

# ARCHITECTURE ARM

---

## 1. Présentation

- ARM (Advanced RISC Machines) est une famille d'architectures de processeurs basées sur l'architecture RISC.
- Bien connu pour son efficacité énergétique; d'où largement utilisé dans les appareils mobiles, tels que les téléphones intelligents et les tablettes.
- L'entreprise fait la conception des processeurs ARM; ne fabrique pas, mais vend la licence à des partenaires de semi-conducteurs qui ajoutent leur propre propriété intellectuelle (IP) à la propriété intellectuelle d'ARM.

L'architecture cortex est divisée en trois familles :

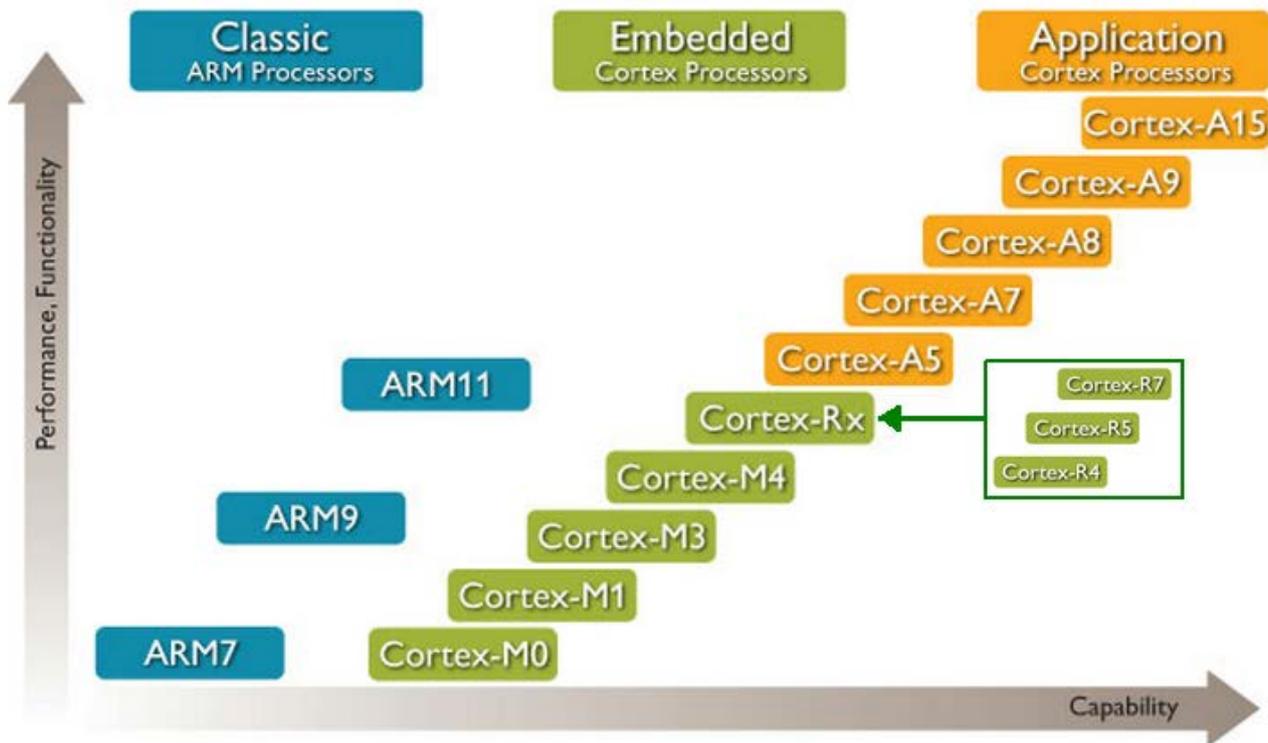
Cortex A (Application) : Microprocesseurs à haute performances, capables de supporter des systèmes d'exploitation. (Application : smartphones, digital TV, smart-books...

Cortex R (Real Time) : Hautes performances pour les applications temps réel, grande fiabilité.

Applications : Système de freinage automobile, les groupes motopropulseurs, etc...

Cortex M (microcontrôleur) : Des solutions à faible coût pour des applications déterministes

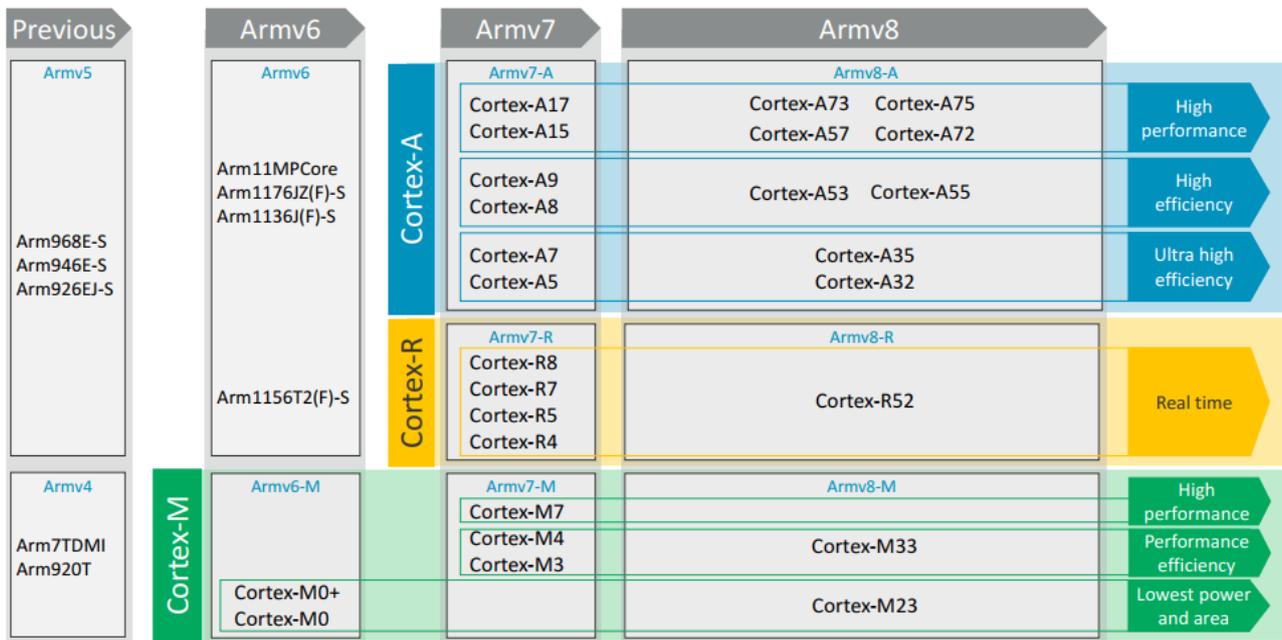
Applications : sonorisation, électroniques de carrosserie automobile et les airbags.



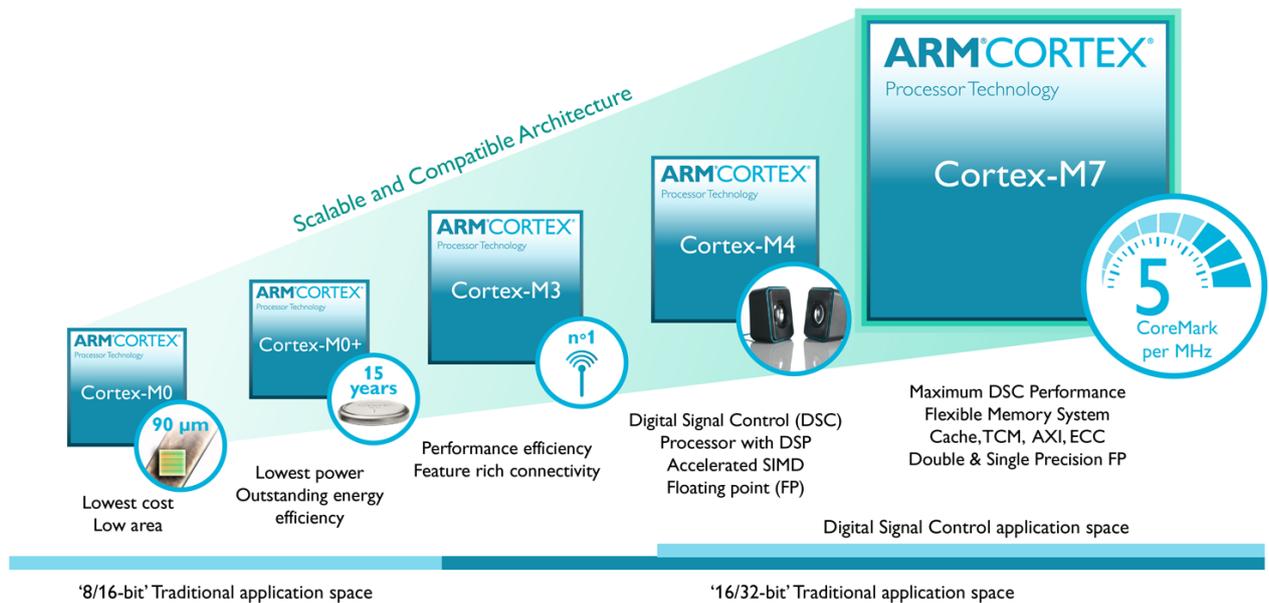
Processeur ARM vs Architecture ARM

Architecture ARM

- Décrit les détails du jeu d'instructions, du modèle du programmeur, le modèle d'exception et le mappage de la carte mémoire.
- Documenté dans le Manuel de Référence de l'Architecture.
- Processeur ARM
- Développé en utilisant l'une des architectures ARM.
- Plus de détails de mise en œuvre.



## 2. Architecture Cortex M



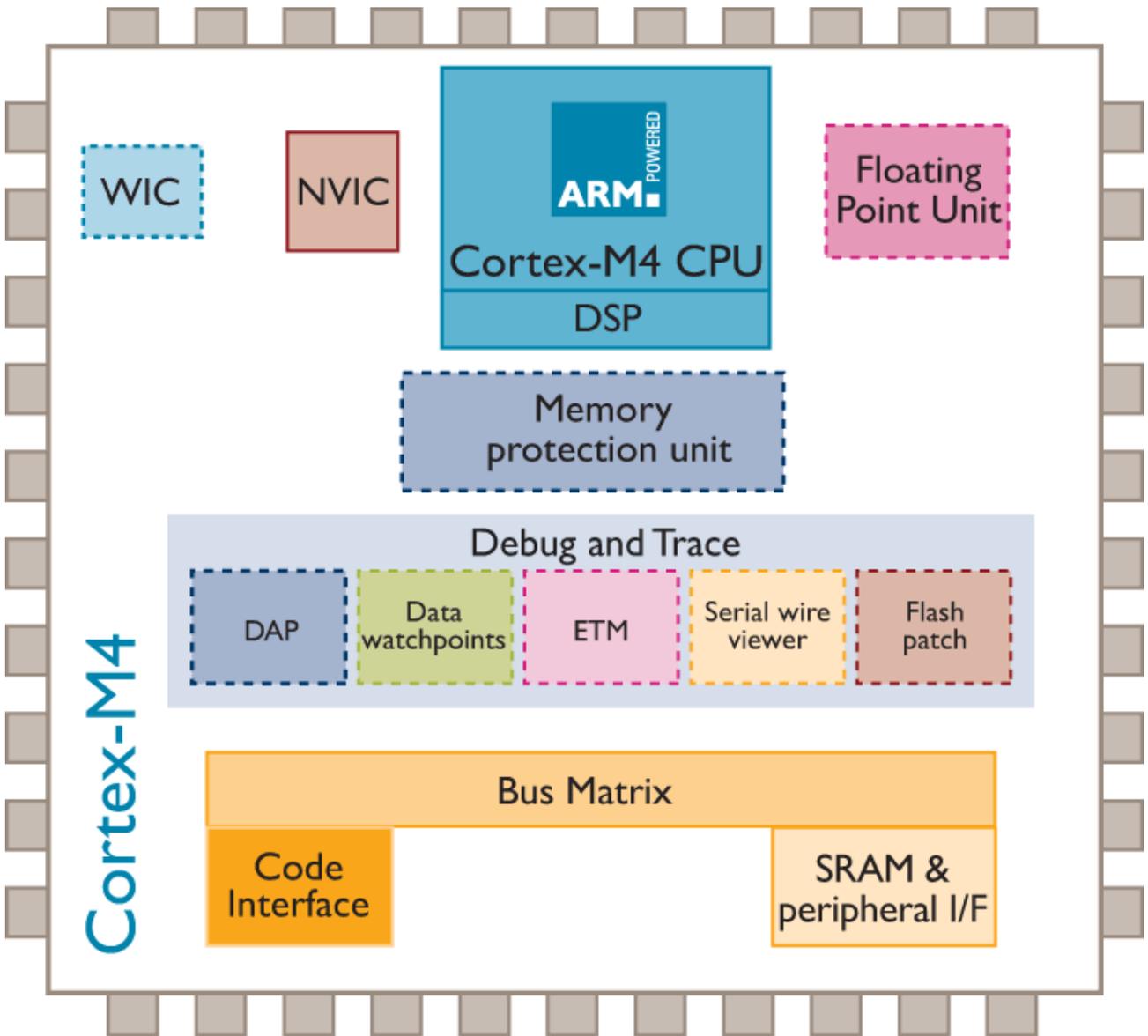
SIDM : Signe Instruction Multiple Data

TCM : Tightly Coupled Memory (Mémoire à Couplage Etroit) : ITCM (64 bits), DTCM (2 x 32 bits).

AXI : Advanced External Interface : connexion des mémoires externes avec un bus de 64 bits.

ECC : Error Code Correction.

## Cortex-M4



WIC : Wakeup Interrupt Controller (optionnel)

NVIC : Nested Vectored Interrupt Controller

ETM : Embedded Trace Macrocell (optionnel)

DAP : Debug Access Port (optionnel)

## 2.1. Les unités exécutives

**CM4CORE** : il s'agit du cœur en lui-même. Dans cette unité on trouve les différents registres, tous les mécanismes de lecture/écriture des instructions et des données ainsi que l'unité arithmétique et logique (UAL) pour l'exécution des différentes instructions.

### Nested Vectored Interrupt Controller

Le NVIC est un contrôleur d'interruption intégré ayant une faible latence de traitement d'interruption. L'imbrication des interruptions permet de fournir 256 niveaux de priorité.

### System Control Block (SCB)

Une partie du processeur qui est fusionné dans l'unité NVIC est le SCB. Il fournit des informations de mise en œuvre et le contrôle du système, y compris la configuration, le contrôle et la notification des exceptions.

### System timer

La Timer système, sysTick, est un compteur à rebours 24-bits. Ce timer est utilisé par le système d'exploitation temps réel (RTOS).

### Memory Protection Unit

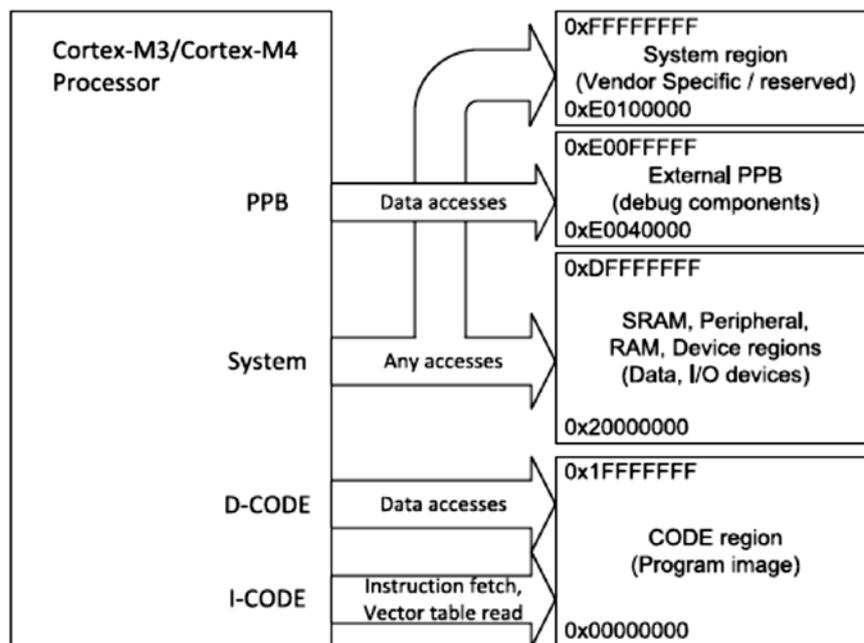
Le MPU est un dispositif programmable qui surveille les transactions de bus et doit être configuré par un logiciel, typiquement un système d'exploitation intégré. Si un MPU est inclus, les applications peuvent diviser l'espace mémoire dans un certain nombre de régions et de définir les autorisations d'accès pour chacun d'eux. Quand une règle d'accès est violée, une exception de défaut est générée.

### Le bus Matrix

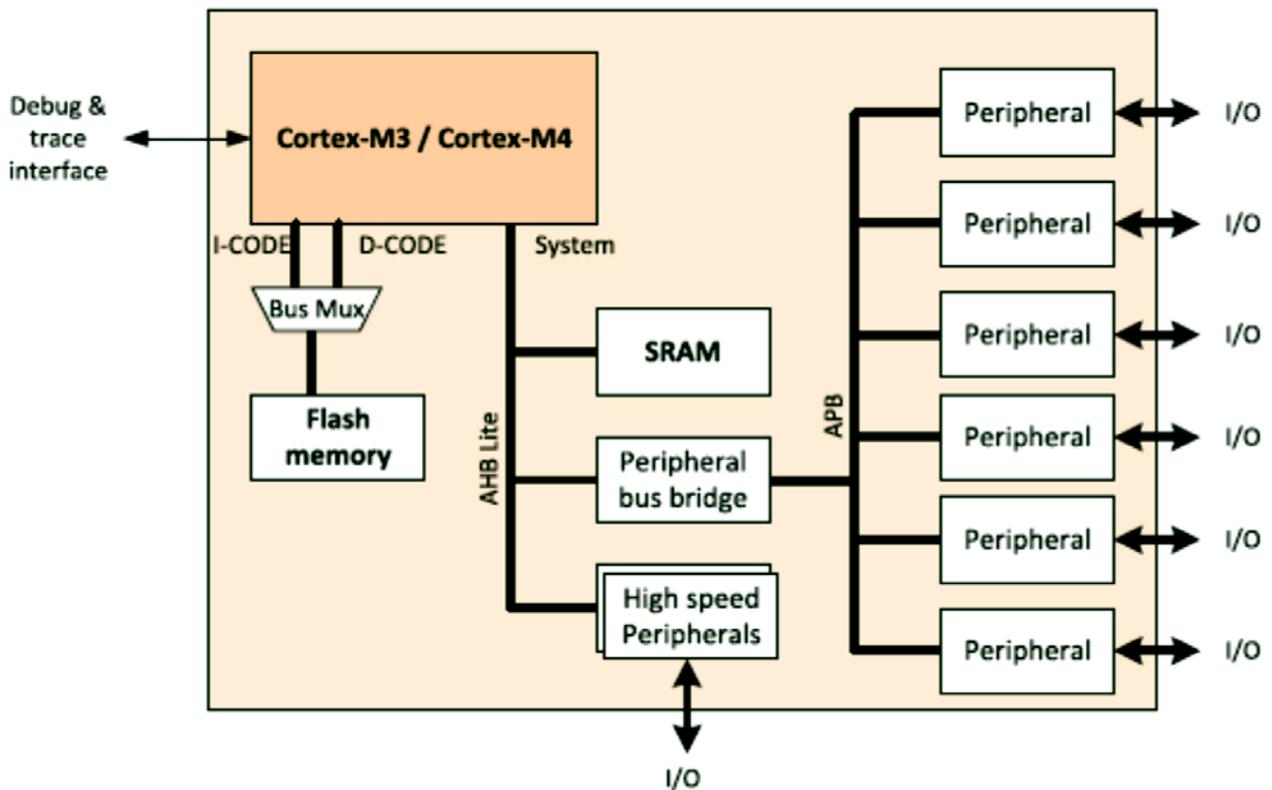
Cette unité est un multiplexeur intelligent. Le processeur contient quatre bus externes basés sur le protocole AMBA (Advanced Microcontroller Bus Architecture).

- Le bus ICode : principalement pour la mémoire de programme. Ce bus de 32 bits permet la recherche des instructions et des vecteurs à partir de l'adresse 0x00000000 à 0xFFFFFFFF. Basé sur le protocole AMBA 3.0 AHB Lite (AHB : Advanced Hi-performance Bus).
- Le bus Dcode : principalement pour la mémoire de programme. Il permet l'accès aux données à partir de l'adresse 0x00000000 à 0xFFFFFFFF. Basé sur le protocole AMBA 3.0 AHB Lite.

- Le bus système : principalement pour la RAM et les périphériques. Il permet l'accès à la totalité de l'espace compris entre 0x20000000 et 0xFFFFFFFF à l'exception de la région PPB. Basé sur le protocole AMBA 3.0 AHB Lite.
- Le bus PPB pour Private Peripheral Bus. Bus de débogage des composants privés au niveau d'adresse 0xE0040000 à 0xE00FFFFFF. Basé sur le protocole AMBA 3.0 APB (Advanced Peripheral Bus) moins performant que l'AHB Lite.



Les périphériques d'interface sont généralement basés sur le protocole APB. Cependant, le protocole AHB peut être utilisé pour les interfaces de haute performance.



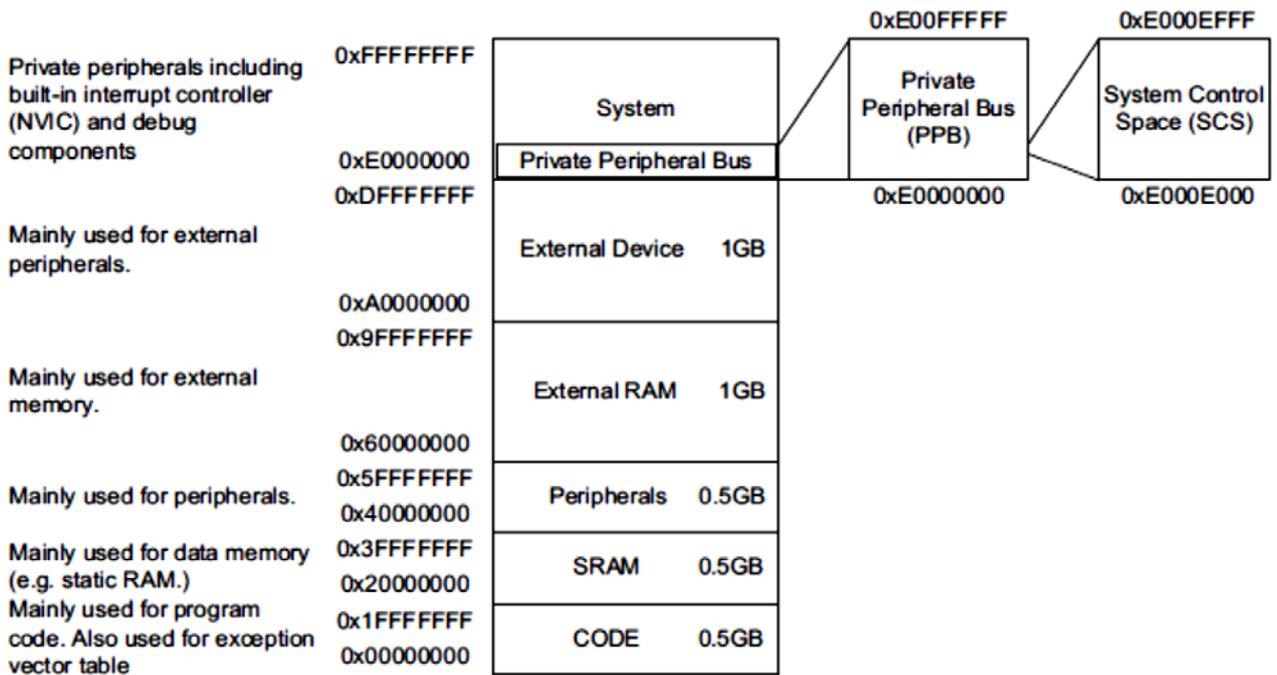
## 2.2. Mappage mémoire

L'espace d'adressage de 4 Go de Cortex-M est divisé en un certain nombre de régions de mémoire. Le partitionnement est basé sur des usages typiques de sorte que différentes zones sont conçus pour être utilisés principalement pour :

- Accès au code du programme (région de CODE)
- Accès de données (région de SRAM)
- Périphériques (région périphérique)
- contrôle et déboguer les composants internes du processeur (par exemple, le bus PPB)

L'architecture permet également une grande flexibilité pour permettre à des régions de mémoire d'être utilisé pour d'autres fins. Par exemple, les programmes peuvent être exécutés à partir du CODE, ainsi que la région de SRAM, et un microcontrôleur peut également intégrer des blocs SRAM dans la région de CODE.

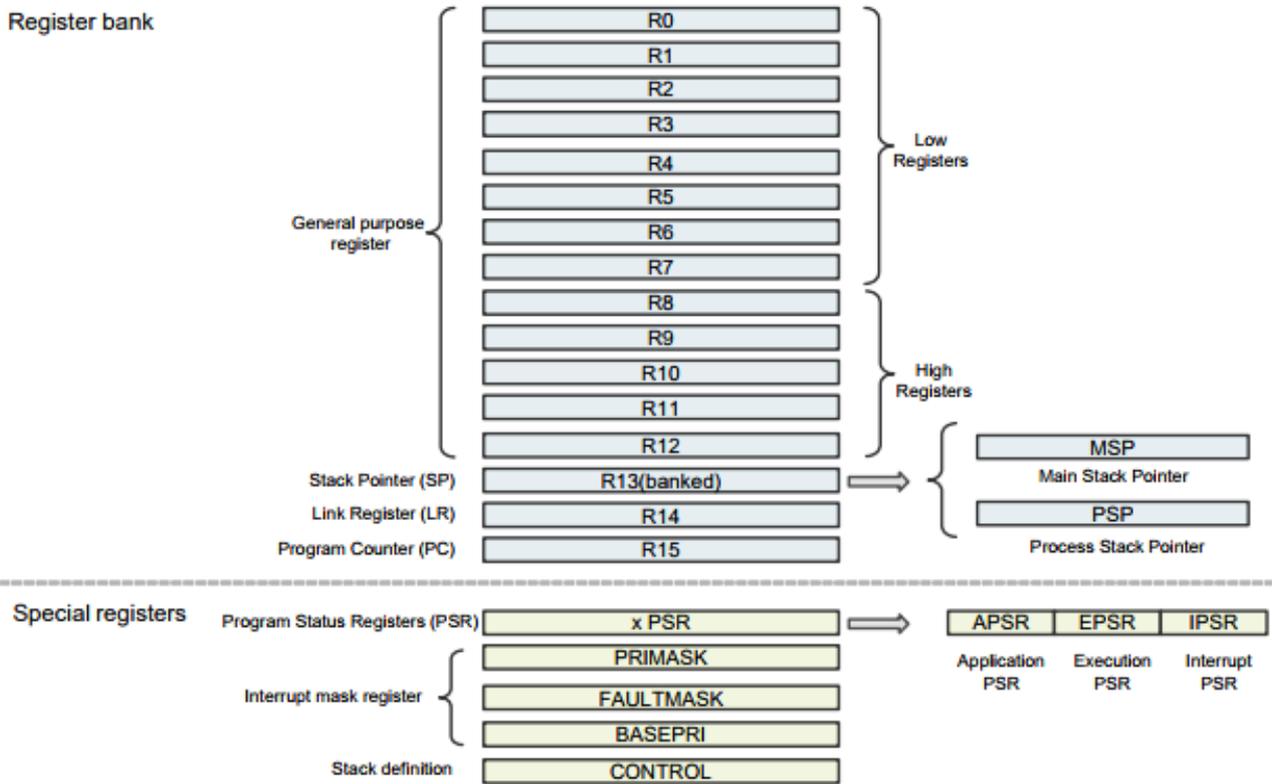
Dans la pratique, de nombreux microcontrôleurs utilisent seulement une petite partie de chaque région pour le programme flash, SRAM, et les périphériques. Certaines régions ne sont pas utilisées.



### 2.3. Les Registres

Similaire à la plupart des microprocesseurs, l'architecture Cortex-M dispose d'un certain nombre de registres à l'intérieur du core du processeur pour le traitement et le contrôle des données. Cortex-M est une architecture de type **Load-Store** ; cela signifie s'il y a des données en mémoire à traiter, il doit transférer les données de la mémoire vers les registres, les traitées à l'intérieur du processeur, puis faire le transfert vers la mémoire si nécessaire. Ayant un nombre suffisant de registres dans la banque de registres, cet arrangement est facile à utiliser, et permet au compilateur C de générer un code de programme efficace.

La banque de registre cortex-M4 comporte 16 registres. Dont treize registres à usage général.



Les Registres Bas (R0 à R7) sont des registres à usage général accessibles par toutes les instructions, alors que les registres R8 à R12 sont accessibles par les instructions codées sur 32 bits, et un nombre limité d'instructions codées sur 16 bits.

R13 est un pointeur de pile (Stack Pointer). Physiquement il y a deux pointeurs de pile. MSP ou SP\_Main et ou PSP SP\_Process. MSP est le pointeur par défaut, il est utilisé en mode Handler (Interruption ou système d'exploitation). En dehors du mode handler, il est possible d'utiliser SP\_Process.

R14 appelé aussi (LR : Link Register), chargé par l'adresse de retour lors de l'appel d'une fonction ou d'une procédure.

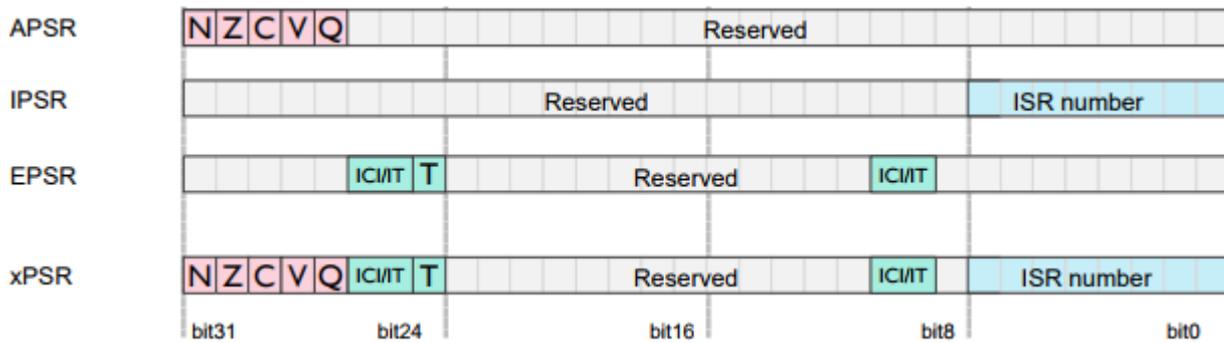
R15 joue le rôle de compteur de programme.

A part ces registres, il y a d'autres registres spéciaux; on peut citer ici le registre xPSP. Ce registre est composé de trois registres (APSP : Application Program Status Register ; EPSP : Execution Program Status Register ; IPSP : Interrupt Program Status Register).

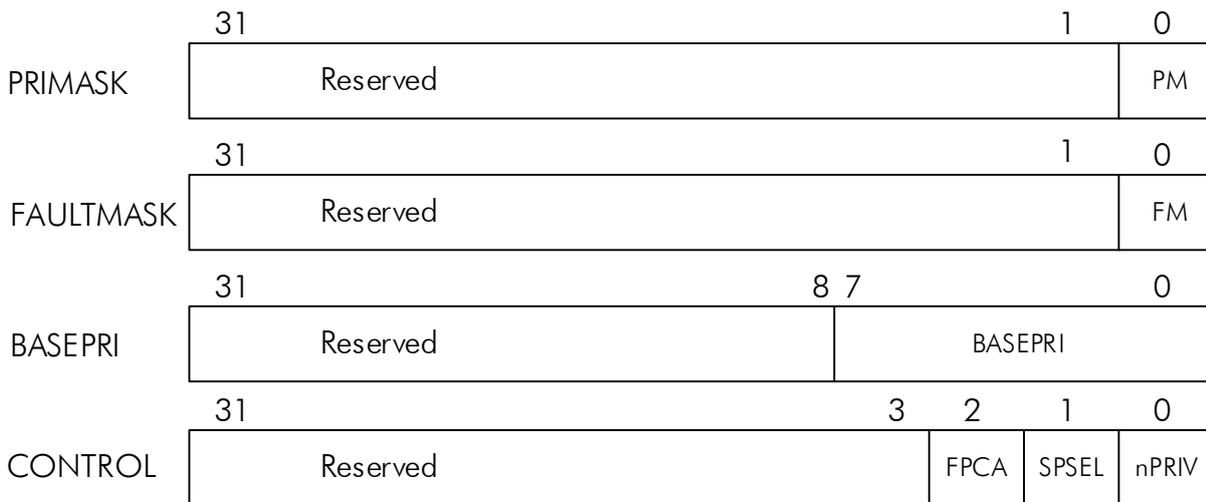
APSP : contient les indicateurs : de retenue C, de signe N, du résultat nul Z...

EPSP : indique si le microprocesseur exécute une instruction If-Then ou une instruction Load/Store Multiple Register.

IPSP : contient le numéro d'exception ou d'interruption.



### Registres PRIMASK, FAULTMASK, BASEPRI et CONTROL



Registre PRIMASK : lorsque le bit PM = 1, toutes les interruptions sont masquées, sauf NMI (Non-Maskable Interrupt) et l'exception HardFault.

Registre FAULTMASK : lorsque le bit FM = 1, l'exception HardFault est masquée.

Registre BASEPRI : Fixe le niveau de priorité (voir chp. NVIC).

Registre CONTROL :

Bit FPCA : Lorsque le paramètre flottant est implémenté, ce bit indique si le contexte à virgule flottante est actuellement actif.

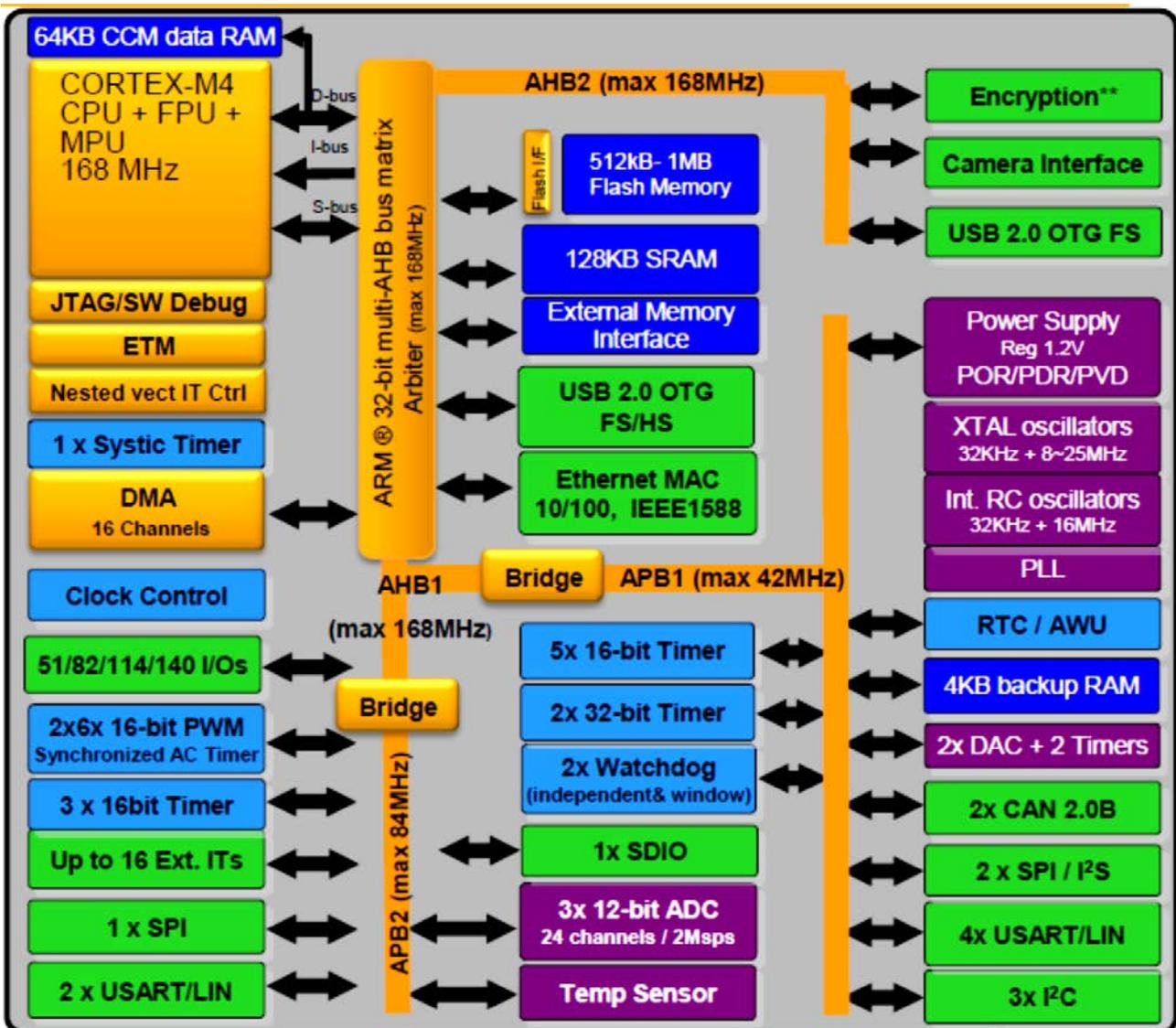
Bit SPSEL : définit le pointeur de pile actif (0 : MSP , 1 : PSP).

Bit nPRIV : définit le niveau d'exécution des thread (0 : mode privilégié, 1 : non privilégié).

### 3. STM32F4xx

#### 3.1. Présentation

Le circuit STM32F4xx est basé sur une architecture ARM® Cortex™-M4 RISC 32 bits, fonctionnant à une fréquence maximale de 168 MHz. Il intègre une mémoire flash de 1 Mo, une mémoire système SRAM de 128 Ko, une mémoire SRAM de sauvegarde de 4 Ko, une RAM couplée au core de 64Ko et une vaste gamme de périphériques d'entrées-sorties. L'intégration efficace des différents composants exige l'emploi d'un bus système à haute performance et ayant une grande bande passante pour effectuer des opérations en parallèle.



### 3.2. Matrice d'interconnexion

L'intégration efficace des différents composants exige l'emploi d'un bus système à haute performance et ayant une grande bande passante pour effectuer des opérations en parallèle

Pour résoudre le problème de la bande passante, la société ARM a proposé le « Multi-layer AHB bus », sous forme d'une matrice d'interconnexion basée sur le protocole AHB. Cette matrice permet à plusieurs composants maîtres d'être interconnectés avec plusieurs esclaves.

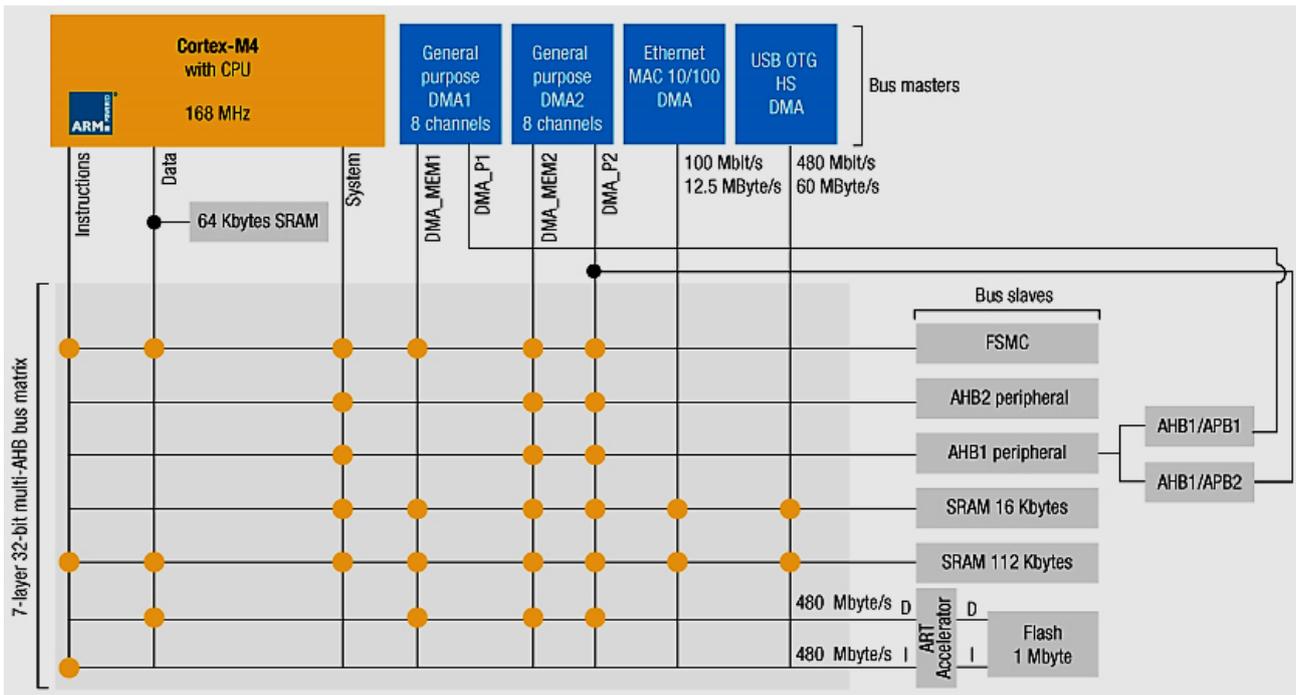
Le microcontrôleur STM32F4xx comporte huit bus maîtres et 7 esclaves :

#### Bus maîtres :

- Cortex<sup>®</sup>-M4 ICode, DCode et System bus
- DMA1 memory bus
- DMA2 memory bus
- DMA2 peripheral bus
- Ethernet DMA bus
- USB OTG HS DMA bus

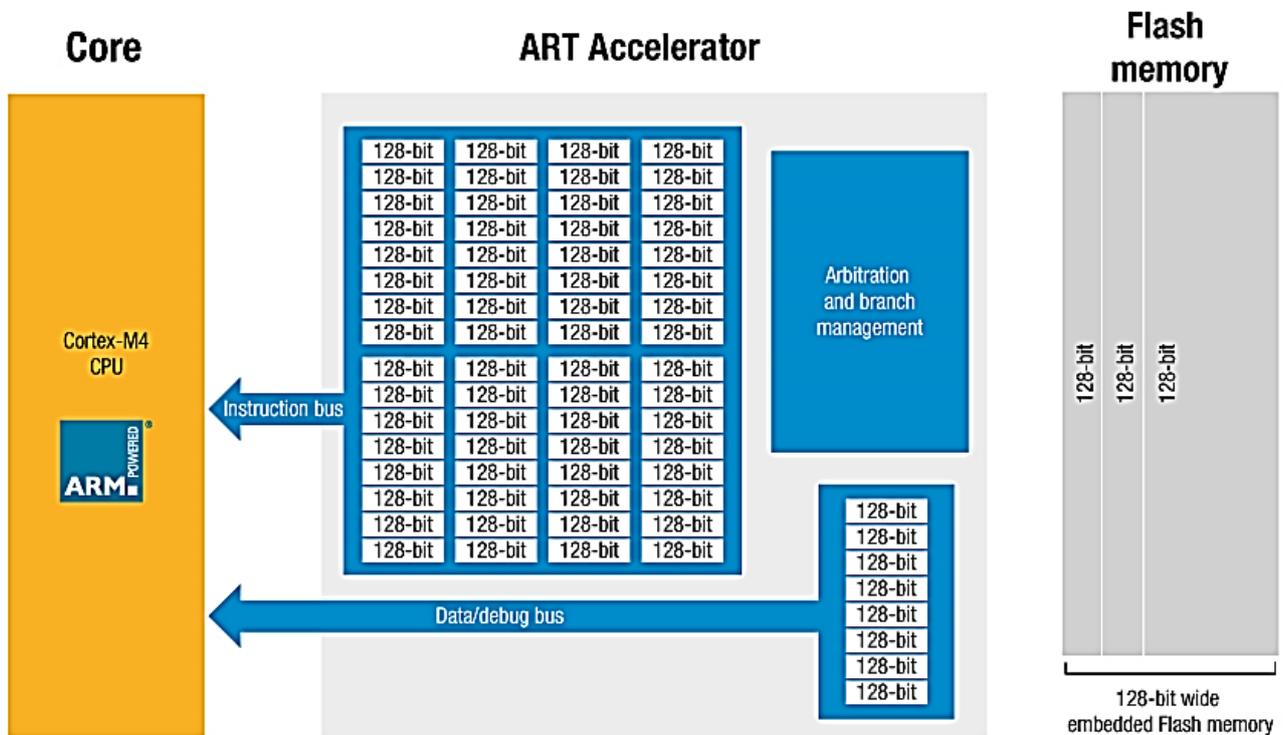
#### Bus esclaves :

- Internal Flash memory ICode bus
- Internal Flash memory DCode bus
- Main internal SRAM1 (112 KB)
- Auxiliary internal SRAM2 (16 KB)
- AHB1 peripherals including AHB to APB bridges and APB peripherals
- AHB2 peripherals
- FSMC



La matrice de bus permet l'interconnexion d'un maître à un esclave, ce qui permet un accès simultané et le fonctionnement efficace même lorsque plusieurs périphériques à haute vitesse fonctionnent simultanément.

### 3.3. Adaptive Real-Time (ART) Accelerator



### 3.4. Architecture du microcontrôleur STM32F4xx

L'architecture du microcontrôleur STM32F4xx est donnée par la figure suivante :

