

Guide de référence sur les processeurs ARM

ARM, qui signifie « Advanced RISC Machines » (à l'origine « Acorn RISC Machine ») est une famille d'architectures RISC (« Reduced Instruction Set Computer », ordinateur à jeu d'instructions réduit) pour microcontrôleurs et microprocesseurs. Ces processeurs sont développés par la société britannique Arm Holdings, plc (anciennement Advanced RISC Machines Ltd, avant cela Acorn) et ont été commercialisés pour la première fois en 1985.

Qu'est-ce qu'un RISC ?

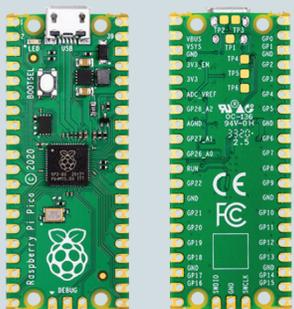
Un ordinateur à jeu d'instructions réduit (RISC) est une architecture informatique conçue pour simplifier les instructions individuelles données à l'ordinateur pour accomplir des tâches. L'objectif est de compenser la nécessité de traiter davantage d'instructions en augmentant la vitesse de chaque instruction, notamment en mettant en œuvre un pipeline d'instructions, ce qui peut être plus simple à réaliser grâce à des instructions plus simples. Le concept opérationnel clé est que chaque instruction n'exécute qu'une seule fonction (par exemple, copier une valeur de la mémoire vers un registre).

Les générations d'architecture ARMv1 (1985) à ARMv5 (1997) avec 32 bits/32 bits ont été implémentées dans les versions Core ARM1 à ARM10. Ces innovations ont donné naissance à des modèles de cœur qui définissent l'industrie, comme l'ARM8 (qui permet d'utiliser le célèbre microprocesseur StrongARM) et les ARM7/9 (processeur XScale).

Avec le passage à une architecture 32 bits/64 bits et l'architecture d'instructions Thumb v2, la génération ARMv6 (2002) a donné naissance au cœur ARM11 (largement utilisé dans les smartphones), mais aussi à l'approche ARM Cortex : des versions de cœurs spécialisées pour différents domaines d'application. Toutes les architectures ARM les plus récentes sont organisées en trois profils : Cortex-M, Cortex-R et Cortex-A.

À chaque occasion son Cortex

Cortex-M : pour des applications de microcontrôleur.
 Cortex-R : pour des applications en temps réel.
 Cortex-A : pour des applications qui utilisent des intergiciels ou des systèmes d'exploitation.



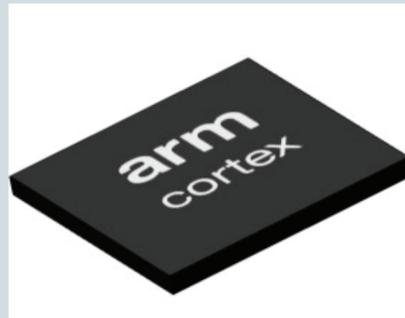
Le Cortex-M (« microcontrôleur »)

- ARMv6-M, ARMv7-M et ARMv8-M 32 bits
- Intégré dans des microcontrôleurs, des ASIC, des FPGA et des SoC
- Systèmes intégrés, IoT, nœuds de capteurs, tâches automobiles simples
- Jeu d'instructions Thumb

M0	M0+	M1	M3	M4	M7
ARMv6-M	ARMv6-M	ARMv6-M	ARMv7-M	ARMv7E-M +DSP	ARMv7E-M +FPU
nRF51	Raspberry Pi Pico	FPGA Flash ProASIC3	LPC18xx	STM32G4	SMART

M23	M33	M35	M52	M55	M85
ARMv8-M + TrustZone	ARMv8-M	ARMv8-M	ARMv8.1-M	ARMv8.1-M	ARMv8.1-M
SAM	EFM32	ST33K	Appareils IoT	Apollo 510	Carte de visualisation

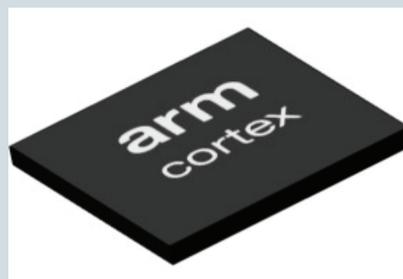
Le Cortex-R (« en temps réel »)



- ARMv7-R et ARMv8-R
- Applications hautes performances en temps réel
- Applications critiques pour la sécurité, notamment les dispositifs médicaux, les systèmes de contrôle industriel et les systèmes instrumentés de sécurité
- Jeu d'instructions Thumb et Thumb2

R4(F)	R5(F)	R7(F)	R8(F)	R52(F)	R52+(F)	R82(F)
			ARMv7-R	ARMv8-R	ARMv8-R	
RM44	TMS570	Double cœur	Quadruple cœur	RZ/N2L	SR6P7C4	Solutions de stockage

Le Cortex-A (« Application »)



- 32 bits (ARMv7-A) et 64 bits (ARMv8-A)
- Processeurs d'application avec prise en charge du système d'exploitation et des applications tierces
- Smartphones et serveurs
- Jeu d'instructions Thumb et Thumb2

A5 / A7 / A8 / A9 / A12 / A15 / A17	A32 / A34 / A35 / A53 / A57 / A72 / A73	A55 / A65 / A75 / A76 / A77 / A78	A510, A710 et A715	A520 et A720
ARMv7-A	ARMv8-A	ARMv8.2-A	ARMv9-A	ARMv9.2-A
PC industriels	AM65xx	SHARC	Google Tensor	

Jeu d'instructions

Thumb : Jeu d'instructions compressé : codage compact de 16 bits pour un sous-ensemble du jeu d'instructions ARM. Gain d'espace de stockage grâce à certains opérandes d'instructions implicites et au nombre limité de possibilités par rapport au jeu d'instructions ARM.

Thumb-2 : Jeu d'instructions à longueur variable, étend le jeu d'instructions limité à 16 bits de Thumb avec des instructions 32 bits supplémentaires.

Jazelle : Le Jazelle DBX (Direct Bytecode eXecution) est une technique qui permet d'exécuter directement le bytecode Java.

VFP : La technologie Vector Floating Point (« virgule flottante vectorielle ») est une extension de coprocesseur d'unité à virgule flottante (UVF).

Neon : Extension SIMD avancée, un jeu d'instructions SIMD combiné de 64 et 128 bits qui fournit une accélération standardisée pour les applications de traitement des médias et du signal.

Helium : Cette extension vectorielle de profil M (MVE) ajoute environ 150 instructions scalaires et vectorielles.

Pour aller plus loin

Architecture ARM : <https://www.arm.com/architecture>

Histoire de la société ARM : <https://newsroom.arm.com/blog/arm-official-history>