

A RISC-V projekt

Nyílt, moduláris és globális utasításkészlet-architektúra



Felhasznált és ajánlott irodalom

- ❖ Eduardo Corpeño: [An Introduction to RISC-V—Understanding RISC's Open ISA](#)
- ❖ SiFive – [RISC-V 101 \(PDF, bevezető prezentáció\)](#)
- ❖ CERN – [An Introduction to RISC-V](#)
- ❖ RISC-V International: [Official RISC V Specifications](#)
- ❖ Espressif: [ESP32-C3 Wireless Adventure: A Comprehensive Guide to IoT](#)
- ❖ Espressif: [ESP32-C3 Datasheet \(SoC adatlap\)](#)
- ❖ Espressif: [ESP32-C3 Technical Reference Manual \(TRM\)](#)

A RISC filozófiája

❖ **RISC = Reduced Instruction Set Computer**

A fogalmat *David Patterson* alkotta meg az 1980-as évek elején (Berkeley RISC-projekt)

❖ **A cél:** egyszerű, gyorsan végrehajtható utasításokra épülő architektúra, kiszámítható időzítéssel. A komplexitás a hardver helyett a fordítóba kerül

❖ **A RISC-architektúrák jellemzői:**

- sok regiszter
- egységes, fix utasításformátum
- load–store modell (memóriához csak LOAD/STORE fér hozzá)
- egyszerűbb vezérlőlogika → magasabb órajel, jobb energiahatékonyság

❖ **A RISC-V projekt** egy nyílt, moduláris **RISC ISA**-t tervez

Az **ISA** (Instruction Set Architecture) az a formális utasításkészlet és programozási modell, amelyet a hardver megvalósít és amelyre a szoftver épül

Egyszerűségből hatékonyság

- ❖ A **Reduced Instruction Set** nem azt jelenti, hogy kevés az utasítás
 - A modern **RISC** ISA-k (ARMv9, RISC-V) több száz utasítással rendelkeznek
 - A „reduced” arra utal, hogy az utasítások egyszerűek, egységesek és gyorsan végrehajthatók, nem pedig arra, hogy kevés van belőlük
 - A **RISC** lényege: minden összetett művelet egyszerű lépésekre bontható
- ❖ **CISC** (Complex Instruction Set Computer) és a **RISC** összehasonlítása (példa):

CISC	RISC
MUL R1, [MEM]	LOAD R2, [MEM] MUL R1, R2
egyetlen komplex utasítás, memóriaművelet + szorzás, több ciklus	két egyszerű, egyciklusos utasítás, kiszámítható végrehajtás

- ❖ Tanulság: a **RISC** nem „kisebb tudású”, hanem másképp szervezi a komplexitást

A RISC-V projekt eredeti célkitűzése

- ❖ Milyen céllal indult el a **RISC-V**? (UC Berkeley, 2010)
 - **Egy modern, tiszta és oktatható RISC-architektúra létrehozása**, amely mentes a korábbi ISA-k történelmi terheitől
 - **Szabadon hozzáférhető, licencmentes** ISA biztosítása kutatáshoz és oktatáshoz
 - **Egyszerű, moduláris alap** kialakítása, amelyre könnyen lehet új kiterjesztéseket és architektúrákat építeni
 - **Kutatási platform** új mikroarchitektúrák, fordítótechnológiák és rendszerkísérletek számára
 - **Hosszú távon fenntartható, stabil ISA** megalkotása, amely nem kötődik egyetlen céghez vagy technológiai örökséghez

A RISC-V fejlődésének fő állomásai

- ❖ **2010 – UC Berkeley kutatóprojekt**

A RISC-V első verziója megszületik egyetemi környezetben, kutatási célokra

- ❖ **2011–2014 – Az ISA első publikációi**

Megjelennek a korai specifikációk, a közösség és az érdeklődés gyorsan növekedni kezd

- ❖ **2015 – RISC-V Foundation megalakulása**

Ipari szereplők csatlakoznak, a projekt kilép az akadémiai keretek közül

- ❖ **2017–2018 – Kritikus eszköztámogatás megjelenése**

GCC, binutils, Linux kernel és glibc hivatalos támogatást kap

A RISC-V ekkor válik valódi, használható platformmá

- ❖ **2018–2019 – Ipari áttörés**

Nagyvállalatok (Western Digital, NVIDIA, Alibaba, Huawei stb.) bejelentik saját RISC-V terveiket. A RISC-V több milliárd eszközben jelenik meg vezérlőmagként

A RISC-V fejlődésének fő állomásai

❖ 2019 – Költözés Svájcba geopolitikai okokból

Az USA exportkorlátozási szabályai (EAR) és a Huawei-ügy miatt a projekt nyíltságának megőrzése érdekében a szervezet elhagyja az amerikai joghatóságot

❖ 2020 – A RISC-V International Svájcban bejegyezve

A semleges jogi környezet biztosítja, hogy az ISA továbbra is mindenki számára szabadon hozzáférhető maradjon

❖ 2023 – A Debian hivatalos RISC-V architektúra lesz

A nagy Linux-disztribúciók közül elsőként teljes értékű, első osztályú támogatást kap

A RISC-V projekt finanszírozása

❖ **Kezdeti időszak (2010–2014) – Egyetemi kutatás**

A RISC-V a UC Berkeley kutatócsoportjának projektjeként indult

Finanszírozás: amerikai állami kutatási alapok (DARPA, NSF, DOE)

A cél: szabadon használható, modern kutatási-oktatási ISA létrehozása

❖ **Ipari bekapcsolódás (2015–2019) – RISC-V Foundation**

A szervezet működését tagsági díjak fedezték

Premier tagok: nagyvállalatok (Western Digital, NVIDIA, Google, Huawei, Qualcomm)

A költségvetés ekkor már több millió USD évente

❖ **Globális ökoszisztéma (2020–tól) – RISC-V International**

Továbbra is a tagsági díjak adják a fő bevételi forrást

Kiegészítő finanszírozás: konferenciák, munkacsoportok, kutatási programok szponzorálása révén

A szervezet feladata a szabványosítás és az ökoszisztéma koordinálása

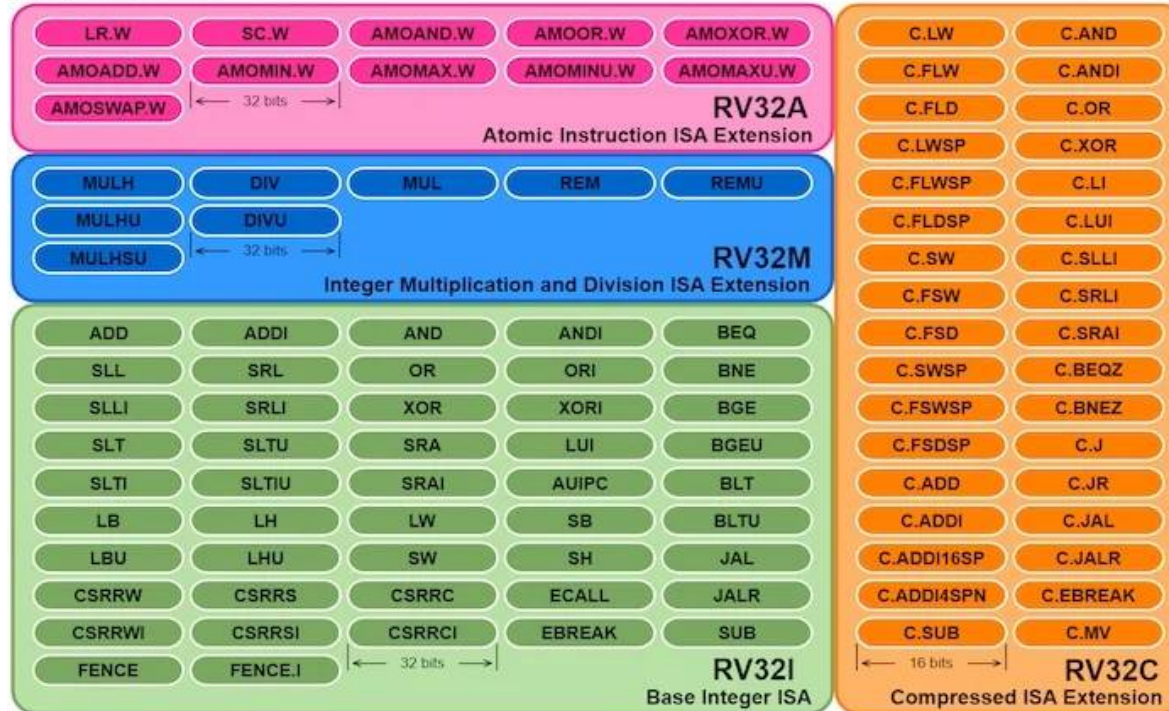
RISC-V ISA és kiterjesztések – konvenciók

- ❖ A **RISC-V** az 1980-ban kezdődött RISC-kutatások 5. generációja, amelyet eleve moduláris ISA-ként terveztek. Minden megvalósítás egy kötelező alap ISA-ból (pl. RV32I) és opcionális kiterjesztésekből épül fel
- ❖ A névkonvenció: **RV\<bitszám\>\<bázis\> + kiterjesztések**
- ❖ Példa: **RV32IMAC** = 32 bites alap utasításkészlet integer műveletekkel
 - **I**: Base Integer ISA
 - **M**: Integer Multiply/Divide
 - **A**: Atomic Instructions
 - **C**: Compressed Instructions
- ❖ A **moduláris felépítés célja**: alkalmazáshoz igazítható CPU-k tervezhetők
- ❖ A fordítóprogram pedig a megadott kiterjesztésekhez igazodva állítja elő a gépi kódot. Ha a program olyan utasítást használna, amelyet a hardver nem támogat, a CPU megszakítást vált ki, és egy szoftveres kezelőrutinhoz ugrik

A RISC-V ISA moduláris felépítése

- ❖ Ahogy az ábrán is látható, a RISC-V ISA egy kötelező alaptól és opcionális kiterjesztésekből áll, amelyek külön blokkokként kapcsolhatók az alap ISA-hoz

RV32IMAC



I: Base Integer ISA
M: Integer Multiply/Divide
A: Atomic Instructions
C: Compressed Instructions

Forrás: Eduardo Corpeño: [An Introduction to RISC-V—Understanding RISC's Open ISA](#)

Base Integer ISA (RV32I)

- ❖ Az alap utasításkészlet 47 utasításból áll, amely a minimálisan szükséges műveleteket tartalmazza
- ❖ 32 darab 32 bites általános célú regiszter + program counter
- ❖ Egyetlen speciális regiszter van: x0, amely mindig 0-t ad vissza.
- ❖ A műveletek lefedik:
 - aritmetika (összeadás, kivonás)
 - bitműveletek
 - load/store (memóriaelérés csak ezekkel)
 - ugrások és elágazások
- ❖ Az ISA 32 bites utasításkódolást használ, egyszerű, egységes formátumokkal

Opcionális RISC-V ISA kiterjesztések

❖ Multiplication & Floating-Point (M, F, D)

- **M**: egész szorzás/osztás hardveres támogatása
- **F / D**: 32/64 bites lebegőpontos műveletek **IEEE-754** szerint

❖ Compressed Instructions (C)

Az RV32C kiterjesztés a leggyakrabban használt utasításokhoz alternatív, 16 bites kódolást biztosít, jelentősen javítva a kód-sűrűséget (a mikrovezérlőknél fontos)

❖ Egyéb kiterjesztések (A, B, V, P, stb.)

- **A**: atomikus műveletek (lock-free szinkronizáció)
- **B**: bitmanipulációs utasítások (gyorsabb bit-szintű műveletek)
- **V**: vektorműveletek (skálázható SIMD)
- **P**: DSP-szerű packed-SIMD műveletek (DSP = Digital Signal Processing)

❖ A moduláris felépítés miatt minden implementáció csak a szükséges funkciókat tartalmazza

RISC-V Privilege módok és memóriavédelem

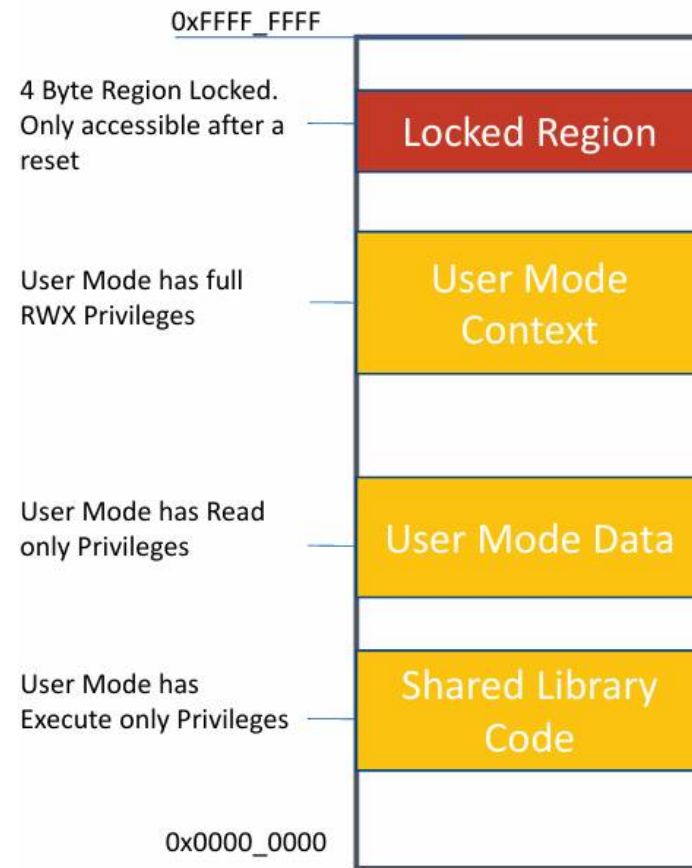
A RISC-V ISA három jogosultsági szintet definiál:

- ❖ M-mode (Machine) – teljes hardverhozzáférés
- ❖ S-mode (Supervisor) – operációs rendszerekhez
- ❖ U-mode (User) – alkalmazások izolált futtatásához

A Linux-képes magokhoz implementálni kell az M/S/U módokat

- ❖ **PMP – Physical Memory Protection** (Fizikai memóriatartományok védelme) amelyet az M-mode állít be. MCU-knál ez a fő biztonsági mechanizmus, és operációs rendszer nélkül is használható

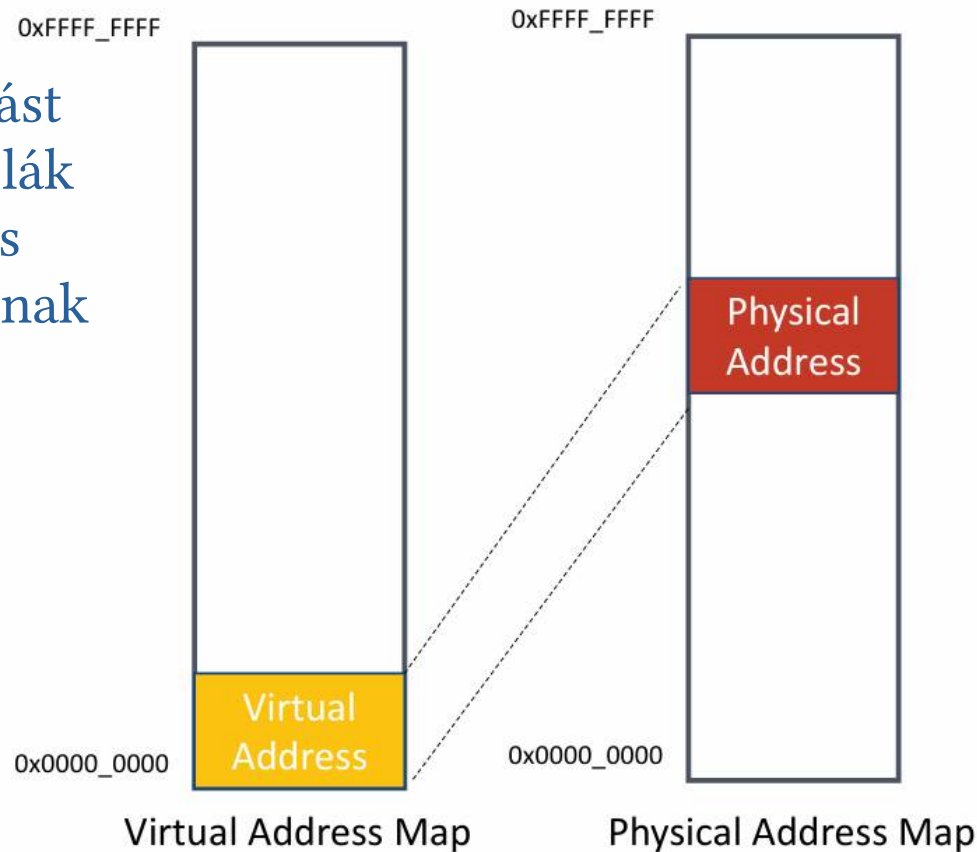
Forrás: SiFive – [RISC-V 101 \(PDF, bevezető prezentáció\)](#)



RISC-V Privilege módok és memóriavédelem

A **RISC-V** támogatja a virtuális memóriát, amely fejlett memóriakezelést és OS-támogatást (pl. Linux) tesz lehetővé. A többszintű lap-táblák a virtuális címeket fizikai címekre fordítják, és hozzáférési jogosultsági adatokat is tartalmaznak

- ❖ Supervisor módot igényel
- ❖ Sv32 (RV32)
 - 32 bites virtuális cím
 - 2 szintű lapozás: 4 KiB, 4 MiB lapok
- ❖ Sv39 (RV64)
 - 39 bites virtuális cím
 - 3 szintű lapozás: 4 KiB, 2 MiB, 1 GiB



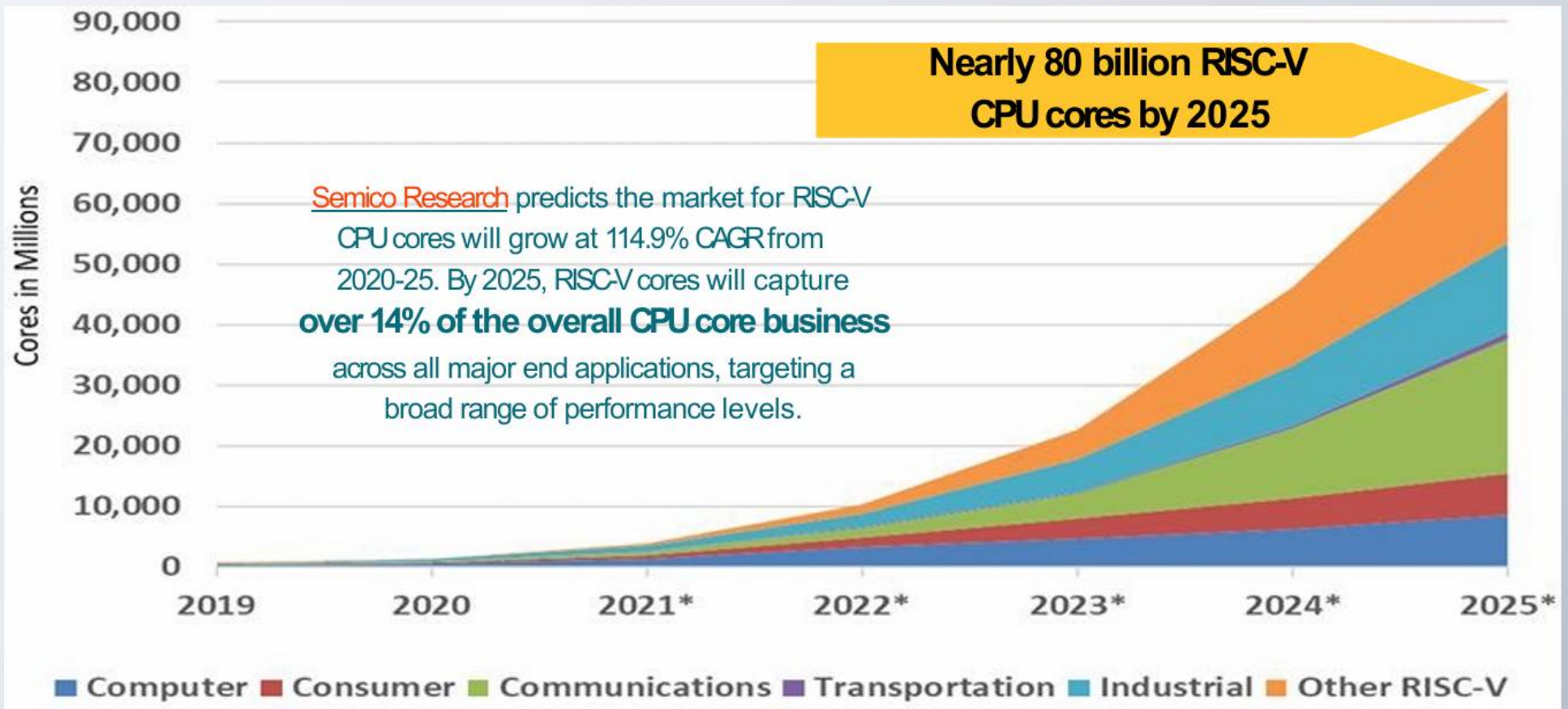
Forrás: SiFive – [RISC-V 101 \(PDF, bevezető prezentáció\)](#)

CPU-magok méretösszehasonlítása

- ❖ Az alábbi táblázatban kis méretű, 32 bites MCU-processzormagok (ARM és RISC-V) kapuszám szerinti összehasonlítása látható

CPU-mag	Kapuszám (becsült)	Megjegyzés
ARM Cortex-M0	~12 000	ipari referencia
PicoRV32 (minimál RV32I)	~10 000 – 15 000	32 bites integer alap utasításkészlettel.
PicoRV32 (IMC)	~20 000 – 30 000	M és C kiterjesztéssel
VexRiscv	~20 000 – 40 000	gyorsabb pipeline, plugin-alapú
ESP32-C3 (T-Head E907 mag)	~60 000 – 80 000	3-lépcsős pipeline, branch predictor, interrupt controller, debug, stb.

Rapid RISC-V growth led by industrial





RISC-V International now > more than 3,800 members
70 countries

RISC-V Free, open, extensible ISA for all computing devices



Nem konvencionális RISC-V alkalmazások

- ❖ A **RISC-V** nem csak önálló CPU-ként jelenik meg: számos nagy gyártó használja beágyazott vezérlőként GPU-kban, AI-gyorsítókban, hálózati és adattároló chipekben. A cél: rugalmas, testre szabható, licencdíj-mentes vezérlőmag
- ❖ **NVIDIA – GPU-vezérlőmagok:** Az NVIDIA GPU-kban több tucat RISC-V mag dolgozik firmware-szintű vezérlőként (power management, scheduling, biztonsági funkciók)
- ❖ **Google – TPU és adatközponti gyorsítók:** A Google saját RISC-V magokat használ az AI-gyorsító processzora és más adatközponti gyorsítók belső vezérlőegységeiben
- ❖ **Western Digital – háttértár-vezérlők:** A WD/HGST meghajtókban RISC-V alapú vezérlők futnak (flash-kezelés, wear-leveling, ECC)
- ❖ **Intel – beágyazott vezérlők és IP-blokkok:** Az Intel több platformon alkalmaz RISC-V magot alrendszerek vezérlésére, és saját RISC-V processzormag-terveket is kínál
- ❖ **Samsung – memóriavezérlők és biztonsági alrendszerek:** A Samsung memóriavezérlőkben és mobilos rendszerchipek egyes alrendszereiben **RISC-V** alapú mikromagok jelennek meg energiagazdálkodási- és biztonsági feladatokra

Kínai nagyipari gyártók

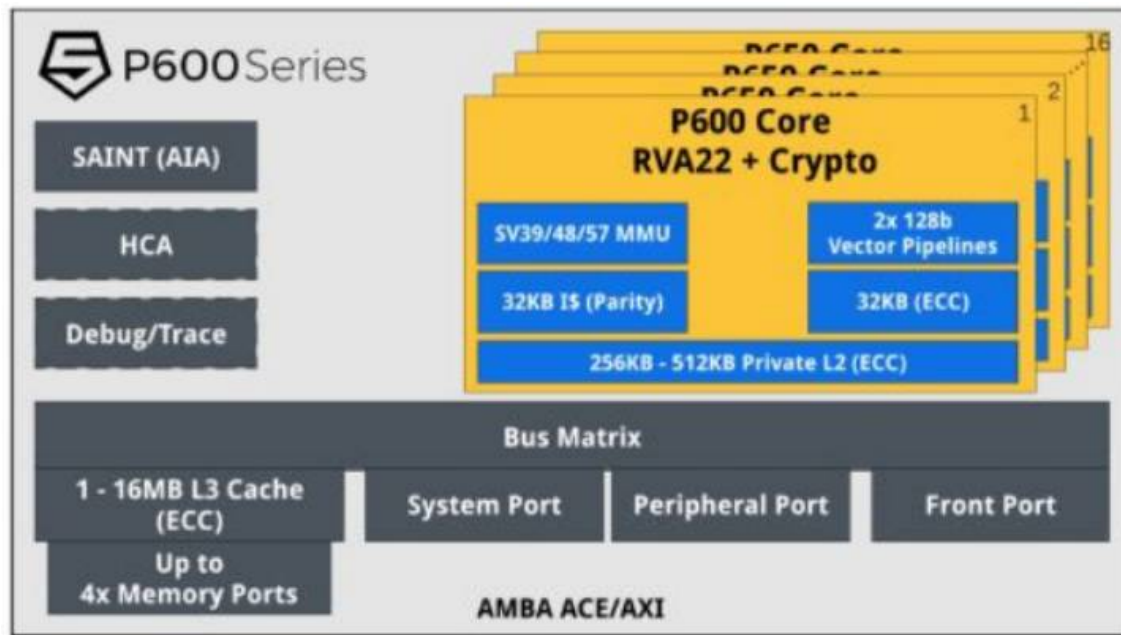
- ❖ A kínai gyártók kevésbé transzparenssek, ezért ritkán derül ki pontosan, hogy hol használnak RISC-V-alapú belső vezérlőmagokat AI gyorsítókban és speciális SoC-okban
- ❖ **Alibaba T-Head**
 - RISC-V mikromagok AI-gyorsítók és adatközponti chipek belső vezérlőiben
 - Xuantie magok széles körben beépítve saját infrastruktúrába
- ❖ **Huawei / HiSilicon**
 - RISC-V alapú IoT és autóipari vezérlőchipek
 - Belső alrendszerekben (PMU, security, always-on) megjelenő RISC-V mikromagok
- ❖ **Cambricon**
 - AI-gyorsítók menedzsment- és vezérlőegységeiben RISC-V mikromagok
 - Heterogén gyorsítóarchitektúrák belső firmware-vezérlése
- ❖ **Birentech**
 - Adatközponti GPU-k (BR100/BR104) belső vezérlőmagjai
 - Energia- és ütemezési alrendszerek RISC-V-vel
- ❖ **Zhaoxin / Loongson**
 - SoC-alrendszerek (I/O, PMU, biztonsági modulok) RISC-V-alapú mikromagokkal
 - Nem a fő CPU, hanem háttérvezérlő szerepben

RISC-V implementációk – CPU és SoC

- ❖ A **RISC-V** csak egy nyílt utasításkészlet. A tényleges hardvert két szinten valósítják meg:
- ❖ **RISC-V mag (CPU core)** - az ISA konkrét hardveres megvalósítása, amely a RISC-V utasításkészletet hajtja végre. Ezek a magok licencelhetők, testre szabhatók, és ugyanaz a mag több, egymástól teljesen különböző SoC-ban is megjelenhet
- ❖ **RISC-V SoC (System-on-Chip):** A SoC a CPU-mag köré épített teljes rendszer: memóriavezérlők, buszok, perifériák, gyorsítók, I/O blokkok, rádiós egységek stb. A SoC-gyártók gyakran külső magokat licencelnek, vagy sajátot fejlesztenek
- ❖ **RISC-V mag-implementátorok**
 - SiFive – U-, S-, E-sorozat; ipari standard
 - Alibaba T-Head – Xuantie C906/C910/C920
 - Andes Technology – N- és AX-sorozat
 - OpenHW Group – nyílt forrású magok
 - Codaip – alkalmazás-specifikus magok
- ❖ **RISC-V SoC-implementátorok**
 - Espressif – ESP32-C3 / C6 / H2 IoT
 - StarFive – JH7100 / JH7110 (VisionFive SBC)
 - Allwinner – D1 / D1s (MangoPi, LicheeRV)
 - WCH – CH32V-sorozat (003, 203, 305)
 - GigaDevice – GD32VF103 (Longan Nano)

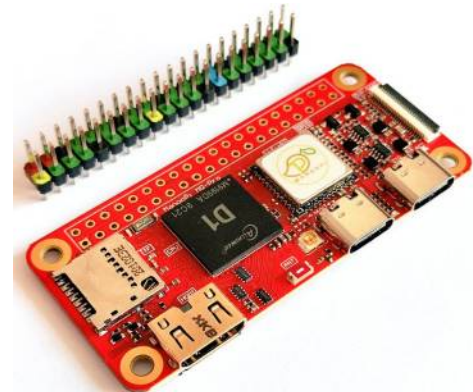
SiFive P470 / P670 – csúcskategóriás RV64 magok

- ❖ A **SiFive Performance** sorozat a RISC-V ökoszisztéma „flagship” CPU-magja:
- ❖ **Fő jellemzők:**
 - 64 bites, Linux-képes RISC-V magok, modern RISC-V profilok
 - Out-of-order végrehajtás a magas egy szálás teljesítményért
 - Többmagos, SMP
 - GHz-es órajelek
- ❖ **Tipikus alkalmazások:**
 - Rendszámfelismerő kamerák
 - Okos routerek / otthoni hálózati központok
 - Ipari adatgyűjtő és vezérlő/ beágyazott Linux
 - Autós fedélzeti egységek (infotainment / műszerfal)



Alibaba T-Head RISC-V magok és SoC-ok

- ❖ Az Alibaba T-Head három nagy RISC-V magcsaládot gyárt:
 - **C906** – középkategóriás, Linux-képes mag
 - **C910** – nagy teljesítményű, out-of-order mag
 - **C920** – még erősebb, RVV-t támogató mag
- ❖ **Tipikus alkalmazások:**
 - **Milk-V Pioneer** – RISC-V laptop (TH1520 SoC, 4× C910 mag)
 - **Milk-V Meles** – mini PC / SBC (TH1520 SoC, 4× C910 mag)
 - **MangoPi MQ-Pro** – ultraolcsó Linux SBC (Allwinner D1, C906 mag)



GreenWaves GAP8 / GAP9

- ❖ A **GAP8** és **GAP9** európai (Grenoble, Franciaország) fejlesztésű, ultraalacsony fogyasztású **RISC-V** klaszterprocesszorok, amelyek 1 vezérlőmagot és 8 párhuzamos compute magot használnak helyi AI- és képfeldolgozási feladatokra. A fogyasztás mW tartományban marad, így hónapokig működő akkumulátoros eszközökbe is alkalmasak
- ❖ **Konkrét alkalmazás:**
akkumulátoros okos biztonsági kamera, amely helyben futtat mozgás- és emberfelismerést (CNN), és csak releváns képkockákat küld tovább, így minimális energiafogyasztással működik
- ❖ **További felhasználások:**
 - drónok (akadályfelismerés)
 - ipari szenzorok (rezgésanalízis)
 - viselhető eszközök (gesztus- és hangfelismerés)

Mire jó az AI egy fülhallgatóban?

- ❖ A **GreenWaves GAP9 RISC-V** processzor valós időben futtat AI-alapú audioalgoritmusokat TWS fülhallgatókban és más hordható eszközökben
- ❖ Mit tud a **GAP9** a fülhallgatóban?
 - **AI-alapú aktív zajszűrés:** neurális hálós „acoustic scene detection” → környezetfüggő, intelligens ANC.
 - **Forrás-szétválasztás:** valós idejű vokál/zenesáv szétválasztás, beszédkiemelés (AudioSourceRE).
 - **Hangazonosítás:** ultra-kisméretű, alacsony fogyasztású voice-ID hitelesítés (eszköz feloldása, fizetés jóváhagyása)
 - **Immersive 3D audio:** fejkövetéssel működő térhangzás mW-os fogyasztással (IDUN audio)
 - **Augmented hearing:** zajos környezetben is tiszta beszéd, kritikus kommunikáció javítása (Augmented Hearing)



GREENWAVES
TECHNOLOGIES

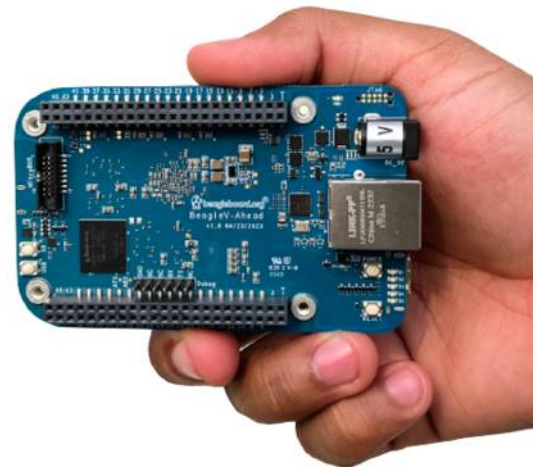
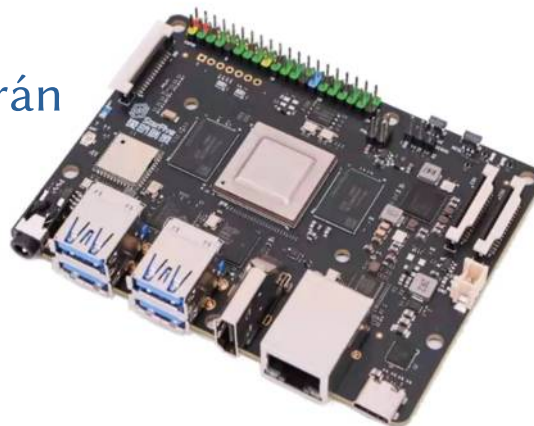
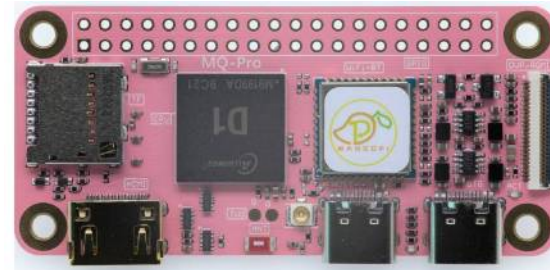
Mikroprocesszor a forrasztópákában?!

- ❖ Az előző példában egy vezeték nélküli fülhallgatót „okosított” egy **GAP9** RISC-V chip – de ennél extrémebb is alkalmazásokat is találunk, mint például a **Pinecil V2 (Pine64)** – **RISC-V forrasztópáka**, amelyben egy GigaDevice **GD32VF103TB** MCU ketyeg
- ❖ **Miért kell ekkora teljesítmény egy pákába?**
 - A modern hordozható pákák USB-C tápról működnek, kommunikálniuk kell a töltővel: feszültséget kérnek, módosítanak, figyelnek. Ez komoly protokollkezelést igényel.
 - Gyors és precíz PID hőmérséklet-szabályozás
 - Felhasználói felület és „okos” funkciók: Kijelző, menürendszer, profilok, alvó mód, mozgásérzékelő (ha leteszed, visszahűt)
 - Biztonság és védelem: Hibás szenzor felismerése, túlmelegedés elleni védelem, időzített lekapcsolás, feszültségfigyelés.
 - Nyílt firmware és frissíthetőség (IronOS nyílt forrású firmware)



RISC-V alapú hobbiszintű SBC-k

- ❖ A **RISC-V** már a hobbiszintű mini-számítógépek világában is megjelent. Ezek az SBC-k teljes Linuxot futtatnak, GPIO-val és fejlesztőbarát környezettel
- ❖ **Néhány jól használható, megfizethető opció:**
 - MangoPi MQ-Pro (Allwinner D1) 1 mag, 1 GB RAM
 - StarFive VisionFive 2 – 4 mag, 1–8 GB RAM, HDMI + PCIe
 - Milk-V Mars – Pi 4 kategória, 4 mag, akár 8 GB RAM
 - Sipeed Lichee Pi 4A – 4 mag, NVMe támogatás, fejlesztők kedvence
 - BeagleV-Ahead – 64 bites RISC-V, 4 GB RAM, nyílt hardver
- ❖ **Miért érdekesek?**
 - Teljes értékű Linux RISC-V architektúrán
 - Nyílt ISA → oktatásbarát, átlátható
 - Gyorsan fejlődő szoftveres támogatás



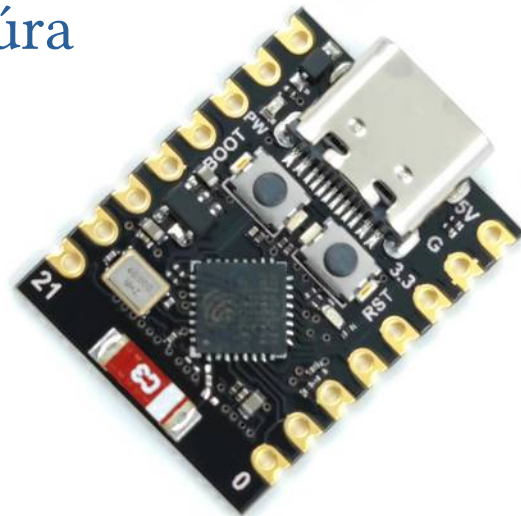
RISC-V mikrokontroller kártyák hobbitáknak

- ❖ **CH32V003** – a „RISC-V ATtiny”
48 MHz, 2 kB RAM, 16 kB Flash
- ❖ **CH32V203** – belépőszintű MCU
72 MHz, 20 kB RAM, 64 kB Flash
- ❖ **GD32VF103** (Longan Nano) – klasszikus hobbikártya
108 MHz, 32 kB RAM, 128 kB Flash
Jó tanulóplatform, bár már nem a legfrissebb
- ❖ **ESP32-C3** – a „Wi-Fi-s RISC-V”
160 MHz, 400 kB SRAM, Wi-Fi 4 + Bluetooth LE
ESP-IDF, Arduino, CircuitPython támogatás
- ❖ **ESP32-C6** – új generáció, nagyobb tudás
160 MHz RISC-V mag, Wi-Fi 6 + BLE 5.3
Több periféria, modernebb rádiós képességek



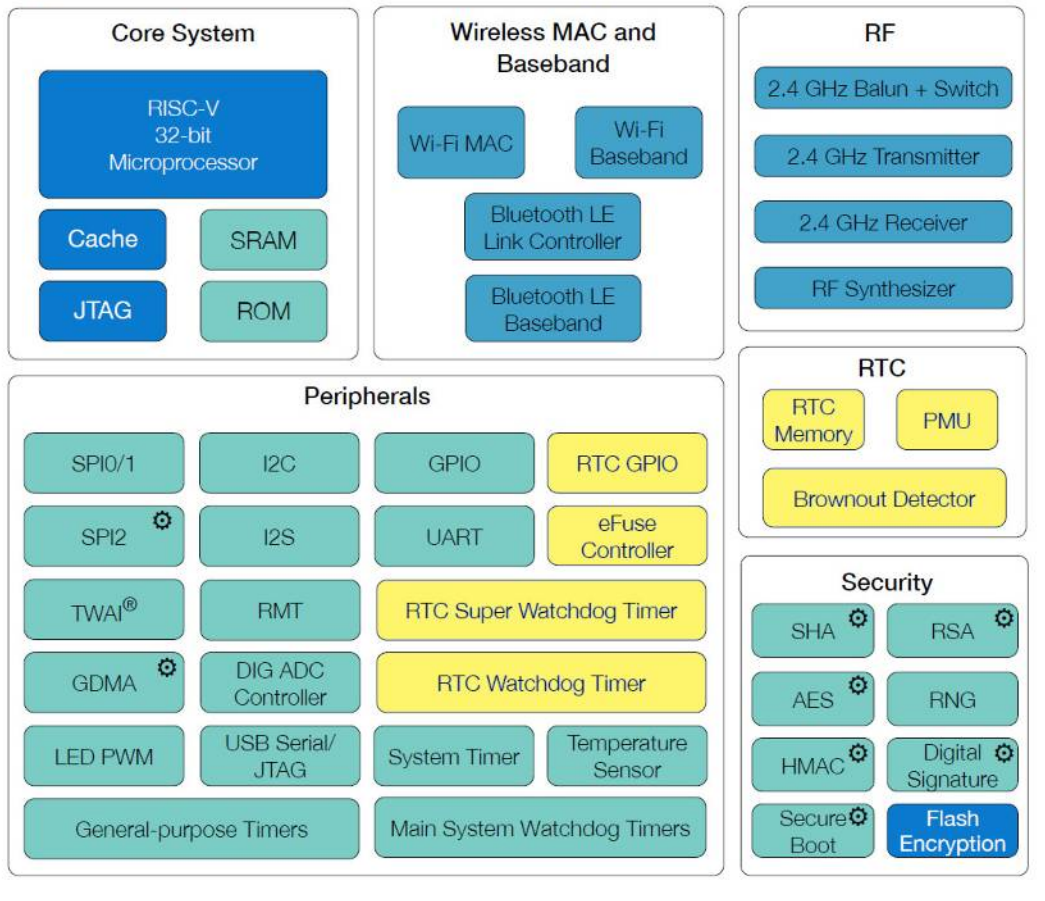
Az ESP32 C3 Super Mini kártya

- ❖ Az **ESP32-C3 Super Mini** egy kompakt fejlesztői kártya
- ❖ Mikrovezérlő: **ESP32-C3**, 32 bites **RISC-V** architektúra
- ❖ Órajel: Akár 160 MHz
- ❖ Memória: 400 KB SRAM, 384 KB ROM
- ❖ Tároló: 4 MB Flash memória
- ❖ Wi-Fi: 802.11 b/g/n (2.4 GHz)
- ❖ Bluetooth: Bluetooth 5.0 LE
- ❖ GPIO: Több mint 20 általános célú bemenet/kimenet (GPIO), melyek közül 13 van kivezetve
- ❖ Interfészek: SPI, I2C, UART, ADC, DAC, PWM



Az ESP32-C3 SOC blokkvázlata

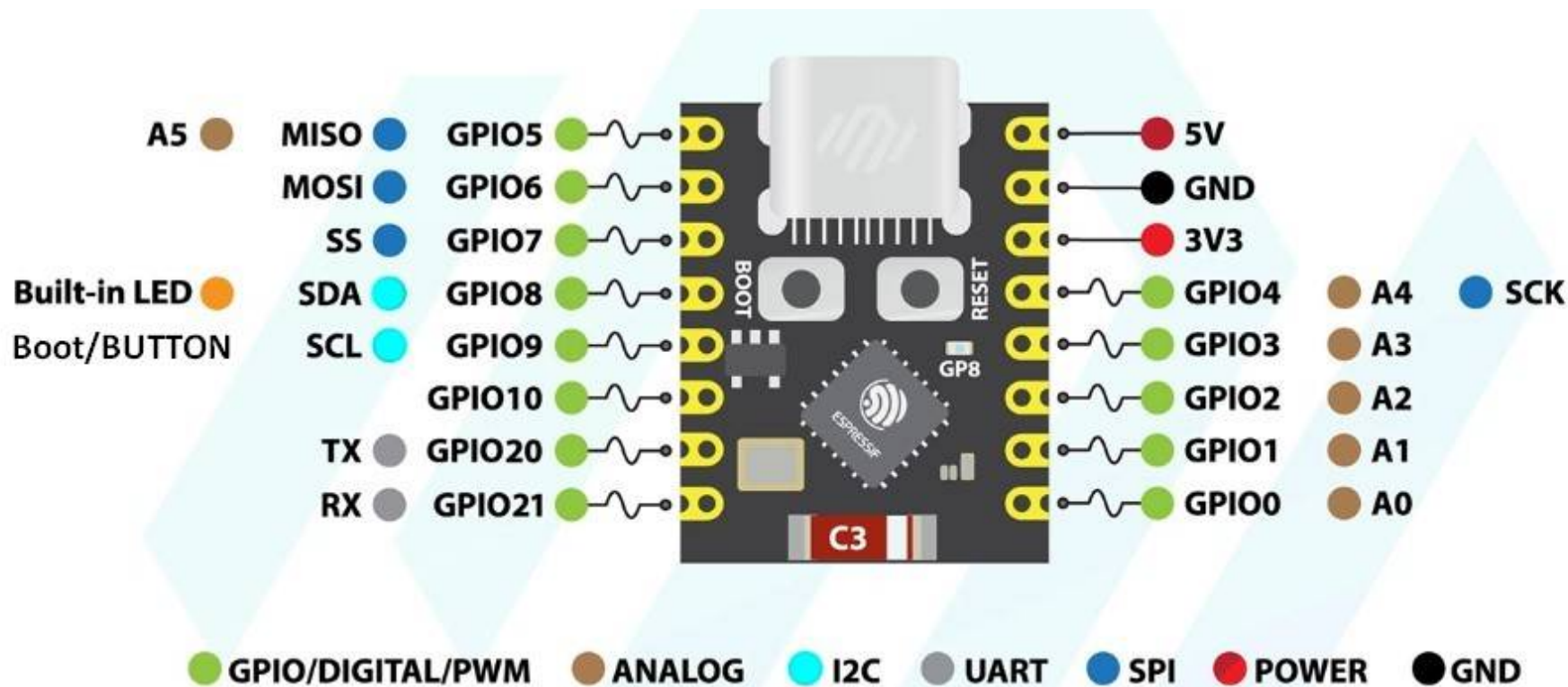
Espressif's ESP32-C3 Wi-Fi + Bluetooth® Low Energy SoC



Modules having power in specific power modes:

- Active
- Active and Modem-sleep
- Active, Modem-sleep, and Light-sleep
- All modes
- optional in Light-sleep

Az ESP32 C3 Super Mini kártya kivezetései



ESP32 C3 Super Mini

Megjegyzés: Az A5 analóg bemenet (ADC2) nem használható, ha a WiFi használatban van!