Base Pointer (maintenant Frame Pointer)
    - ESP - Stack Pointer

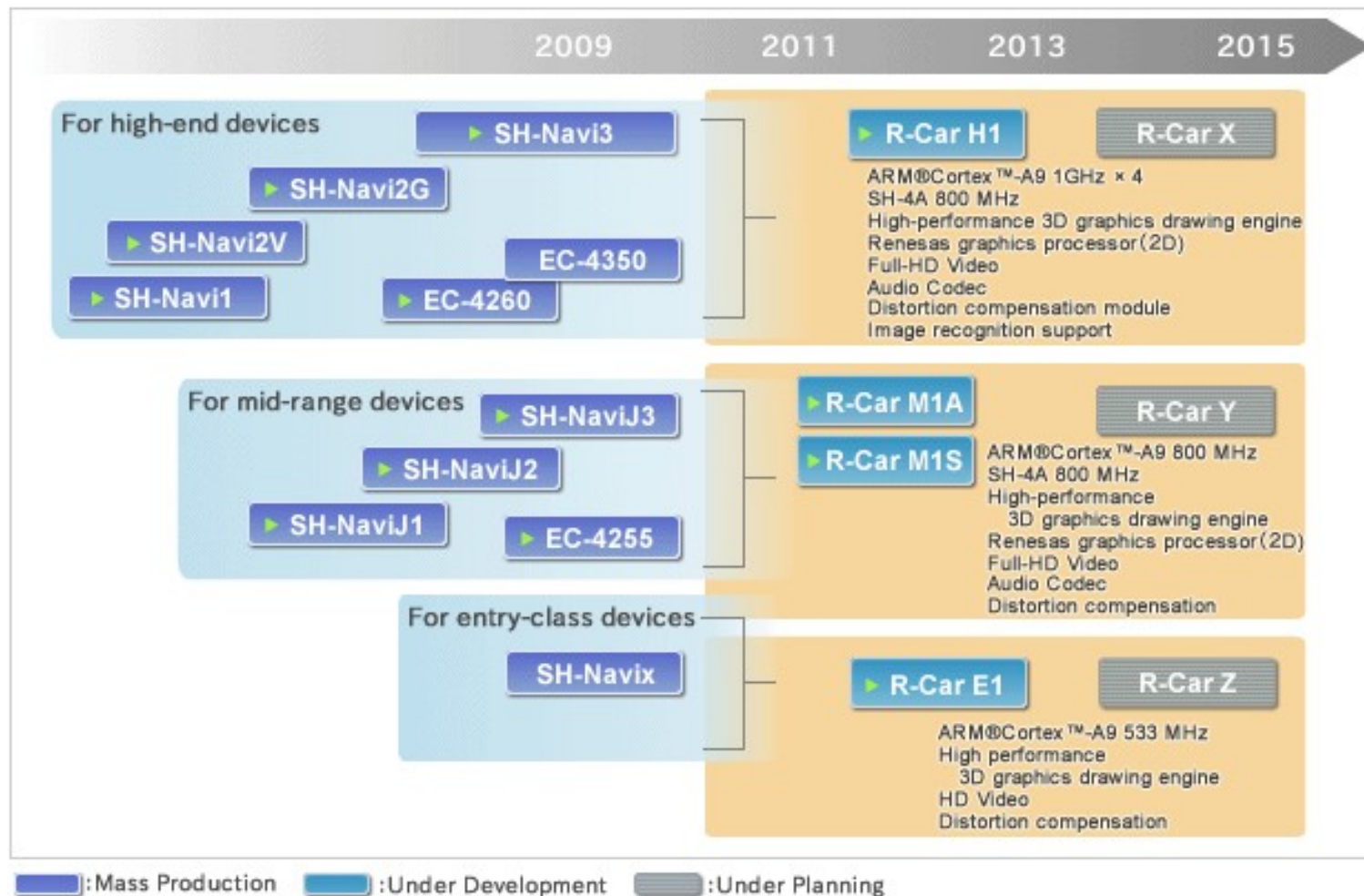
# CPU pour systèmes embarqués

- SuperH

- Hitachi → Hitachi + STMicro → Renesas
- Architecture RISC super scalaire 2 voies (SH4), 16, 32 et 64 bits, Harvard unifié + FPU.
- Exemples : SH7750 (SH4), SH7780 (SH4A)
- Optimisés pour
  - Rapport puissance de calcul / consommation
  - Densité de code
  - Faible taille sur silicium
- Cibles principales :
  - Automobile
  - Contrôle de moteur
  - Équipement réseau
  - Caméras
  - Petit matériel

# CPU pour systèmes embarqués

- SuperH
  - Dans l'automobile, évolue vers SoC SuperH + ARM



source Renesas

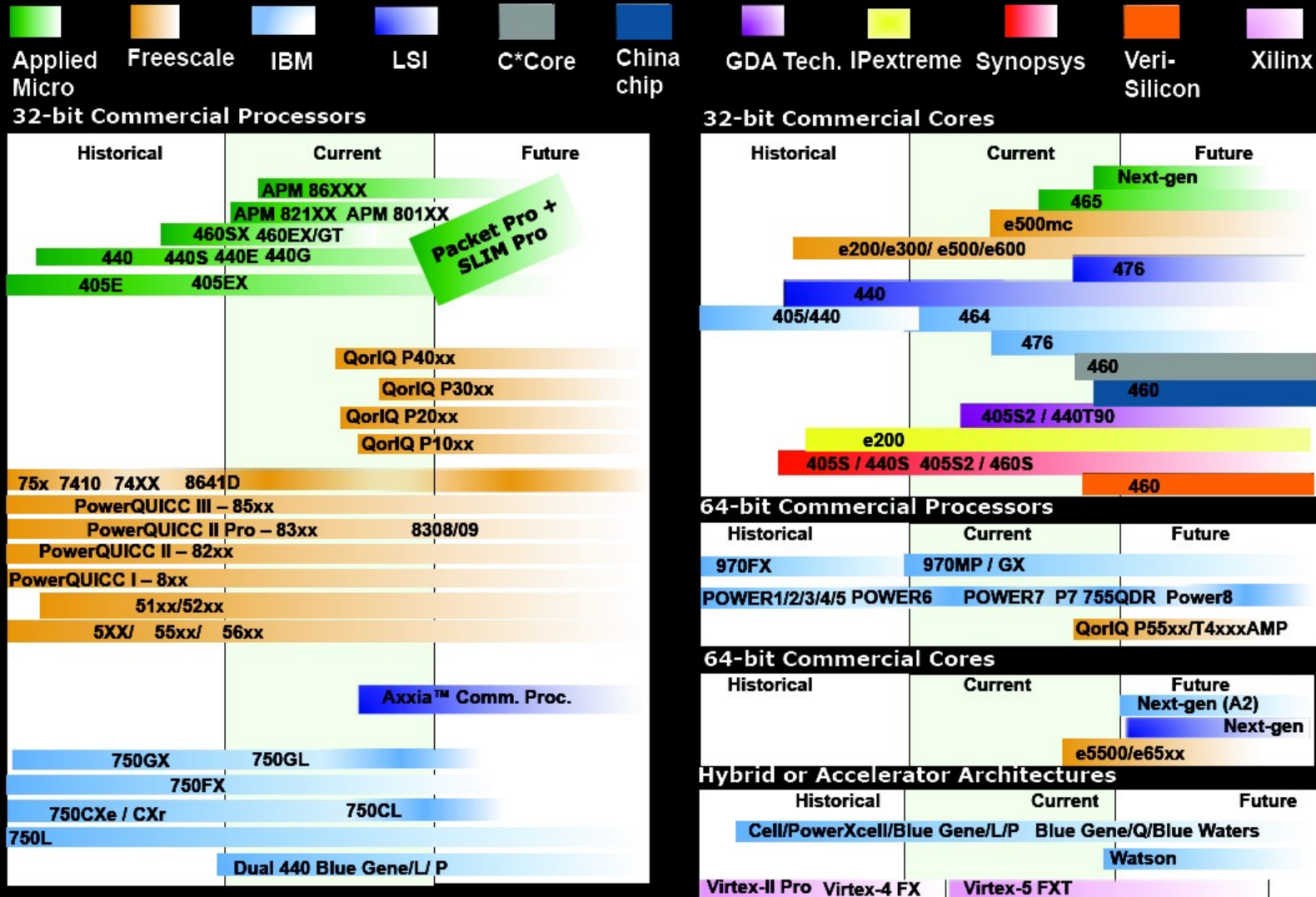
# CPU pour systèmes embarqués

- PowerPC (power.org)
  - Architecture créée en 1991 par IBM et Motorola, étendue en 1999 par le Book E pour assurer une compatibilité entre les différentes architectures PowerPC embarquées.
  - Les architectures sont maintenant spécifiés par des ISA (2.03 en 2005 à 2.06B en 2010)
  - Architecture RISC, 32 ou 64 bits, de 33Mhz à 5GHz.
  - Seule architecture à être embarquée dans toutes les missions sur Mars.
  - 3ème place mondiale des contrôleurs 32 bits.
  - Quel que soit le domaine, placé entre 1 et 3ème mondial.



# CPU pour systèmes embarqués

## Power Architecture Silicon Roadmap The Heart of Ecosystem





# CPU pour systèmes embarqués

- PIC (Microchip)
  - Picocontrôleurs, RISC, 8, 16 ou 32 bits
  - Taillés pour des applications réduites :
    - peu de RAM on chip
    - rarement de bus processeur
    - OTP, EPROM ou Flash (512 à 536k mots)
    - minimisent le nombre de composants externes
  - Périphériques embarqués
    - UART, I2C, SPI, CAN
    - Timers, watchdog, PWM
    - analogiques : convertisseurs AD, comparateurs
  - PIC16F / 18F pas du tout adaptés au C.
  - Les versions 32 bits sont des MIPS32 4k, donc supportés par gcc

# CPU pour systèmes embarqués

- AVR (Atmel)
  - Comparables aux PIC en fonctionnalités, mais architecture beaucoup plus propre que les 16F / 18F
    - Conçus pour l'exécution de programmes C
    - Beaucoup de registres (au moins 16)
    - Jeu d'instruction presque orthogonal
    - Harvard réel (!!!), 2 stage pipeline
    - Instructions spéciales (LPM) pour accès à .rodata en flash
    - Supportés par gcc / gdb
  - Beaucoup de périphériques :
    - UART, I2C, SPI, CAN, USB, Ethernet
    - Timers, watchdog, PWM
    - analogiques : convertisseurs AD, comparateurs, convertisseurs DA
    - TouchSense

# Plan

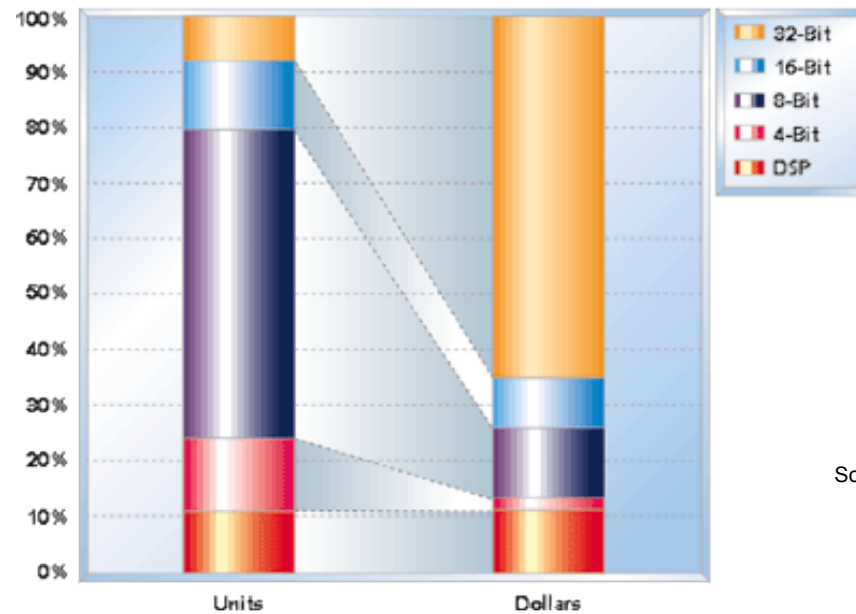


- Architecture générale
- Composants discrets usuels
- Capteurs / effecteurs
- Flashes
- RAM
- CPU
  - Grandes familles
  - Répartition
  - Comment le choisir ?



# CPU pour systèmes embarqués

En 2013  
(worldwide)



Source : EE Times

En Chine  
(la tendance 8 bits est  
plus marquée dans le  
reste du monde)



Source : [www.ic.big-bit.com](http://www.ic.big-bit.com)

## CPU pour systèmes embarqués

- Il n'existe pas de CPU dominant sur le marché de l'embarqué (par contre, il existe des leaders par domaines).
- Le nombre de types de processeurs embarqués augmente.

Entreprise	Coeurs	Parts (%)
Intel	8051	19
Renesas	740, H8/S, M32R	17
Freescale	68xxx	15
Microchip	misc	12
ARM	misc	10
NEC	V850, K3/K4, 78K0	9
ST	misc	6
Atmel	AVR	3

2008 (source Dataweek)

# CPU pour systèmes embarqués

## Segments for ARM in 2010

	Devices Shipped (Million of Units)	2010 Devices	Chips/ Device	TAM 2010 Chips	2010 ARM	2010 Share
Mobile	Smart Phone	280	2-5	1,200	1,100	90%
	Feature Phone	760	1-3	1,900	1,700	90%
	Low End Voice	570	1	570	540	95%
	Portable Media Players	150	1-3	300	220	70%
	Mobile Computing* (apps only)	230	1	230	25	10%
Non-Mobile	PCs & Servers (apps only)	220	1	220	0	0%
	Digital Camera	130	1-2	200	160	80%
	Digital TV & Set-top-box	350	1-2	450	160	35%
	Networking	670	1-2	750	185	25%
	Printers	120	1	120	75	65%
	Hard Disk & Solid State Drives	670	1	670	560	85%
	Automotive	1,800	1	1,800	180	10%
	Smart Card	5,400	1	5,400	330	6%
	Microcontrollers	5,800	1	5,800	560	10%
	Others **	1,700	1	1,800	270	15%
<b>Total</b>		<b>19,000</b>		<b>22,000</b>	<b>6,100</b>	<b>28%</b>

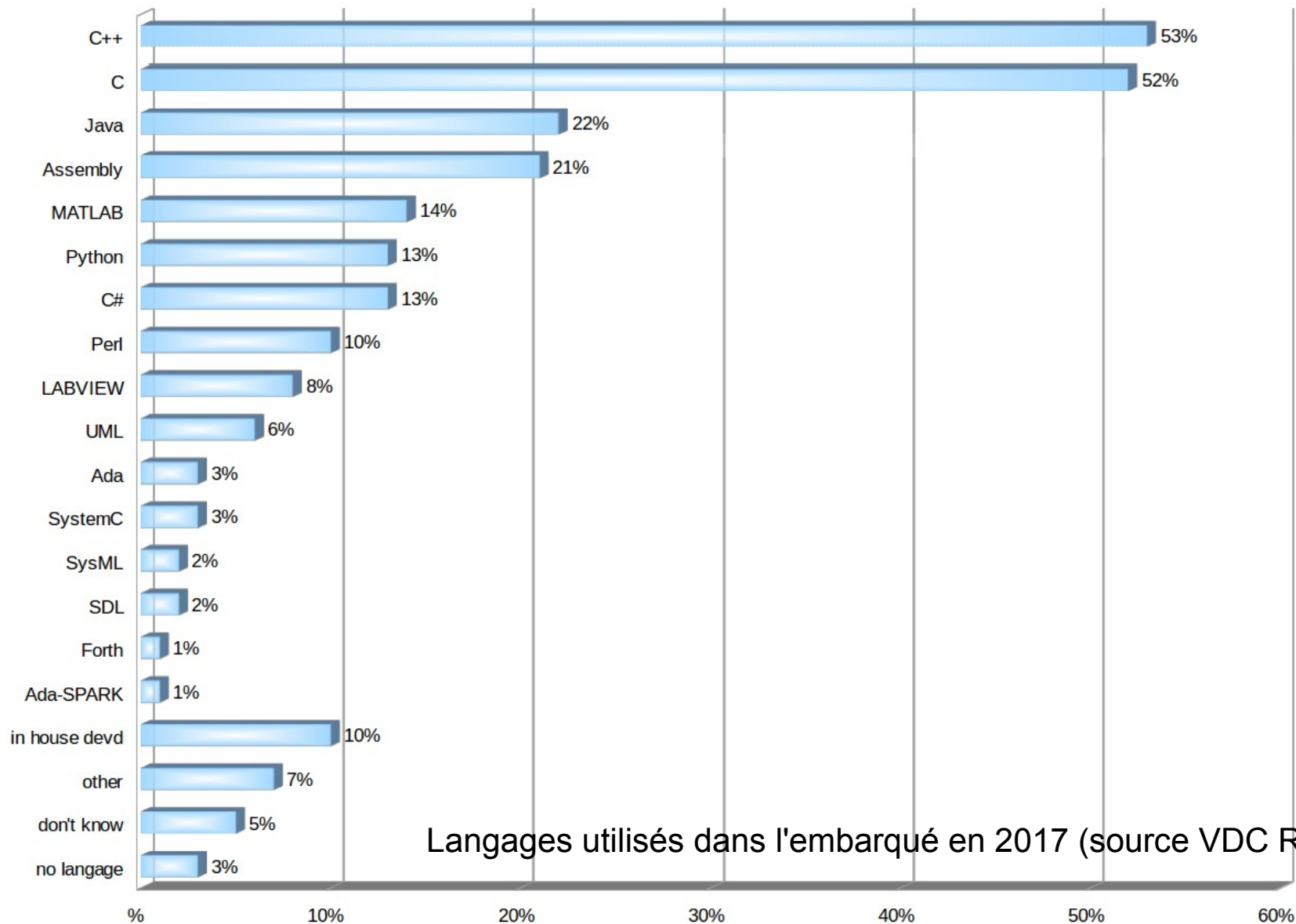
source ZDNet

Source: ABI, Gartner, Semico, Instat, IDC, and ARM estimates

\* Including tablets, netbooks and laptops

\*\* Includes other applications not listed such as headsets, DVD, game consoles, etc

# Langages des systèmes embarqués



Langages utilisés dans l'embarqué en 2017 (source VDC Research)

# Plan



- Architecture générale
- Composants discrets usuels
- Capteurs / effecteurs
- Flashs
- RAM
- CPU
  - Grandes familles
  - Répartition
  - Comment le choisir ?





## • Comment choisir un CPU ?

- Prix, disponibilité...
- Caractéristiques physiques
  - Tension(s) d'alimentation
  - Consommation
    - fréquence de fonctionnement
    - alimentations séparées cœur - entrée/sorties
    - design logique statique / mise en veille
  - Boîtier → impact sur
    - routage et PCB (prix)
    - dissipation
    - fréquence de fonctionnement

# CPU pour systèmes embarqués

- Performances
  - MIPS / FLOPS : **JAMAIS**
  - Doit prendre en compte l'application cible et le compilateur
  - EEMBC
- Nombre et types de périphériques embarqués
- Interfaces
  - Le nombre de broches impacte directement le routage et le prix du produit.
  - Un bus est soumis à des contraintes de fan out : limité en fréquence, surtout s'il est multiplexé.
  - Les interfaces rapides posent des problèmes d'intégrité et de CEM.

- Adaptation aux applications
  - Architecture
    - Nombre de registres, ISA (langage, compilateur)
    - Densité de code (CISC / RISC) : critère important (coût flash)
    - Manipulations de bits
    - Gestion des alignements (vidéo, réseaux)
    - Endianness (réseau)
    - Fenêtres de registres (SPARC)
    - Prédiction des branchements
    - Cache
      - Présence ou non
      - Possibilité de le désactiver
      - Write-back / write-through ?

# CPU pour systèmes embarqués

- Adaptation aux applications (suite)
  - Source d'IRQ externe au processeur (OS préemptif)
  - Protection crash (watchdog)
  - Prédicibilité / déterminisme
    - des performances
    - du comportement

# CPU pour systèmes embarqués

- Il est donc capital d'avoir une idée de l'application à exécuter en termes de :
  - puissance de calcul requise,
  - débit de données,
  - matériel nécessaire,
  - contraintes de sécurité.
- Ainsi qu'une idée des suites de compilation / debug possibles.
- Le dialogue matériel – logiciel n'est pas juste une question de politesse, il est capital !

### • Circuits spécifiques :

- Utilisés quand la quantité d'informations à traiter est trop grande pour le processeur central.
- Coprocesseurs
- SoC dédiés : ASSP
- FPGA :
  - souplesse,
  - nécessitent souvent une mémoire de configuration,
  - coût élevé,
  - plus ils sont gros, plus la consommation statique est grande.

## • Interfaces de communication

- RS232, RS422
- Ethernet, USB, HT, PCIe
- I2C, I2S, SPI
- LIN, CAN, Flexray, MOST
- IrDA, SPDIF
- 802.11, 802.15.4, Bluetooth / BLE, LoRa
- Seront étudiés dans le cours sur les bus...

## ● Alimentation

- souvent le point le plus critique des systèmes embarqués
- facteurs décisifs
  - stabilité (load regulation, line regulation)
  - précision (5 à 10ppm)
  - autonomie
  - taille (poids)
  - sécurité
- sources
  - externe (secteur)
  - autonome (batterie)
  - dépendantes de l'environnement / energy harvesting (panneaux solaires, générateurs magnétiques, ...)
- types
  - linéaires
  - à découpage



# Licence de droits d'usage



Contexte académique } sans modification

***Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.***

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après, et à l'exclusion de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage dans un cadre académique, par un utilisateur donnant des cours dans un établissement d'enseignement secondaire ou supérieur et à l'exclusion expresse des formations commerciales et notamment de formation continue. Ce droit comprend :

- le droit de reproduire tout ou partie du document sur support informatique ou papier,
- le droit de diffuser tout ou partie du document à destination des élèves ou étudiants.

Aucune modification du document dans son contenu, sa forme ou sa présentation n'est autorisée.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel, non exclusif et non transmissible.

Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur :

[alexis.polti@telecom-paristech.fr](mailto:alexis.polti@telecom-paristech.fr)