

## Guida di riferimento ai processori ARM

ARM, o Advanced RISC Machines (originariamente Acorn RISC Machine), è una famiglia di architetture RISC (Reduced Instruction Set Computer) per microcontrollori e microprocessori. Tali prodotti sono stati sviluppati dalla società britannica Arm Holdings, plc, (in precedenza Advanced RISC Machines Ltd, prima ancora Acorn) e introdotti sul mercato per la prima volta nel 1985.

### Che cosa è RISC?

Un computer con set di istruzioni ridotto (RISC) è un'architettura per computer progettata per semplificare le singole istruzioni fornite al computer per svolgere i compiti. L'obiettivo è quello di compensare la necessità di elaborare più istruzioni aumentando la velocità di ciascuna istruzione, in particolare tramite una sequenza di istruzioni, che può essere più semplice da eseguire se si forniscono istruzioni più semplici. Il concetto operativo chiave è che ogni istruzione esegue una sola funzione (ad esempio, copiare un valore dalla memoria a un registro).

Le generazioni di architetture Armv1 (1985) fino a Armv5 (1997) con 32 bit/32 bit sono state implementate nelle versioni Core da Arm1 ad Arm10. Ciò ha dato vita a modelli di base che hanno definito gli standard per il settore, come Arm8 (integrato nel celebre microprocessore StrongArm) e Arm7/9 (processore XScale).

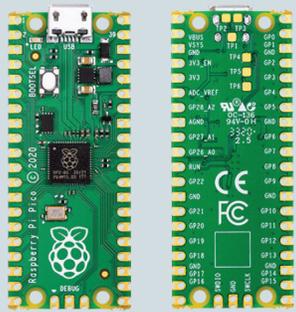
Con il passaggio all'architettura a 32 bit/64 bit e all'architettura delle istruzioni THUMB v2, la generazione Arm v6 (2002) ha dato origine al core Arm11 (ampiamente utilizzato negli smartphone), ma anche all'approccio Arm Cortex: si tratta di versioni di core specializzate per diversi ambiti applicativi. Tutte le architetture Arm più recenti sono organizzate in tre profili: Cortex-M, Cortex-R e Cortex-A.

### Un Cortex per qualsiasi applicazione

Cortex-M: per applicazioni di microcontrollori.

Cortex-R: applicazioni in tempo reale.

Cortex-A: per applicazioni che utilizzano middleware o sistemi operativi.



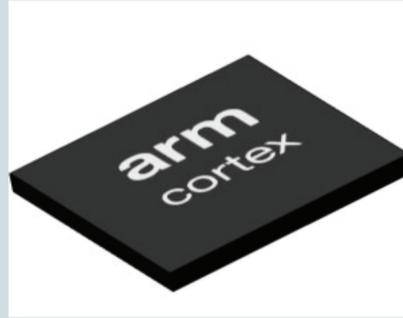
### Cortex-M ("Microcontrollore")

- Armv6-M, Armv7-M e Armv8-M a 32 bit
- Integrati nei microcontrollori, ASIC, FPGA e SoC
- Sistemi integrati, IoT, nodi sensori, semplici funzioni automotive
- Set di istruzioni Thumb

M0	M0+	M1	M3	M4	M7
Arm v6-M	Arm v6-M	Arm v6-M	Arm v7-M	Arm v7E-M +DSP	Arm v7E-M +FPU
<a href="#">nRF51</a>	<a href="#">Raspberry Pi Pico</a>	<a href="#">FPGA ProA-SiC3 Flash</a>	<a href="#">LPC18xx</a>	<a href="#">STM32G4</a>	<a href="#">SMART</a>

M23	M33	M35	M52	M55	M85
Armv8-M + TrustZone	Armv8-M	Armv8-M	Armv8.1-M	Armv8.1-M	Armv8.1-M
<a href="#">SAM</a>	<a href="#">EFM32</a>	<a href="#">ST33K</a>	Dispositivi IoT	<a href="#">Apollo510</a>	<a href="#">Vision Board</a>

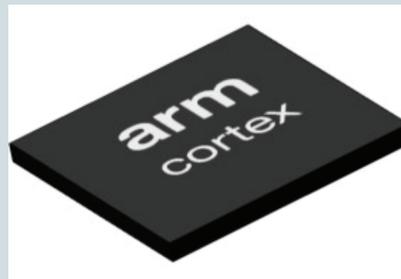
### Cortex-R ("In tempo reale")



- Armv7-R e Armv8-R
- Applicazioni in tempo reale ad elevate prestazioni
- Applicazioni critiche per la sicurezza, tra cui dispositivi medici, sistemi di controllo industriale e sistemi di sicurezza.
- Set di istruzioni Thumb e Thumb2

R4(F)	R5(F)	R7(F)	R8(F)	R52(F)	R52+(F)	R82(F)
Armv7-R	Armv7-R	Armv7-R + DSP	Armv7-R	Armv8-R	Armv8-R	Armv8-R 64-Bit
<a href="#">RM44</a>	<a href="#">TMS570</a>	Dual-Core	Quad-Core	<a href="#">RZ/N2L</a>	SR6P7C4	Soluzioni di archiviazione

### Cortex-A ("Applicazione")



- 32 bit (Armv7-A) e 64 bit (Armv8-A)
- Processori applicativi con supporto per sistemi operativi e applicazioni di terzi.
- Smartphone e server
- Set di istruzioni Thumb e Thumb2

A5 / A7 / A8 / A9 / A12 / A15 / A17	A32 / A34 / A35 / A53 / A57 / A72 / A73	A55 / A65 / A75 / A76 / A77 / A78	A510, A710 e A715	A520 e A720
Armv7-A	Armv8-A	Armv8.2-A	Armv9-A	Armv9.2-A
<a href="#">PC industriali</a>	<a href="#">AM65xx</a>	<a href="#">SHARC</a>	Google Tensor	

### Set di istruzioni

**Thumb:** set di istruzioni compresso: una codifica compatta a 16 bit per un sottoinsieme del set di istruzioni Arm. Risparmio di spazio grazie ad alcuni operandi di istruzioni impliciti e al numero limitato di possibilità rispetto al set di istruzioni Arm.

**Thumb-2:** set di istruzioni a lunghezza variabile, estende il set limitato di istruzioni a 16 bit di Thumb con istruzioni aggiuntive a 32 bit.

**Jazelle:** Jazelle DBX (Direct Bytecode eXecution) è una tecnologia che consente l'esecuzione diretta del bytecode Java.

**VFP:** la tecnologia Vector Floating Point è un'estensione del coprocessore FPU (unità a virgola mobile).

**Neon:** estensione SIMD avanzata, un set di istruzioni SIMD combinato a 64 e 128 bit che fornisce un'accelerazione standardizzata per applicazioni di elaborazione multimediale e di segnali.

**Elio:** questa estensione vettoriale a profilo M (MVE) aggiunge circa 150 istruzioni scalari e vettoriali.

### Altre letture

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

Storia della società Arm: <https://newsroom.arm.com/blog/arm-official-history>