

Informatique embarquée

Les logiciels embarqués : introduction

Philippe.Plasson@obspm.fr

Sommaire

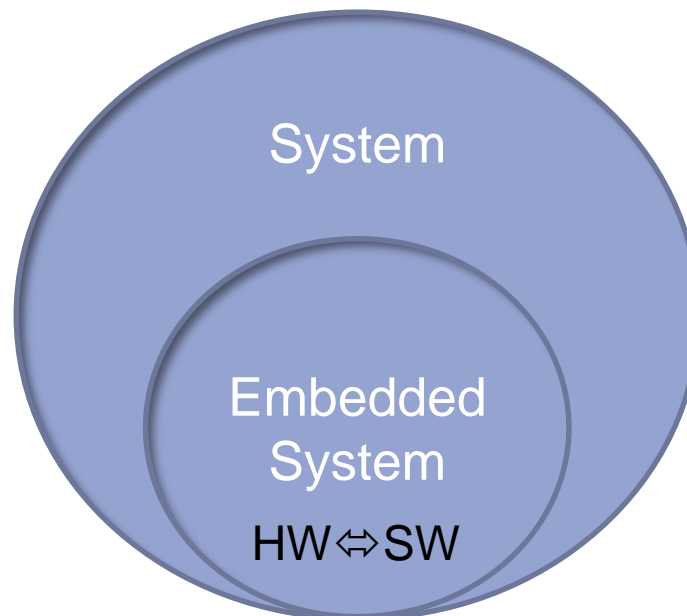
1. Contexte : les systèmes embarqués
2. Logiciels embarqués : définition et caractéristiques
3. Cibles matérielles : aperçu des familles de processeurs et SoC pour l'embarqué (ARM, LEON, ...)

Sommaire

1. Contexte : les systèmes embarqués
2. Logiciels embarqués : définition et caractéristiques
3. Cibles matérielles : aperçu des familles de processeurs et SoC pour l'embarqué (ARM, LEON, ...)

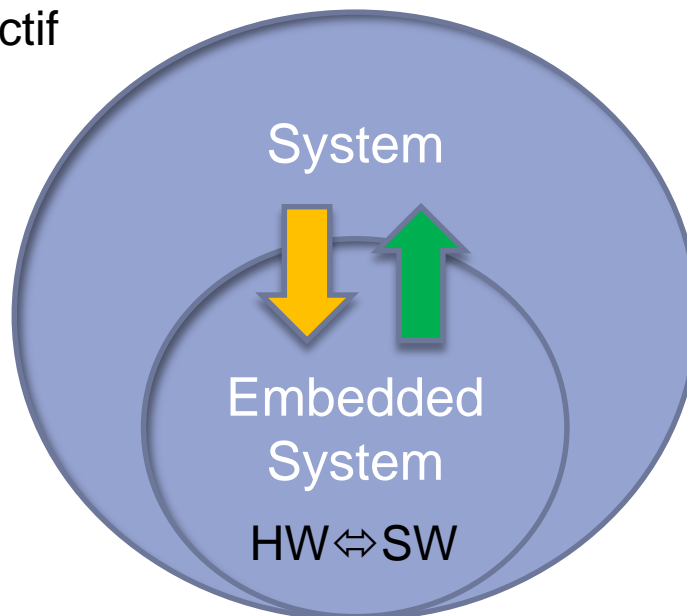
Système embarqué - Définition

- Système embarqué (embedded system) =
 - Système informatique à base de processeur (donc avec du logiciel), intégré dans un dispositif complet plus vaste comprenant souvent du matériel et des pièces mécaniques.
 - Système combinant matériel (H/W) et logiciel (S/W) faisant partie d'un système plus large.



Système embarqué - Définition

- Système embarqué (embedded system) =
 - Système à usage spécialisé (fonction dédiée à une tâche particulière).
 - Système ayant souvent des contraintes de calcul en temps réel
 - Système embarqué temps réel
 - Système en interaction continue avec son environnement et qui s'exécute à un rythme déterminé par cet environnement
 - Système réactif

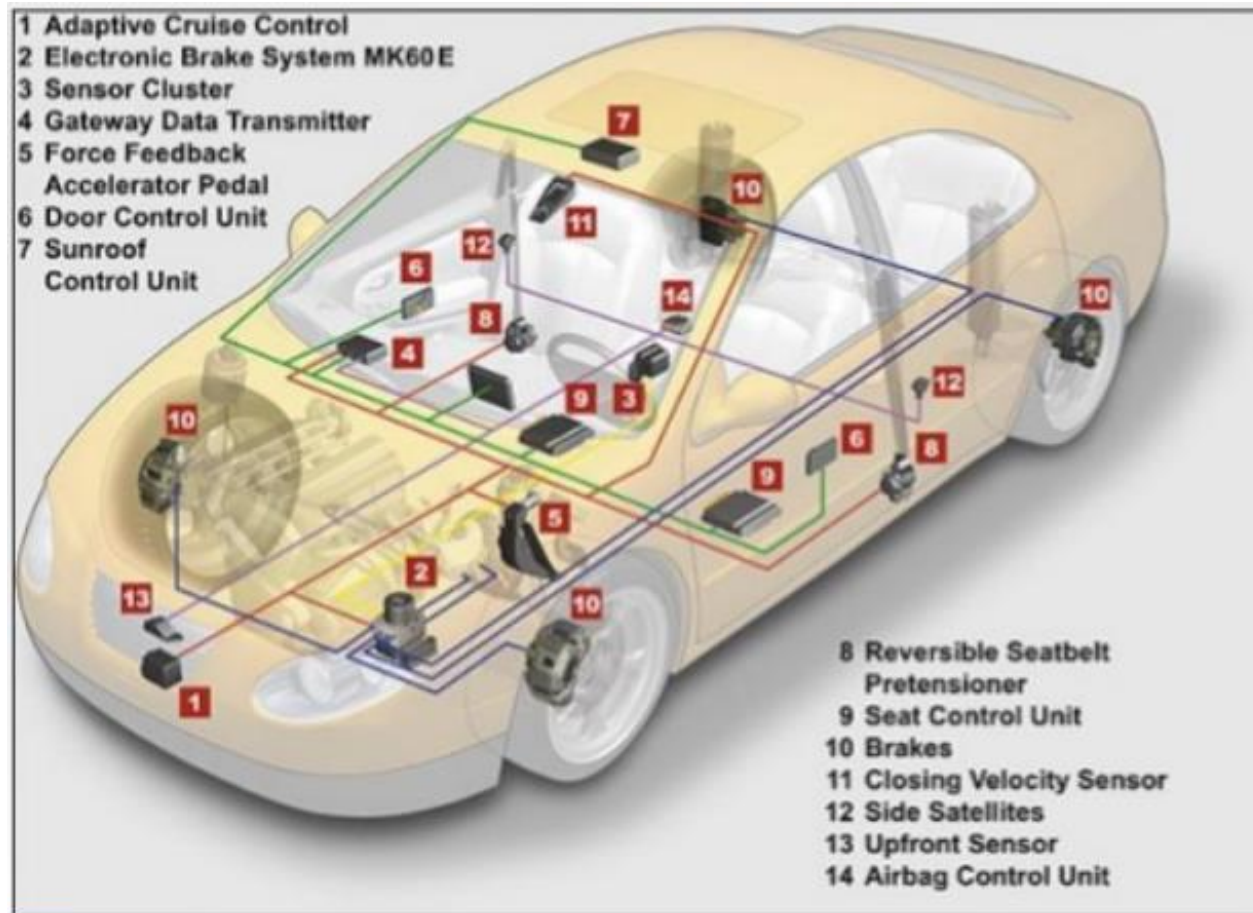


Système embarqué - Définition

- Les systèmes embarqués contrôlent de nombreux dispositifs couramment utilisés aujourd'hui.
- 98% des microprocesseurs fabriqués sont utilisés comme composants de systèmes embarqués.
- Les ordinateurs personnels (desktop, laptop) ou stations de travail (workstation) ne sont pas des systèmes embarqués.
 - Systèmes à usage généraliste.

Système embarqué

Exemple : l'automobile



Crédit : <https://www.slideshare.net/asertseminar/embedded-system-in-automobiles>

Comparaison entre les systèmes embarqués et les systèmes à usage général

Systèmes embarqués

- Application avec un usage connu au moment de la conception.
- Non programmable par l'utilisateur final.
- Exigences en termes de performance fixées dès la conception.

Systèmes généralistes

- Grandes classes d'applications non connues au moment de la conception.
- Programmable par l'utilisateur final.

Comparaison entre les systèmes embarqués et les systèmes à usage général

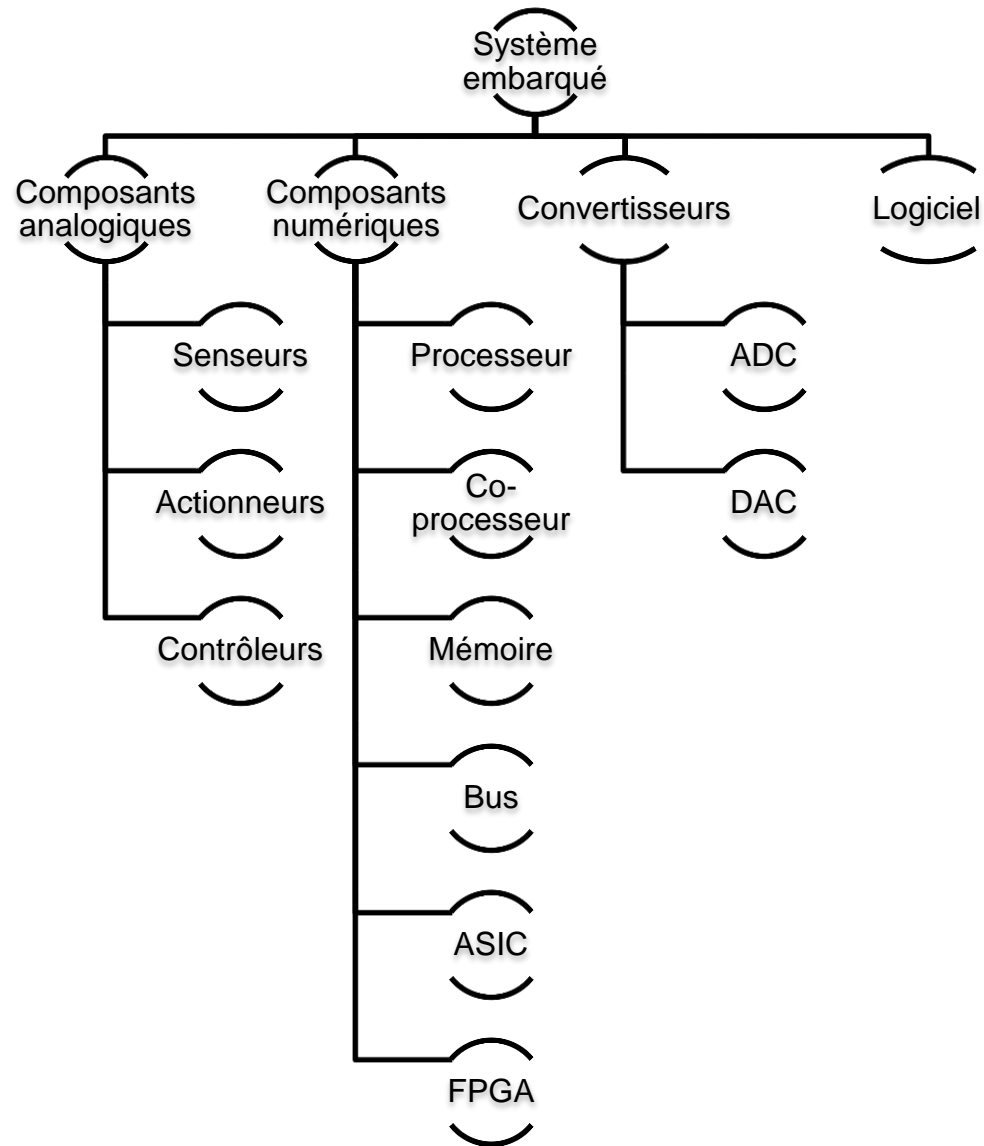
Systèmes embarqués

- Interfaces utilisateurs dédiées : pas de souris, de clavier, d'écrans.
- Interfaces avec l'environnement à travers des capteurs (sensors) et des actuateurs (actuators)
- Critères :
 - coût
 - consommation électrique
 - prédictibilité
 - ...

Systèmes généralistes

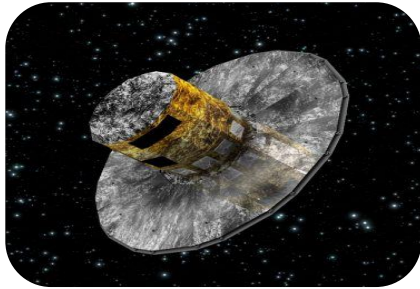
- Interfaces utilisateurs généralistes : souris, claviers, écrans.
- Critères
 - puissance de calcul
 - richesse de l'éco-système logiciel
 - ...

Composants d'un système embarqué

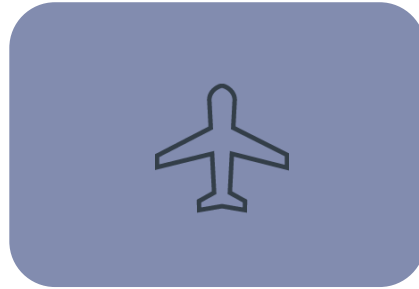


Systèmes embarqués

Domaines d'application



Spatial (satellite,
instrumentation)



Aéronautique



Défense (missiles,
...)



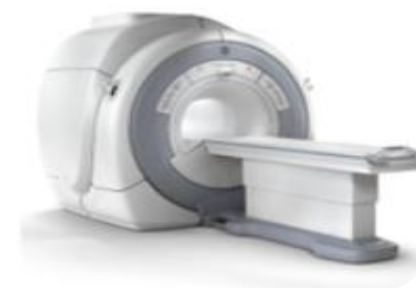
Ferroviaire



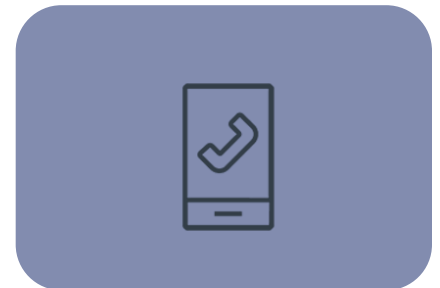
Automobile



Contrôle /
commande
industriel



Médical



Téléphonie

Systèmes embarqués

Domaines d'application



Domotique



Appareils
domestiques



Internet des
objets (IoT)



Drones



Bracelets /
Montres
(Wearables)



Jouets



Imprimantes



Télévisions



Terminaux de
paiement

Systèmes embarqués

Domaines d'application



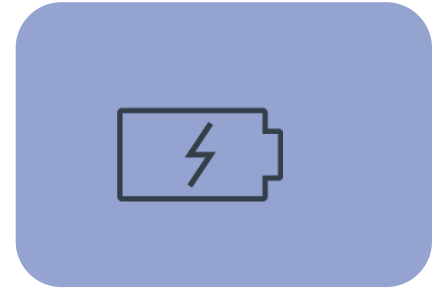
Contrôle de moteur



Traitement audio



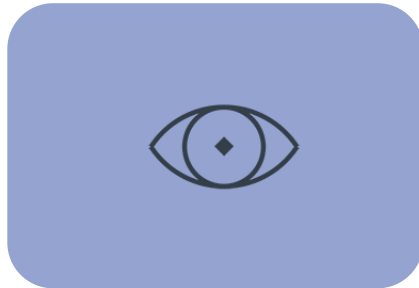
Traitement de
l'image et de la
vidéo



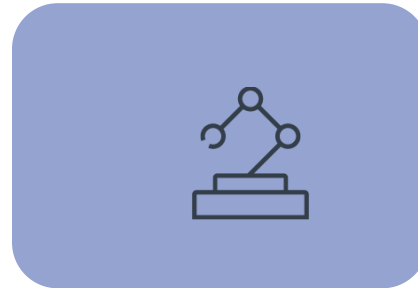
Gestion de la
puissance



Surveillance



Systèmes
d'authentification
biométriques



Nano-robots
médicaux



Stockage de
masse

Systèmes embarqués

Exemples



Robots industriels



Recepteurs GPS



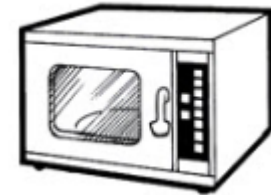
Lecteurs MP3



Appareils photos
numériques



Lecteurs DVD



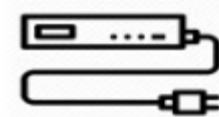
Micro-ondes



Matériel
réseau
(routeurs WiFi,
...)



Photocopieurs



Box TV

Systèmes embarqués

Exemple : le calculateur bord de RPW sur Solo



Sommaire

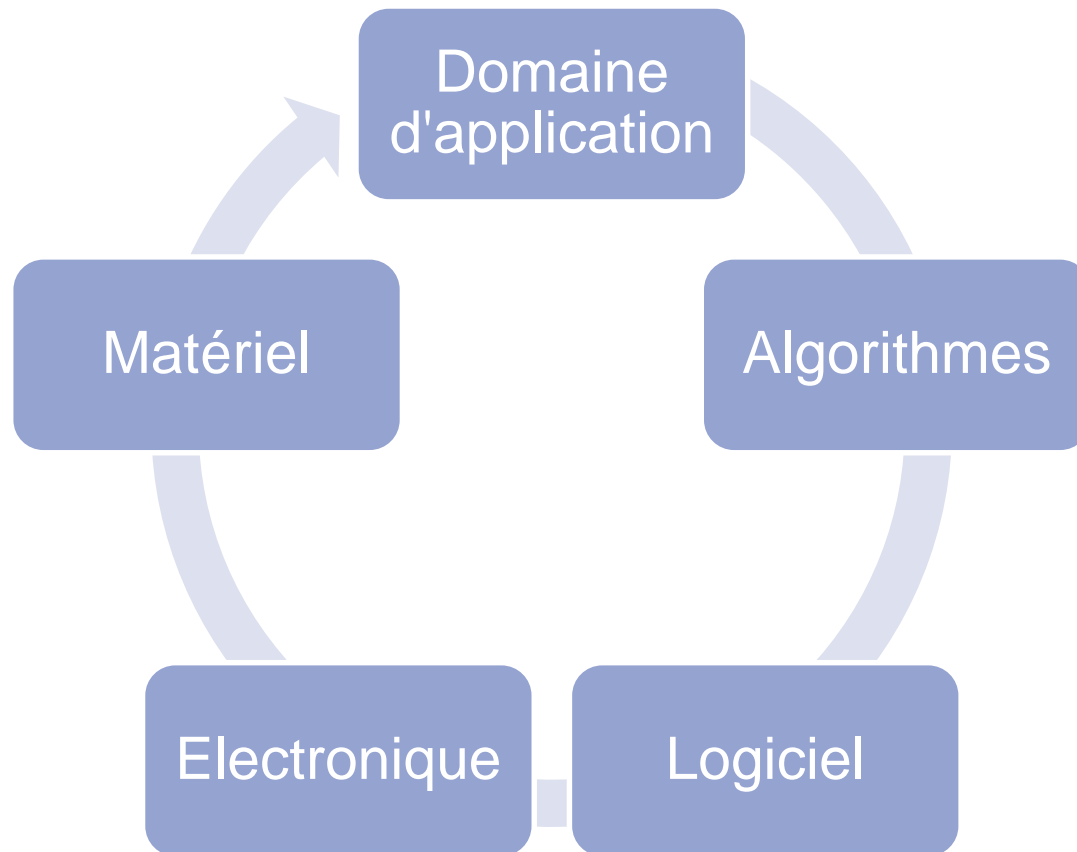
1. Contexte : les systèmes embarqués
2. Logiciels embarqués : définition et caractéristiques
3. Cibles matérielles : aperçu des familles de processeurs et SoC pour l'embarqué (ARM, LEON, ...)

Logiciels embarqués - Définition

- Logiciels embarqués (embedded software) =
 - Logiciels écrits pour contrôler des systèmes embarqués
 - Logiciels écrits pour contrôler des machines ou des appareils qui ne sont généralement pas considérés comme des ordinateurs
- Le comportement d'un logiciel embarqué dépend de ses entrées et de son état courant => modélisable sous forme de machines à états

Logiciels embarqués - Définition

- Concevoir et développer un logiciel embarqué implique un ensemble diversifié de compétences qui s'étendent au-delà des frontières disciplinaires traditionnelles :



Caractéristiques des logiciels embarqués

Lien étroit avec le matériel

- Proximité du matériel
 - Intrication matériel / logiciel
 - Interaction avec le monde physique
- Un logiciel embarqué est typiquement spécialisé pour le matériel particulier sur lequel il fonctionne.
- Les logiciels embarqués doivent inclure tous les pilotes de périphériques nécessaires dès la phase de développement.
- Les pilotes de périphériques sont écrits pour le matériel spécifique.
- Les logiciels embarqués dépendent fortement du processeur choisi.

Caractéristiques des logiciels embarqués

Lien étroit avec le matériel

- Nécessité pour les développeurs d'avoir une bonne connaissance de l'environnement matériel d'exécution (architecture du processeur, registres, endianisme, quantité de mémoire, accès aux périphériques).
- Contraintes non fonctionnelles :
 - charge CPU
 - temps de réponse
 - occupation mémoire
 - consommation électrique
- Fonctions des logiciels embarquées contrôlées par des interfaces machines plus que par des interfaces humaines.

Caractéristiques des logiciels embarqués

Des ressources matérielles limitées

- Caractéristique des systèmes embarqués :
 - Consommation électrique limitée
 - Encombrement limité
 - Masse limitée
- Contrairement aux logiciels applicatifs classiques, les logiciels embarqués ont des exigences et des capacités matérielles fixes.
- Impact direct sur la quantité de mémoire et la puissance de calcul dont on peut disposer, donc sur le logiciel.

Caractéristiques des logiciels embarqués

Des ressources matérielles limitées

- Impact sur l'architecture des systèmes numériques (partage hardware / software)
- Compromis flexibilité / coûts
 - Si une fonction ne peut pas être implémentée dans le logiciel du fait d'un coût CPU trop élevé, alors il faut l'implémenter dans le matériel (dans des FPGA) ➔
 - perte en flexibilité
 - non re-programmabilité
 - contraintes sur le plan de développement

Caractéristiques des logiciels embarqués

Des logiciels optimisés

- Les logiciels embarqués sont des logiciels pour lesquels les critères d'optimisation sont cruciaux :
 - minimiser l'énergie consommée
 - minimiser la taille du code (particulièrement vrai pour les SoC)
 - maximiser la robustesse
 - maximiser les performances

Caractéristiques des logiciels embarqués

Criticité

Réactivité

Autonomie

Disponibilité

Maintenabilité

Robustesse

Sûreté

Sécurité

Fiabilité

Caractéristiques des logiciels embarqués

Des logiciels fiables et tolérants aux fautes

- Robustesse, sûreté et fiabilité → poids prépondérant des tests, de la validation et de la vérification dans le processus de développement (processus de qualification)
 - Objectifs pour les tests unitaires de couverture de 100% du code
 - 100% des lignes pour les logiciels faiblement critiques
 - 100% des branches pour les logiciels fortement critiques
 - Nécessite des outils permettant d'automatiser les tests
- AMDEC logicielle (SFMECA en anglais) = Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité
- Mise en place de mécanismes pour détecter les erreurs, les isoler et reconfigurer le système (FDIR = Failure Detection Isolation and Recovery)

Caractéristiques des logiciels embarqués

Des logiciels fiables et tolérants aux fautes

- Vérification automatique du code via des techniques d'analyse statique (outil PolySpace de la société MathWorks, CppCheck, Parasoft, ...)
- Métriques pour mesurer la qualité du code (taux de commentaires, complexité cyclomatique, longueur des fonctions, ...)
- Standards de codage définissant des règles de programmation pouvant être vérifiées automatiquement :
 - MISRA C
 - <https://www.misra.org.uk/Activities/MISRAC/tabid/160/Default.aspx>
 - MISRA C++
 - <https://www.misra.org.uk/Activities/MISRAC/tabid/171/Default.aspx>
 - JSF AV C++
 - <http://www.stroustrup.com/JSF-AV-rules.pdf>

Caractéristiques des logiciels embarqués

Des logiciels temps-réel et déterministes

- Un logiciel est dit « temps réel » s'il doit respecter un certain nombre de contraintes bornées sur des temps de réponse liés au traitement des stimuli en provenance de son environnement.
- Le non respect de ces contraintes entraîne une dégradation des performances (temps-réel « mou », soft real-time).
- Ou une défaillance complète du système pouvant mener à une catastrophe (temps-réel « dur », hard real-time).
- Le fonctionnement du logiciel doit être déterministe (en termes de résultats, de timings d'exécution, de comportement), c'est-à-dire prévisible et reproductible.

Caractéristiques des logiciels embarqués

Des logiciels temps-réel et déterministes

- La puissance brute du processeur est moins importante que la garantie que les temps de réponse seront tenus.
- Architecture mémoire maîtrisée (pas d'allocation dynamique du type malloc ou new).
- Multi-tâches.
 - Les logiciels embarqués sont souvent construits autour de noyaux temps réel qui offrent un support pour la programmation multi-tâches.
 - Des analyses d'ordonnabilité permettent d'étudier / prouver la faisabilité du système

Logiciels embarqués

Domaines d'application

- Très grande diversité des domaines d'application.
- Très grande diversité de la complexité des applications
 - Micro-contrôleurs 8 bits avec quelques kB de mémoire fonctionnant sans OS → processeurs ARM multi-core avec des GB de mémoire fonctionnant avec un OS et mettant en œuvre des techniques de virtualisation.

Sommaire

1. Contexte : les systèmes embarqués
2. Logiciels embarqués : définition et caractéristiques
3. Cibles matérielles : aperçu des familles de processeurs et SoC pour l'embarqué (ARM, LEON, ...)

Cibles matérielles

- Grande diversité des cibles matérielles :

**Micro-
contrôleurs**

DSP

**Processeurs
généralistes**

SoC

Cibles matérielles

Les micro-contrôleurs

- Applications de type commande / contrôle
- Faible consommation électrique
- Adaptés pour les applications temps réel
 - Temps de latence faibles
 - Support pour le multi-tâches (gestion des changements de contexte et des systèmes préemptifs)
 - Nombreuses sources d'interruption
- Mémoires (RAM, FLASH) et unités de gestion des périphériques (ports I/O, liens de communications, ADC / DAC, amplificateurs, filtres, etc.) intégrées (contrairement aux micro-processeurs « généralistes » → les microcontrôleurs sont désormais aussi des SoC...

Cibles matérielles

Les DSP (Digital Signal Processor)

- Architecture et jeu d'instructions dédiés aux calculs
- Optimisés pour les applications gérant en continu des flux de données plus que pour des applications de type commande contrôle
- Jeux d'instructions étendus permettant de réaliser plusieurs calculs parallèlement (architecture VLIW offrant plusieurs unités d'exécution parallèles)
- Implémentation de boucles sans overhead
- Mémoires spécialisées
- Gestions de flux de données élevés
- Adaptés pour les applications temps-réel

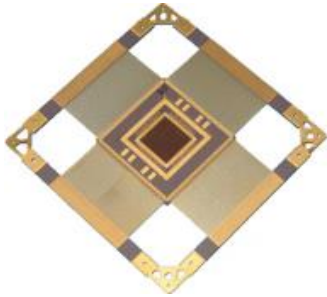
Cibles matérielles

Les SoC

- SoC = System On a Chip
 - Système complet embarqué sur une puce
 - Principe = synthétiser sur la même puce :
 - un ou plusieurs cœurs de processeur,
 - différents modules de gestion des entrées / sorties et des périphériques,
 - de la mémoire,
 - ou tout autre composant nécessaire à la réalisation de la fonction attendue.
 - Faire communiquer ces différentes entités via un bus interne standardisé AMBA
 - Miniaturisation et intégration
 - Des blocs de fonctions sont disponibles à la vente ou en libre téléchargement.
 - Ces blocs sont appelés IP (Intellectual Property).

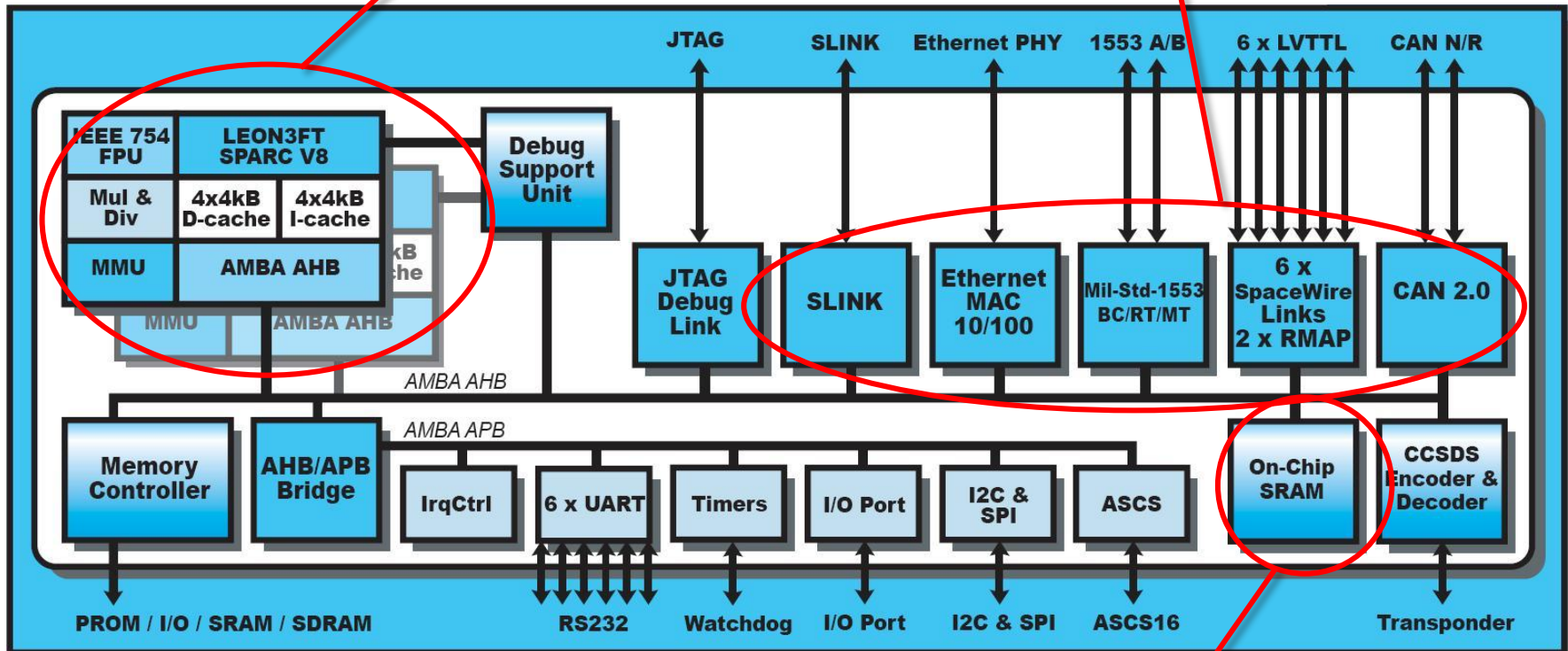
Cibles matérielles – Les SoC

Ex. : le processeur LEON3FT GR712RC



Deux cœurs de processeur (2 x CPU)

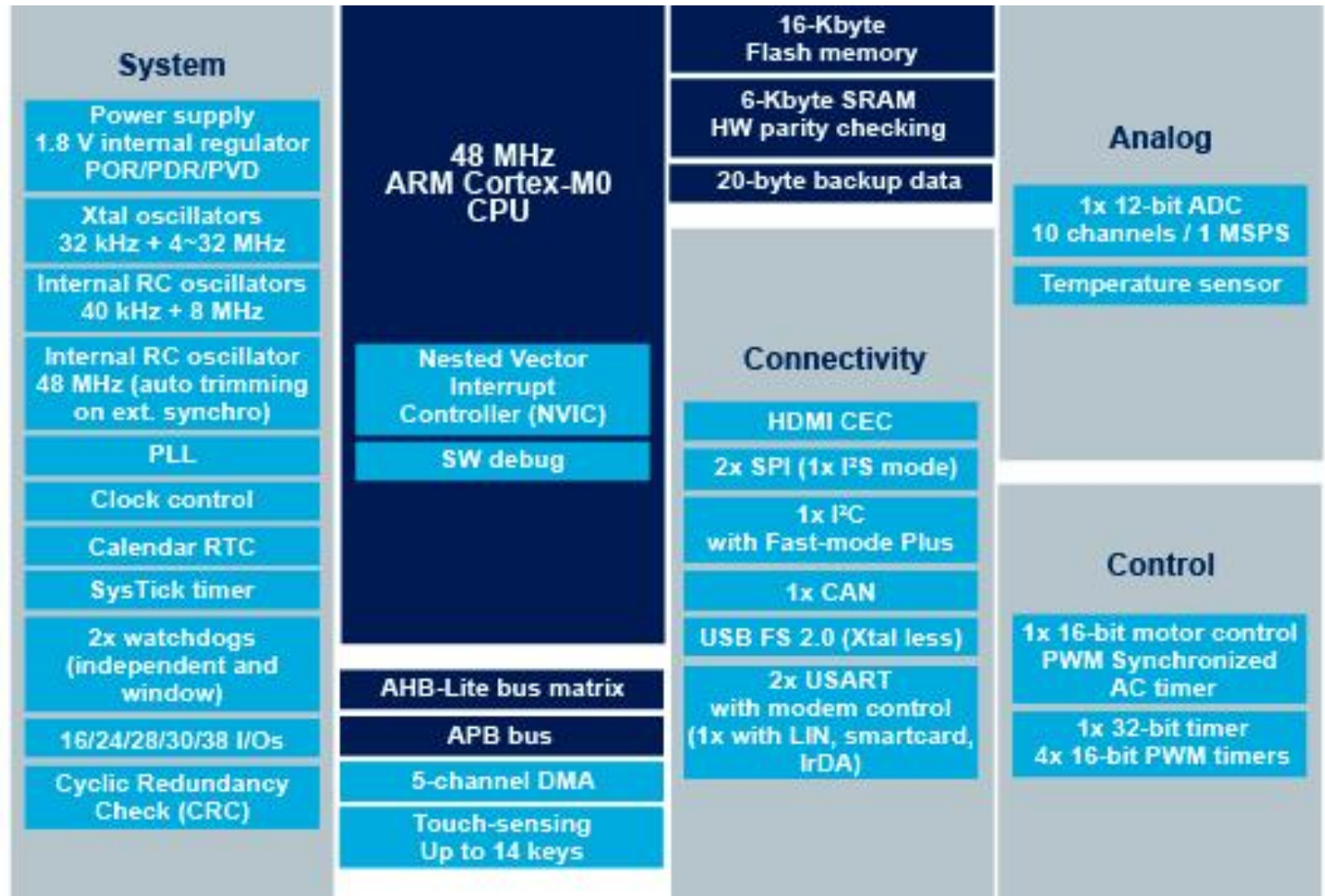
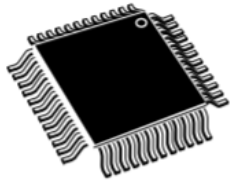
Des modules de gestion des entrées / sorties



De la mémoire
On-Chip

Cibles matérielles – Les SoC

Ex. : le processeur STM32F042x4



Cibles matérielles

L'éco-système ARM

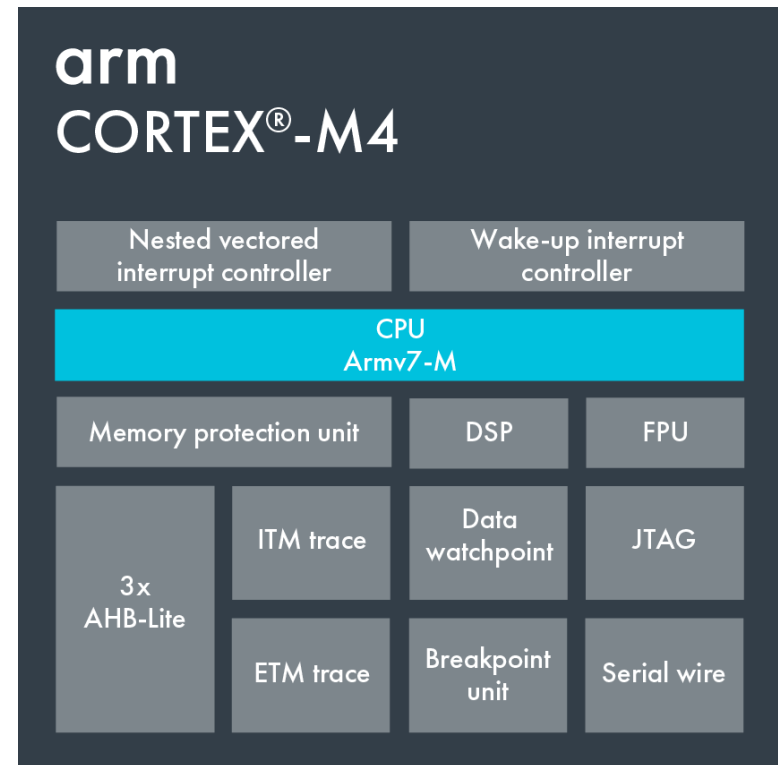
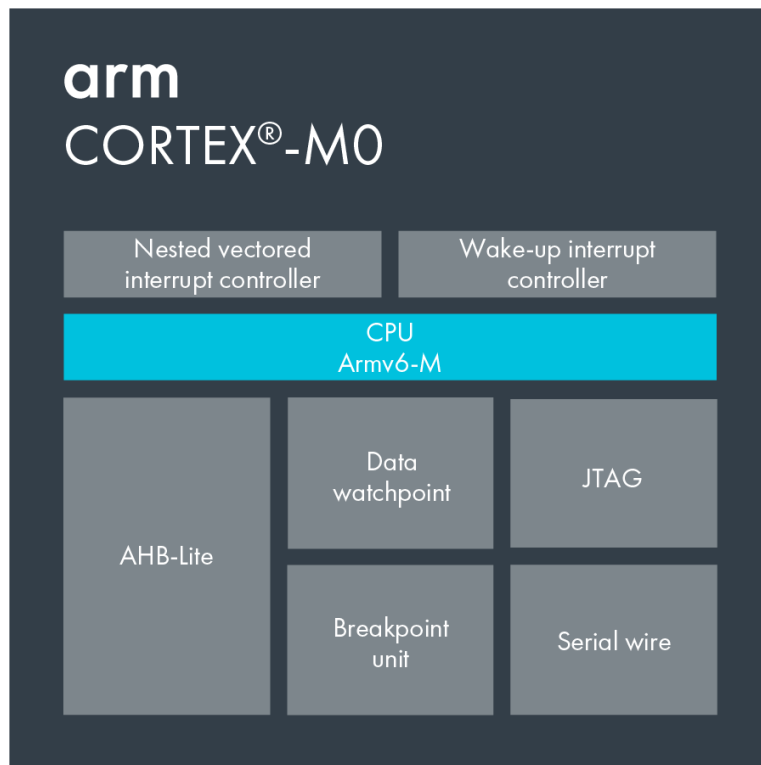
- ARM = Advanced RISC Machines
- Famille de processeur désormais la plus utilisée dans le monde des processeurs embarquées et des téléphones mobile
- Propriété intellectuelle des processeurs ARM = société britannique Arm Holdings
- La société Arm ne fabrique pas elle-même les processeurs physiques : elle conçoit la technologie ARM et vend des licences d'utilisation (IP).
- Principaux fabricants des puces ARM : STMicroelectronics, Microsemi, NXP Semiconductors, Samsung, Texas Instruments, ...

Cibles matérielles

L'éco-système ARM

■ ARM Cortex-M

- Faible puissance / Faible consommation
- Applications de type commande-contrôle (microcontrôleurs) et applications temps-réel



Crédit : <https://developer.arm.com/products/processors/cortex-m>

Cibles matérielles

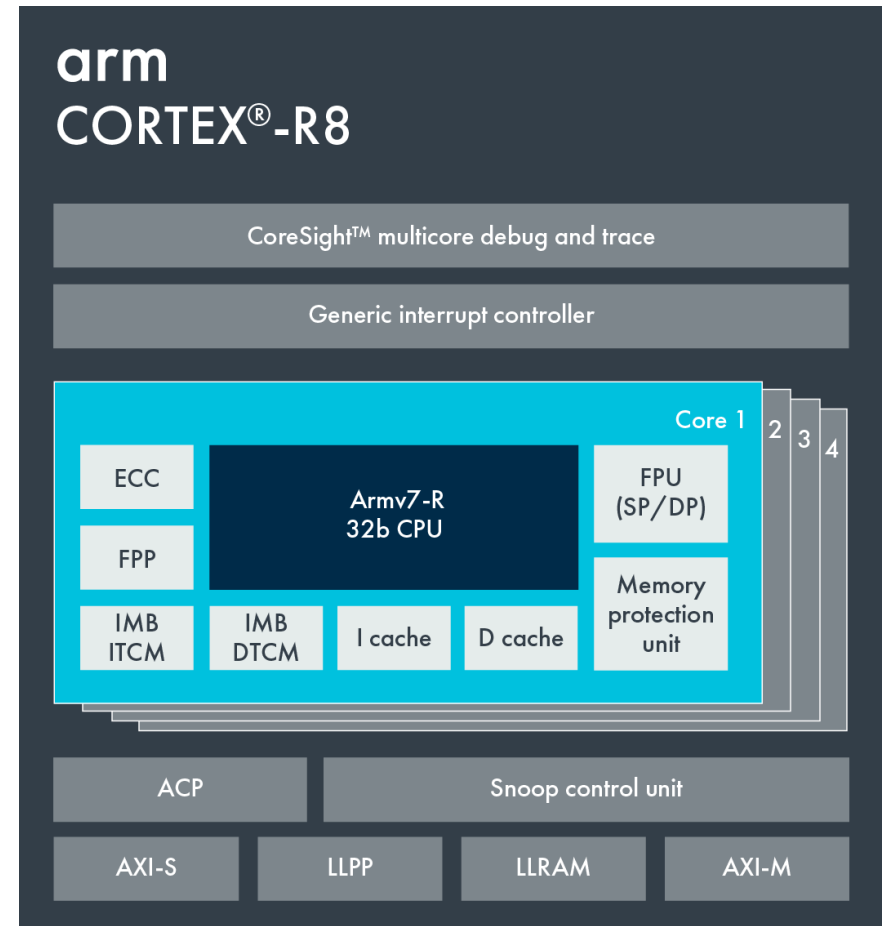
L'éco-système ARM – la famille Cortex-M

Arm Core	Cortex M0 ^[2]	Cortex M0+ ^[3]	Cortex M1 ^[4]	Cortex M3 ^[5]	Cortex M4 ^[6]	Cortex M7 ^[7]	Cortex M23 ^[8]	Cortex M33 ^[12]	Cortex M35P
ARM architecture	ARMv6-M ^[9]	ARMv6-M ^[9]	ARMv6-M ^[9]	ARMv7-M ^[10]	ARMv7E-M ^[10]	ARMv7E-M ^[10]	ARMv8-M Baseline ^[15]	ARMv8-M Mainline ^[15]	ARMv8-M Mainline ^[15]
Computer architecture	Von Neuman	Von Neumann	Von Neumann	Harvard	Harvard	Harvard	Von Neumann	Harvard	Harvard
Instruction pipeline	3 stages	2 stages	3 stages	3 stages	3 stages	6 stages	2 stages	3 stages	3 stages
Thumb-1 instructions	Most	Most	Most	Entire	Entire	Entire	Most	Entire	Entire
Thumb-2 instructions	Some	Some	Some	Entire	Entire	Entire	Some	Entire	Entire
Multiply instructions 32x32 = 32-bit result	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multiply instructions 32x32 = 64-bit result	No	No	No	Yes	Yes	Yes	No	Yes	Yes
Divide instructions 32/32 = 32-bit quotient	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Saturated instructions	No	No	No	Some	Yes	Yes	No	Yes	Yes
DSP instructions	No	No	No	No	Yes	Yes	No	Optional	Optional
Single-Precision (SP) Floating-point instructions	No	No	No	No	Optional	Optional	No	Optional	Optional
Double-Precision (DP) Floating-point instructions	No	No	No	No	No	Optional	No	No	No
TrustZone instructions	No	No	No	No	No	No	Optional	Optional	Optional
Co-processor instructions	No	No	No	No	No	No	No	Optional	Optional
Interrupt latency (if zero-wait state RAM)	16 cycles	15 cycles	23 for NMI 26 for IRQ	12 cycles	12 cycles	12 cycles	15 no security ext 27 security ext	TBD	TBD

Cibles matérielles

L'éco-système ARM

- ARM Cortex-R
 - Performances élevées, contraintes temps réel fortes
 - Déterminisme, tolérance aux erreurs, fiabilité, sûreté de fonctionnement => applications médicales, automobile, ...
 - FPU (Floating Point Unit), instructions DSP
 - MPU (Memory Protection Unit)
 - ECC (Error Correcting Code)
 - Low Latency Peripheral Port (LLPP)
 - Jusqu'à 4 cœurs



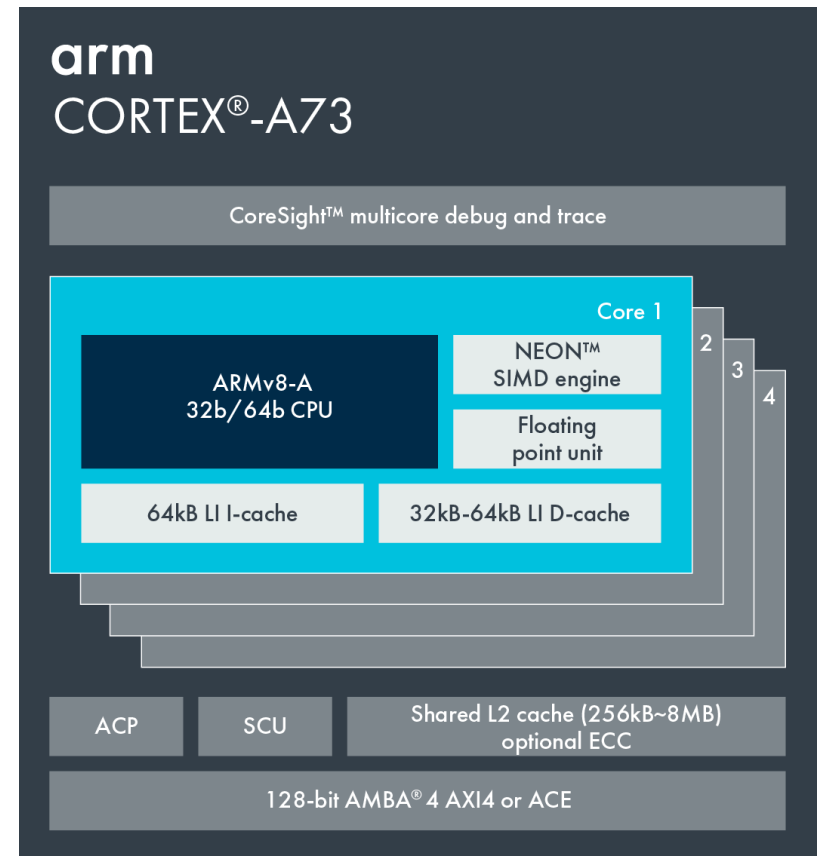
Crédit : <https://developer.arm.com/products/processors/cortex-r/cortex-r8>

Cibles matérielles

L'éco-système ARM

■ ARM Cortex-A

- Multi-core avec support des architectures SMP
- Support pour la virtualisation
- Tâches de calcul complexes
- Prise en charge d'OS complexes (Android...)
- Smartphones, tablettes, etc.
- Extension d'architecture SIMD (single instruction multiple data) : accélération pour les applications de traitements d'image et du signal
- TrustZone : technologie de chiffrement et sécurisation des échanges avec des plates-formes tierce



<https://developer.arm.com/products/processors/cortex-a/cortex-a73>

Cibles matérielles

Les processeurs LEON

- Famille de processeurs 32 bits (big endian) avec une base architecturale commune :
 - Sparc V8 (<https://www.gaisler.com/doc/sparcv8.pdf>)
- LEON-FT : design tolérant aux fautes et insensible aux SEUs → processeurs les plus utilisés dans le domaine du spatial
 - LEON2-FT : licence ESA
 - LEON3-FT, LEON4-FT : licence Cobham/ Gaisler ; multi-processeurs :
 - voir <https://www.gaisler.com/index.php/products/processors>

Cibles matérielles

Les processeurs LEON

- Disponibles sous forme d'un modèle VHDL synthétisable
 - Cœur de processeur configurable
 - Utilisable pour réaliser des system-on-a-chip (SoC) intégrant d'autres IP Core (ex. : codec SpaceWire) autour du bus AMBA 2.0
 - Implémentation dans des FPGA ou des ASIC
- Plusieurs ASIC disponibles sur le marché :
 - AT697 (LEON2-FT single core)
 - UT699 (LEON3-FT single core)
 - GR700 (LEON3-FT single core)
 - GR712RC (LEON3-FT dual-core)
 - GR740 (LEON4-FT quadri-core)
 - GR716 (LEON3-FT microcontrôleur avec jeu d'instructions 16 bits pour un code plus compact)