

**ARM**

**■ POWERED**

# ARM (Advanced RISC Machine)

1983 kísérleti projekt Acorn Computers Ltd .,

1985 ARM1 fejlesztői minták,

1985 ARM2 32 bites adatbusz 64MB memória címezhető,

1989 ARM3 4K cache,

1990 ARM név változtatás Advanced Risc Machine-ra,

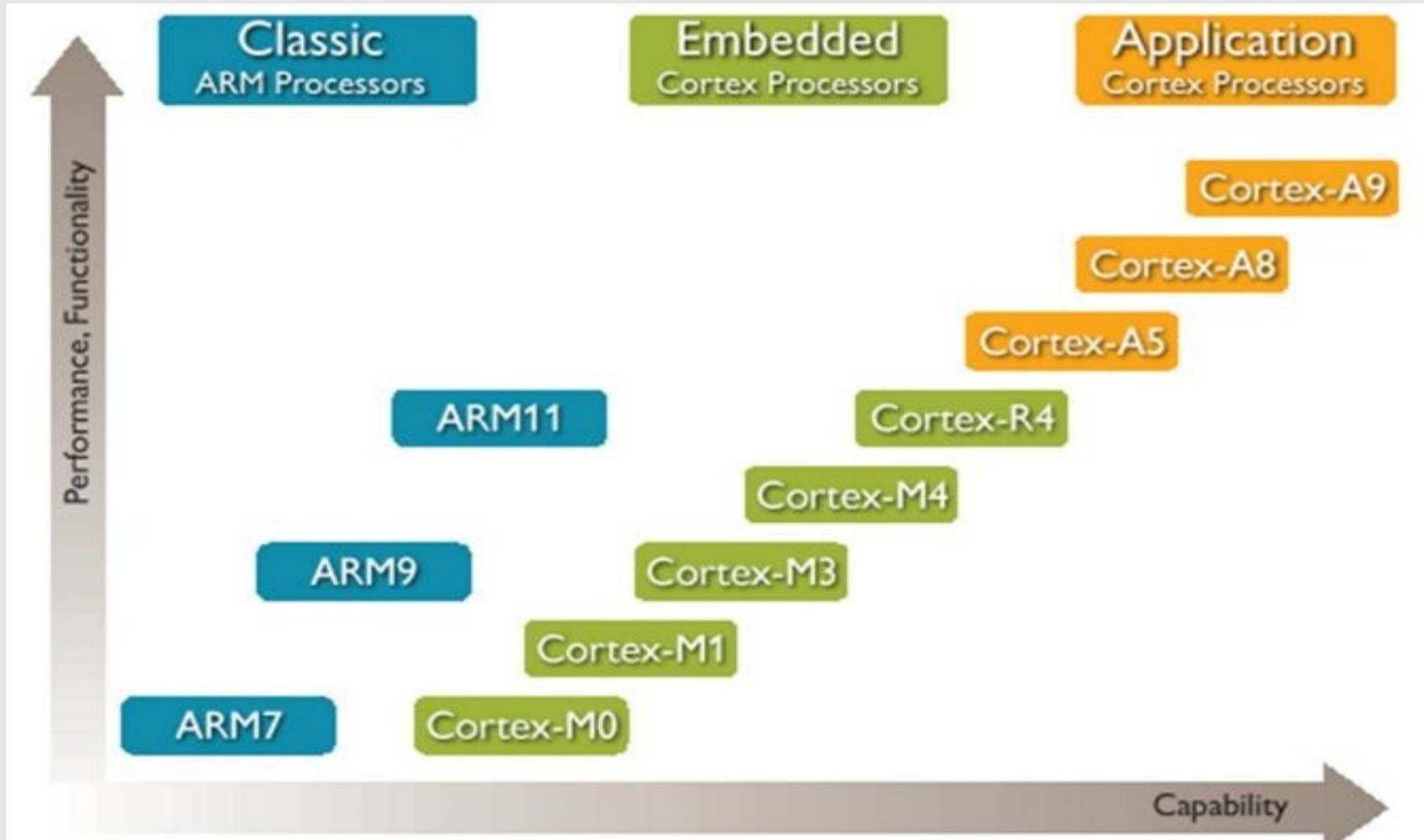
1991 ARM6 Apple megvásárolja a magot és beépíti,

1994 ARM7 kifejlesztése,

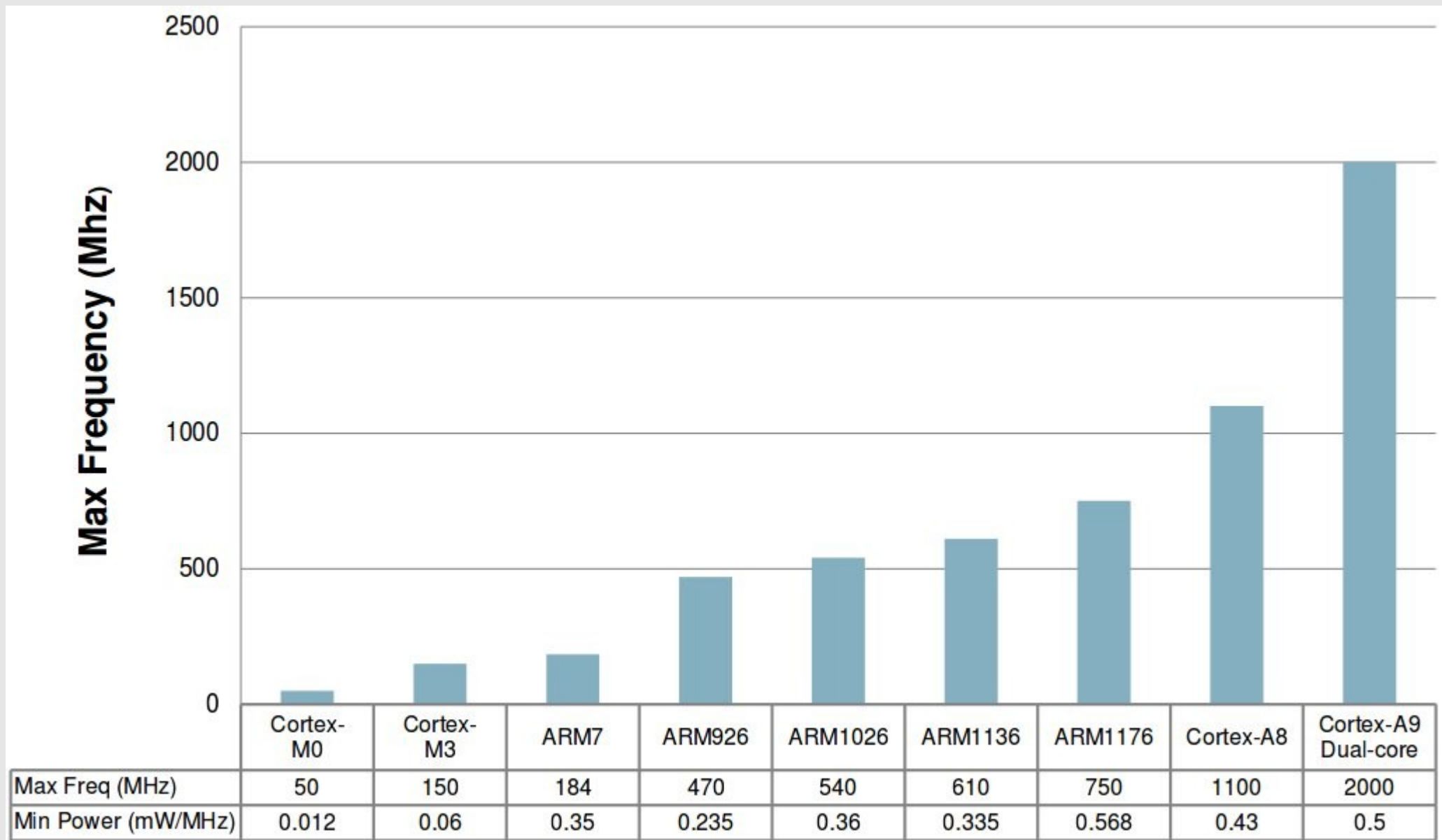
1998 ARM9 kifejlesztése,

2002 ARM11 megjelenése.

# ARM Cortex M és A sorozat




# Teljesítmények



# Cortex A sorozat

- 32 bites RISC processzormagok,
- egyszeres és kétszeres pontosságú lebegőpontos támogatás,
- adott az MMU,
- képes az összes ARM utasításkészlet futtatására.

# ARM Cortex Mx

ARM Cortex-M	SysTick Timer	Bit-banding	Memory Protection Unit (MPU)	Tightly-Coupled Memory (TCM)	CPU cache	Memory architecture	ARM architecture
Cortex-M0 <sup>[1]</sup>	Optional*	Optional <sup>[9]</sup>	No	No	No <sup>[10]</sup>	Von Neumann	ARMv6-M
Cortex-M0+ <sup>[2]</sup>	Optional*	Optional <sup>[9]</sup>	Optional (8)	No	No	Von Neumann	ARMv6-M
Cortex-M1 <sup>[3]</sup>	Optional	Optional	No	Optional	No	Von Neumann	ARMv6-M
Cortex-M3 <sup>[4]</sup>	Yes	Optional*	Optional (8)	No	No	Harvard	ARMv7-M
Cortex-M4 <sup>[5]</sup>	Yes	Optional*	Optional (8)	No	 Possible <sup>[11]</sup>	Harvard	ARMv7E-M
Cortex-M7	Yes	TBD*	Optional (8 or 16)	Optional	Optional	Harvard	ARMv7E-M

# ARM® Cortex®-M0

Nested Vectored  
Interrupt Controller

Wake Up Interrupt  
Controller Interface

CPU

AHB-lite  
Interface

Data  
Watchpoint

Breakpoint

Debug  
Access  
Port

# ARM Cortex M0

- 32 bites ARMv6-M RISC Neumann architektúra
- két állapotú utasítás pipeline
- NVIC (Nested Vector Interrupt Controller)
- hardver által meghatározott megszakítási belépési pontok
- alvó üzemmód, teljesítmény csökkentett mód és jelentős teljesítmény csökkentett mód
- Thumb2-es utasításkészlet (56 utasítás)
- hardver szorzó
- SWD lehetőség (Serial Wire Debug)
- felfelé kompatibilis az M3 és az M4 magokkal



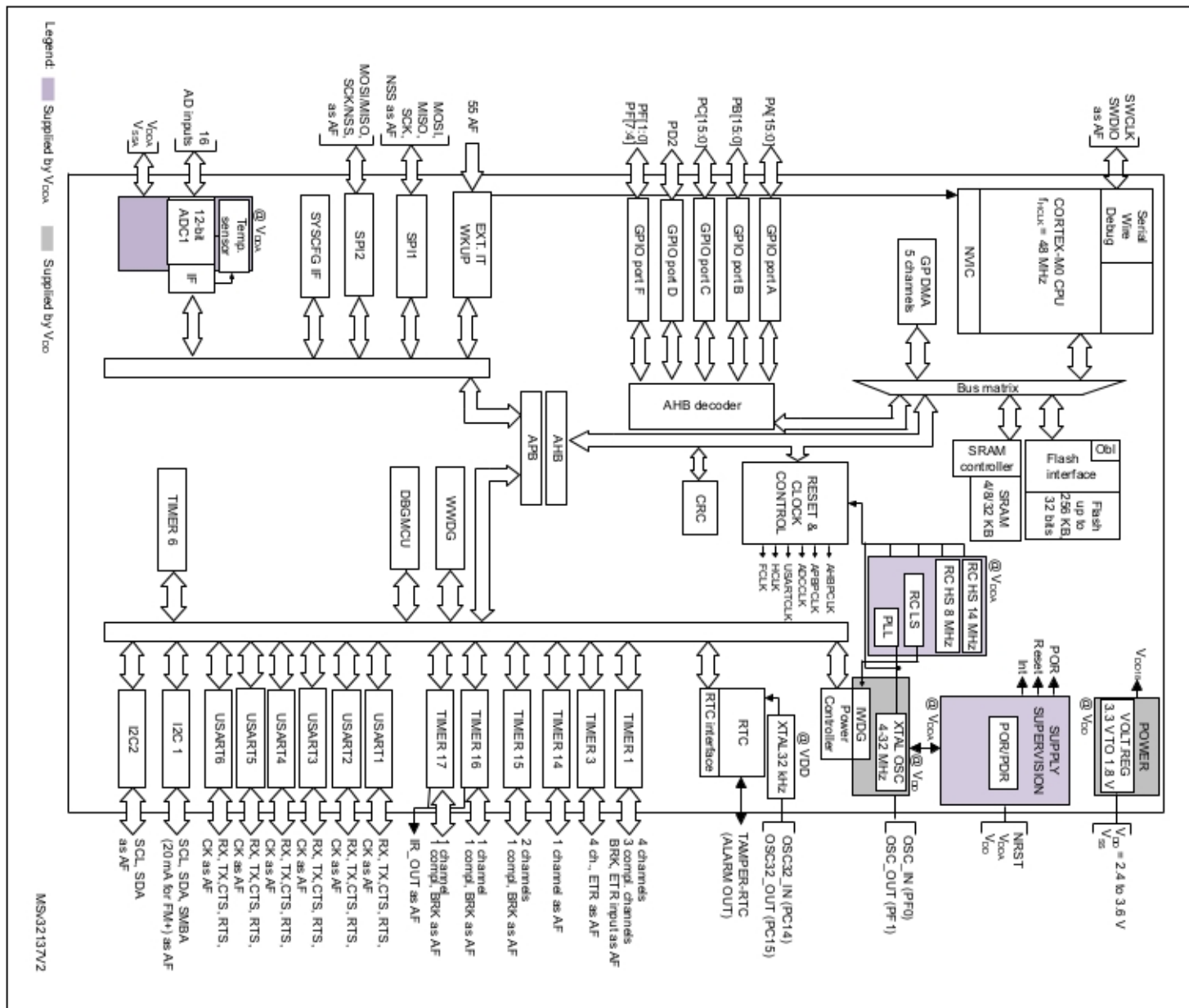
# ARM Cortex M0

tipikus képviselő STM32F0xx

Peripheral		STM32 F030F4	STM32 F030K6	STM32 F030C6	STM32 F030C8	STM32 F030CC	STM32 F030R8	STM32 F030RC
Flash (Kbytes)		16	32	32	64	256	64	256
SRAM (Kbytes)		4	4	4	8	32	8	32
Timers	Advanced control	1 (16-bit)						
	General purpose	4 (16-bit) <sup>(1)</sup>	4 (16-bit) <sup>(1)</sup>	4 (16-bit) <sup>(1)</sup>	5 (16-bit)			
	Basic	-	-	-	1 (16-bit) <sup>(2)</sup>	2 (16-bit)	1 (16-bit) <sup>(2)</sup>	2 (16-bit)
Comm. interfaces	SPI	1 <sup>(3)</sup>	1 <sup>(3)</sup>	1 <sup>(3)</sup>	2	2	2	2
	I <sup>2</sup> C	1 <sup>(4)</sup>	1 <sup>(4)</sup>	1 <sup>(4)</sup>	2	2	2	2
	USART	1 <sup>(5)</sup>	1 <sup>(5)</sup>	1 <sup>(5)</sup>	2 <sup>(6)</sup>	6	2 <sup>(6)</sup>	6
12-bit ADC (number of channels)		1 (11 channels)	1 (12 channels)	1 (12 channels)	1 (12 channels)	1 (12 channels)	1 (18 channels)	1 (18 channels)
GPIOs		15	26	39	39	37	55	51
Max. CPU frequency		48 MHz						

# ARM Cortex M0

tipikus képviselő STM32F0xx



# ARM Cortex M1

- FPGA soft-core optimalizált processzor mag
- 32 bites ARMv6-M RISC Neumann architektúra
- 3 állapotú utasítás pipeline
- Thumb (nem teljes) Thumb2 (néhány) utasításkészlet
- 32 bites hardver szorzás
- 1-32 megszakítás
- opcionális szorosan csatolt memória

# ARM Cortex M3

- 32 bites ARMv7-M RISC Harvard architektúra
- 3 állapotú utasítás pipeline ugrás figyelemmel
- Thumb (teljes) és Thumb-2 (teljes) utasításkészlet
- 32-bit hardver szorzás 64 bites eredménnyel előjelesen és előjel nélkül
- 32-bit hardver osztás
- telítődés figyelés és kezelés
- DSP bővítés
- 1 - 240 megszakítás és NMI
- 12 órajel hosszúságú megszakítás belépés
- integrált sleep módok

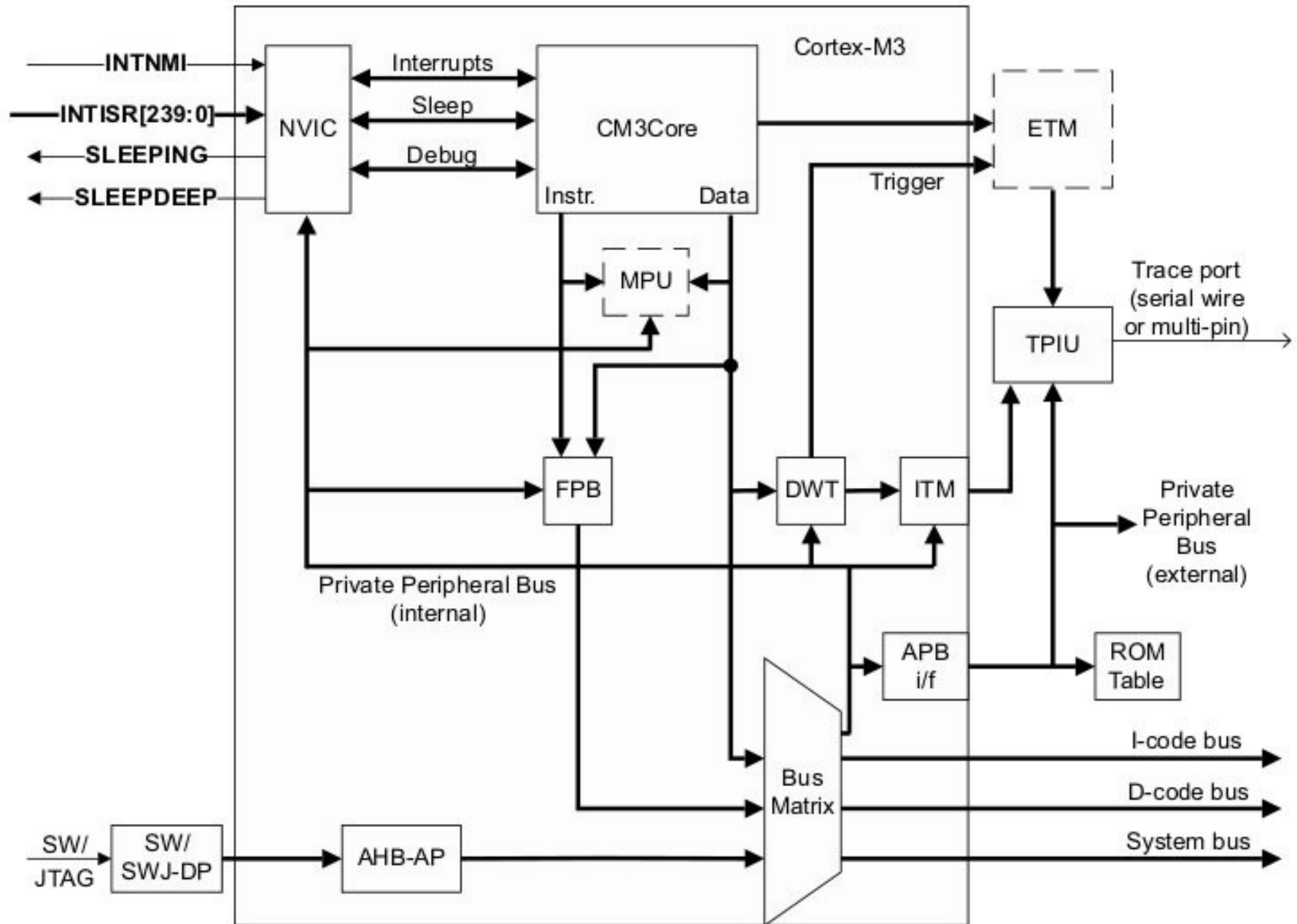
# DMIPS teszt

A Dhrystone benchmark egy olyan sebesség teszt, amely a processzor futási sebességét csak és kizárólag fixpontos utasítások végrehajtására vizsgálja.

A programot 1984-ben Reinhold P. Weicker fejlesztette ki.

Több programozási nyelven implementálták.

# ARM Cortex M3



# ARM Cortex M3

## NVIC

- 1 -240 konfigurálható megszakítási lehetőség
- beállítható megszakítási prioritások
- él és szint érzékeny megszakítás
- dinamikusan állítható prioritási szintek
- prioritások csoportosítása
- megszakítások továbbfűzése
- processzor állapot automatikus mentése és visszaállítása kilépésnél
- 1.25 DMIPS

Sorszám	Prioritás	Funkció
0.	-	SP kezdőcíne
1.	-3	Reset vektor
2.	-2	NMI
3.	-1	Hard hiba
4.	pgm.	Memória védelmi hiba
5.	pgm.	Busz hiba
6.	pgm.	Kezelési hiba
7-10.	-	Foglalt
11.	pgm.	Supervisor hívás
12.	pgm.	Debug monitor
13.	-	Foglat
14.	pgm.	Meghívható rendszerhívás
15.	pgm.	SysTick megszakítás
16-240.	pgm.	Felhasználói megszakítások



# ARM Cortex M3

## **BUSZ MÁRTIX**

- ICode bus. Utasítások és vektorok letöltésére a kód területről 32 bites AHB-Lite busz
- DCode bus. Adatok és debug információk elérése 32 bites AHB-Lite busz
- System bus. Utasítások, vektorok és adatok elérése a rendszer területről 32 bites AHB-Lite busz
- PPB adatok elérése a PPB területről 32 bites APB busz.

# ARM Cortex M3

## **FPB** (Flash Patch and Breakpoint)

- hat hardveres töréspont
- két literál töréspont

## **DWT** (Data Watchpoint and Trace)

- 4 komparátor hardver watchpointnak, vagy ETM triggernek
- számlálók futásidő mérésre
- PC értékek felküldése pl. ISR teljesítmény mérésre

# ARM Cortex M3

## **ITM** (Instrumentation Trace Macrocell)

- szoftver követési lehetőség az alkalmazás számára
- hardveres követési lehetőség a DWT által generált csomagok küldésére
- időbélyezés

# ARM Cortex M3

**MPU** (Memory Protection Unit)

- memória terület védelem (opcionális)

**ETM** (Embedded Trace Macrocell)

- utasítások követését teszi lehetővé

**TPIU** (Trace Port Interface Unit)

- egy híd a Cortex M3 mag ITM ETM és a chip Trace Port Analizátora között (ha van)

**SW/SWJ-DP** (Debug Port)

- szabványos debug interfész JTAG számára

# ARM Cortex M3

## Üzem módok:

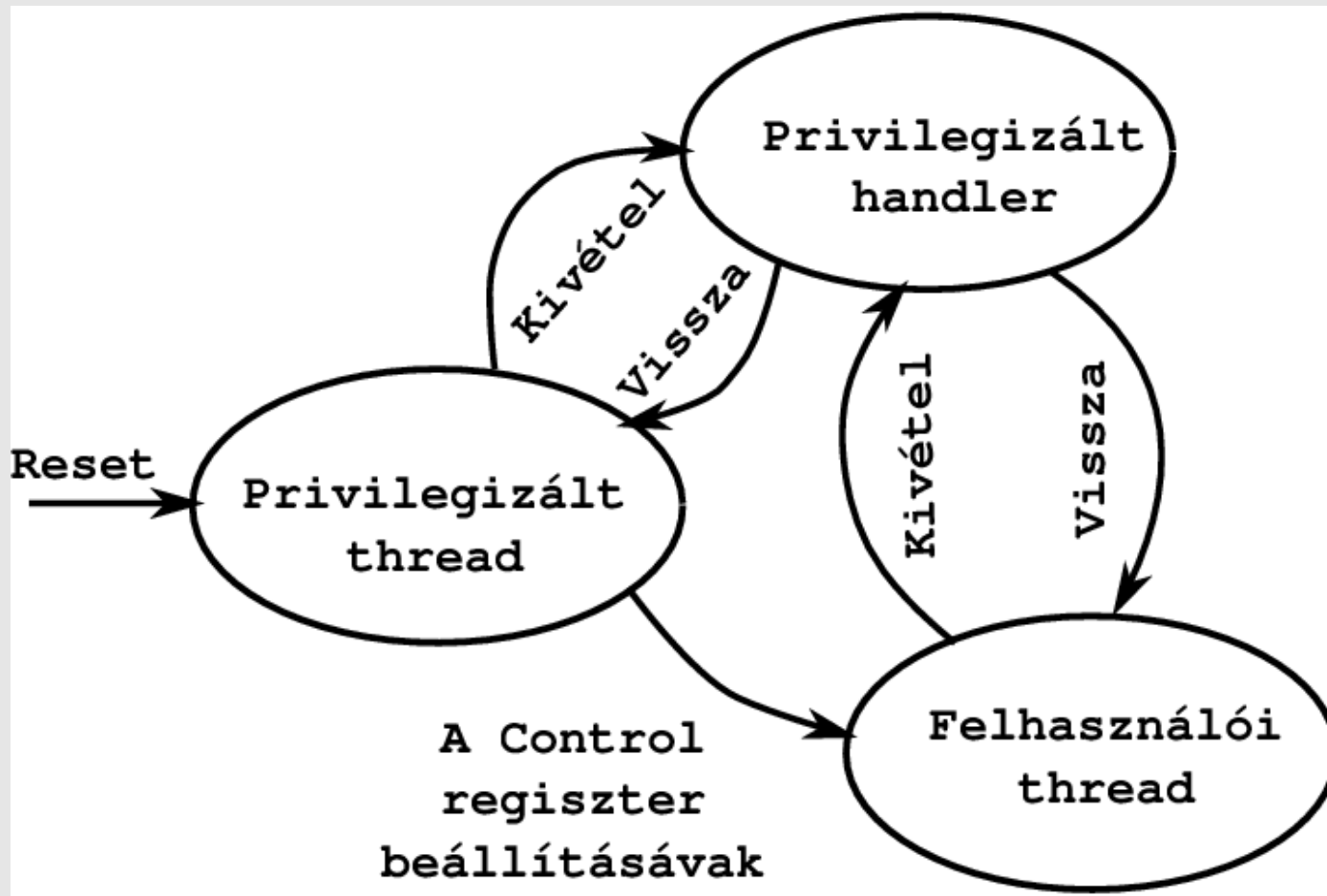
### Thread mód

- RESET esetén a processzor ebben a módban ébred, de beléphet kivétel kezelés esetén is ide. Privilegizált és felhasználói (user) ebben a módban futhat.

### Handler mód

- Kivételkezelés esetén ebben a módban fut a processzor. Handler módban minden kód privilegizált.

# ARM Cortex M3



# ARM Cortex M3

Felhasználói (user) mód

- nem férhető hozzá minden erőforrás pl. kivétel kezelő maszkok

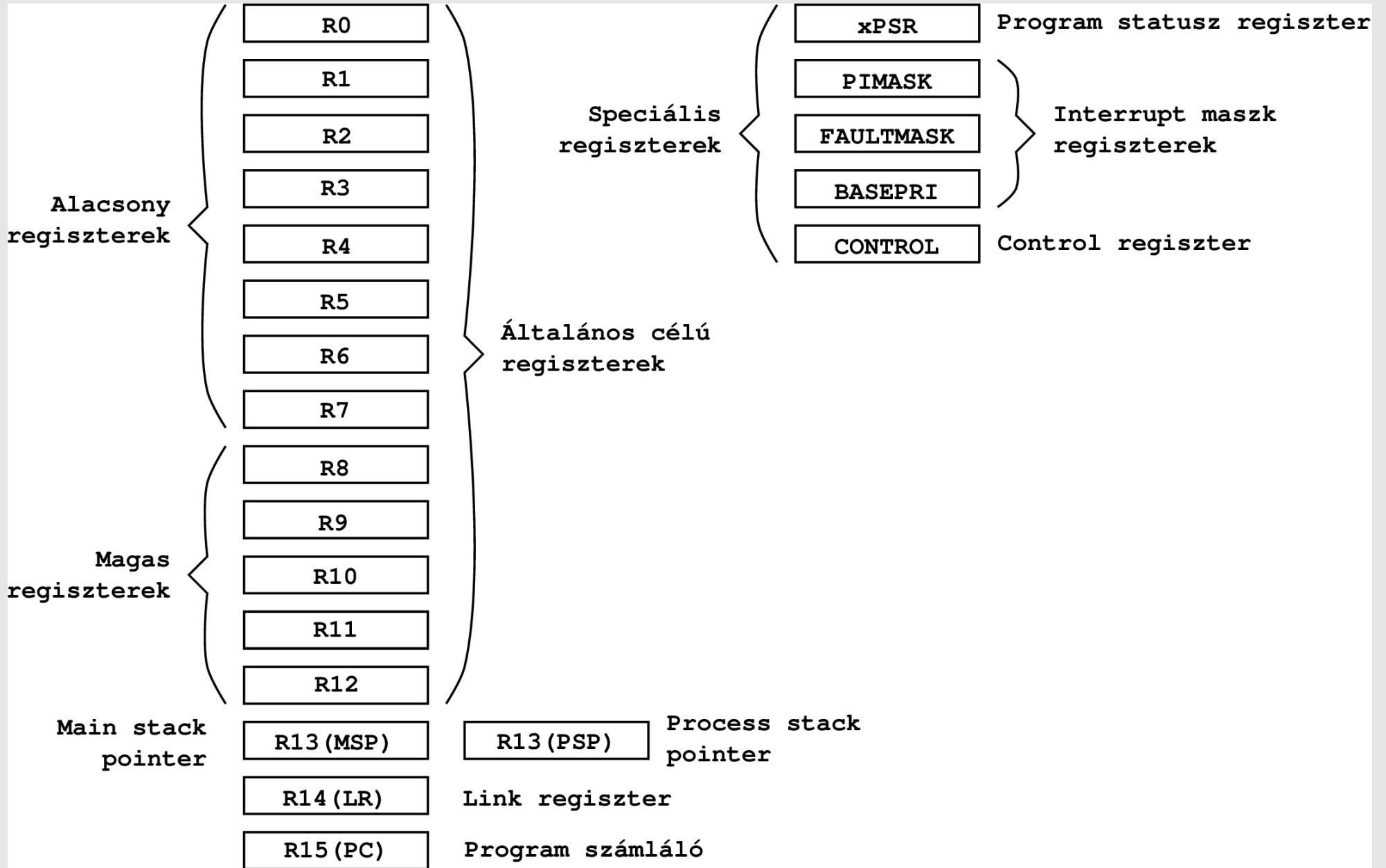
Privilegizált mód

- minden erőforrás hozzáférhető

Az üzemmódok a CONTROL regiszter egy bitjével állíthatók

## Regiszterek

# ARM Cortex M3





# ARM Cortex M3

R0-R7 regiszterek

- minden utasítás elér

R8-R12 regiszterek

- minden 32 bites utasítás elér
- nem ér el minden 16 bites utasítás

R13 stack pointer

- main SP (MSP) a handler mód csak ezt használhatja
- process SP (PSP) mindkét mód használhatja

# ARM Cortex M3

R14 regiszter (Link register)

- szubrutin hívás esetén ide kerül a visszatérési érték

R15 regiszter (PC)

- a program számláló

# ARM Cortex M3

## Speciális regiszterek

- használhatók egyben PSR és részletekben

	31	30	29	28	27	26:25	24	23:20	19:16	15:10	9	8	7	6	5	4:0
APSR	N	Z	C	V	Q											
IPSR												Exception number				
EPSR						ICI/IT	T				IC/IT					

- APSR (Application Program Status Register)
- IPSR (Interrupt Program status Register)
- EPSR (Execution Program Status Register)

# ARM Cortex M3

## Speciális regiszterek

- CONTROL regiszter

### 0. bit

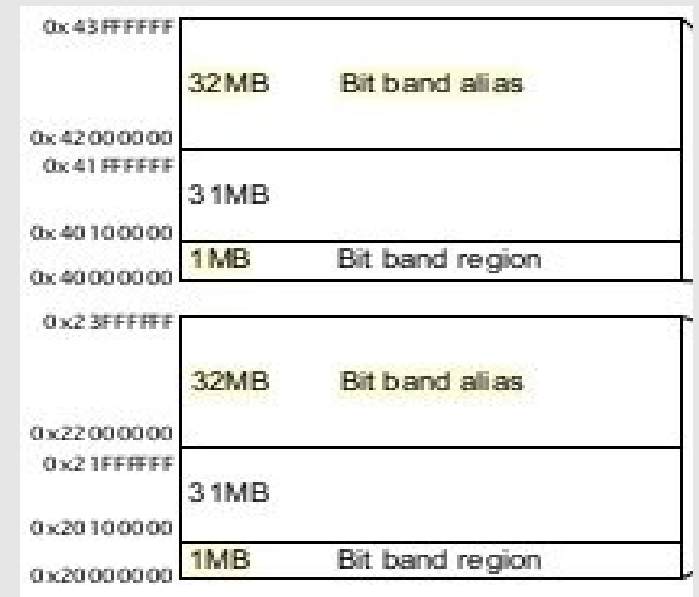
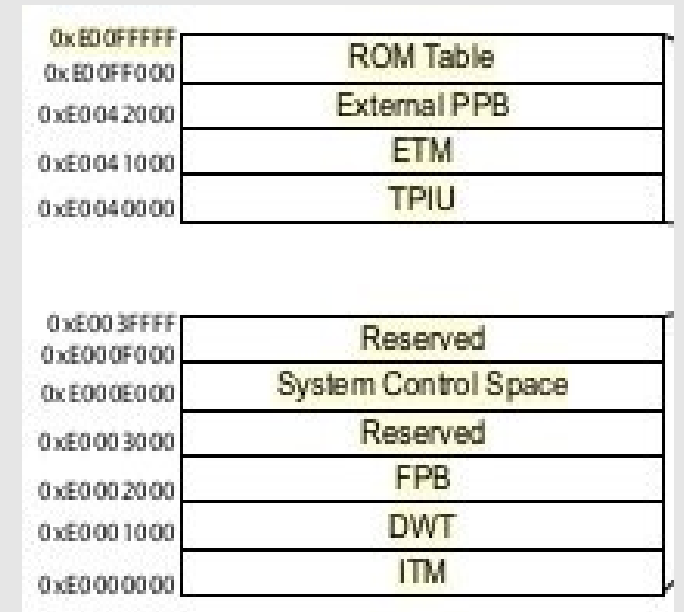
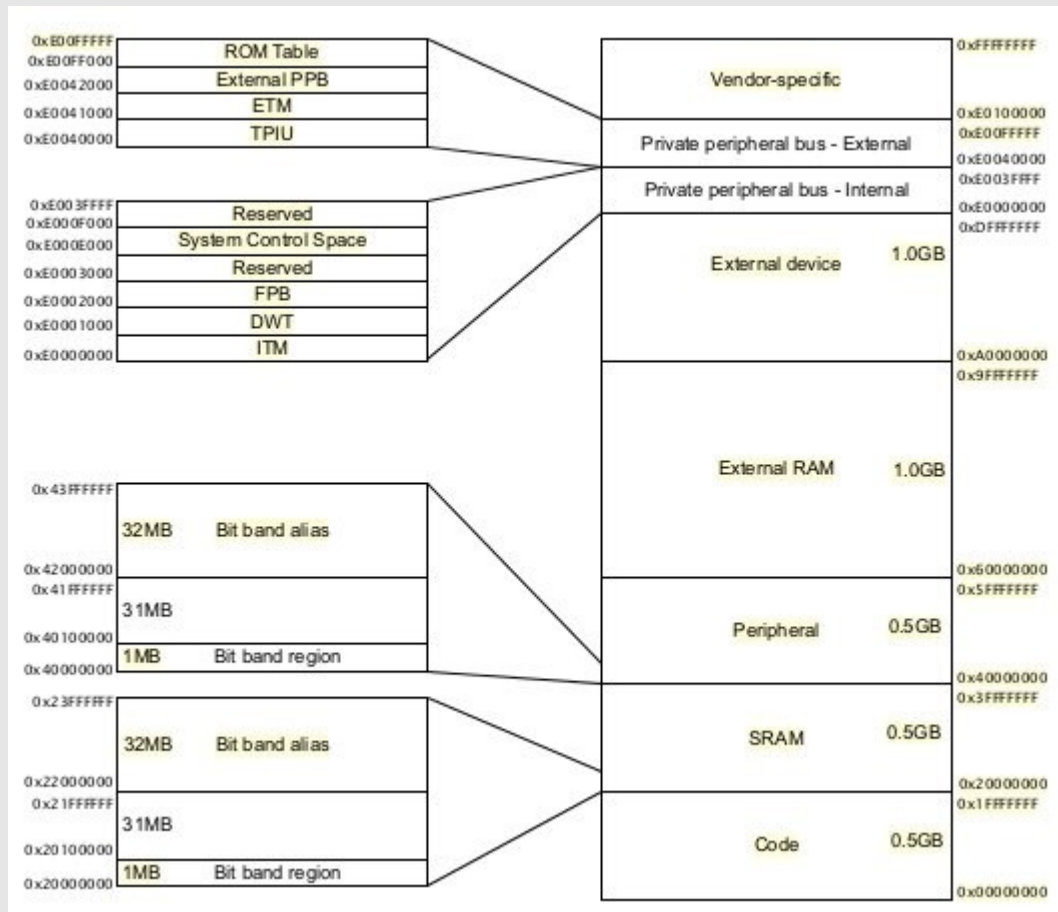
- 0 privilegizált mód
- 1 felhasználói mód

### 1. bit

- 0 MSP van használatban
- 1 PSP van használatban

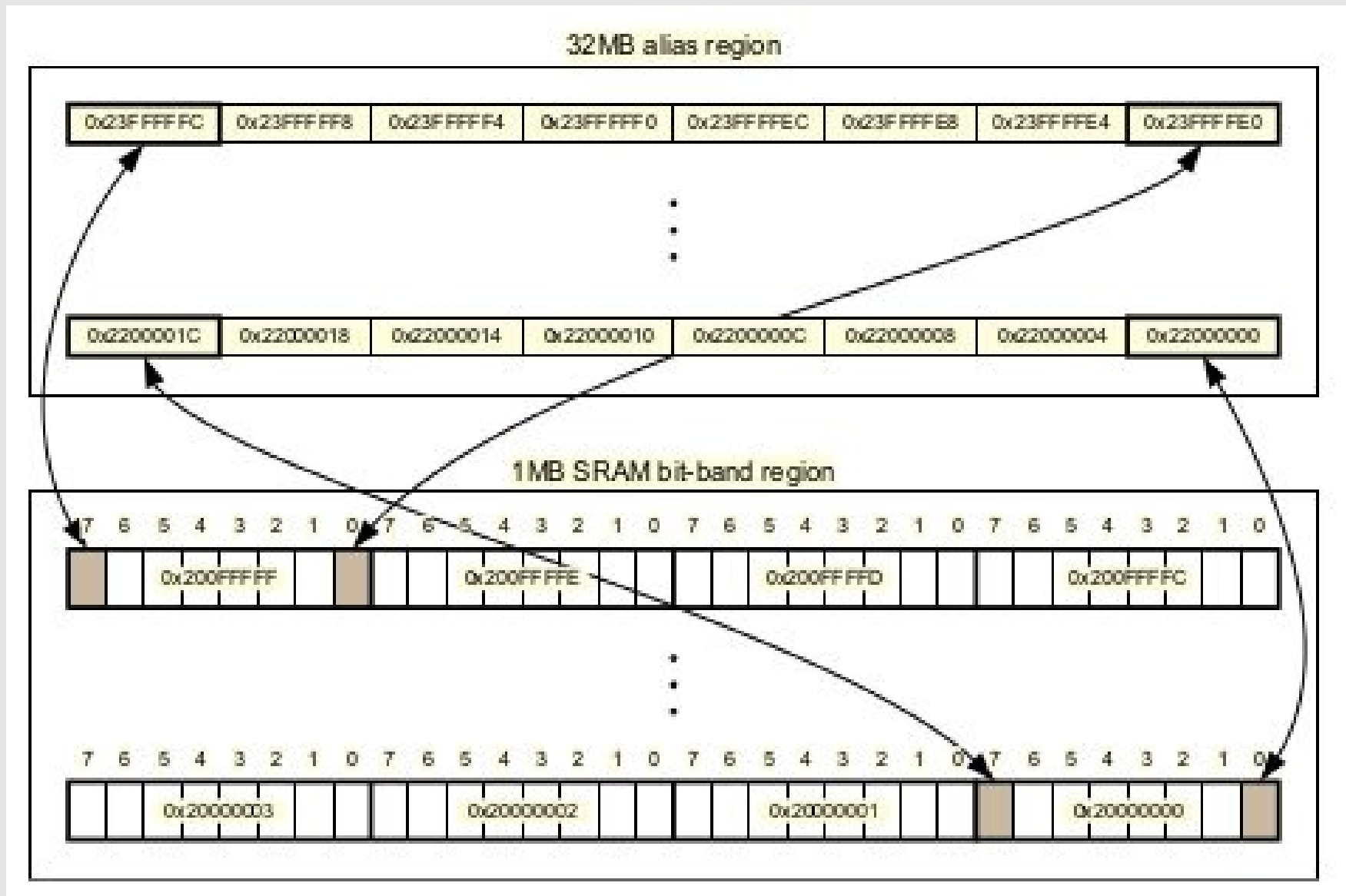
# ARM Cortex M3

## Memória térkép

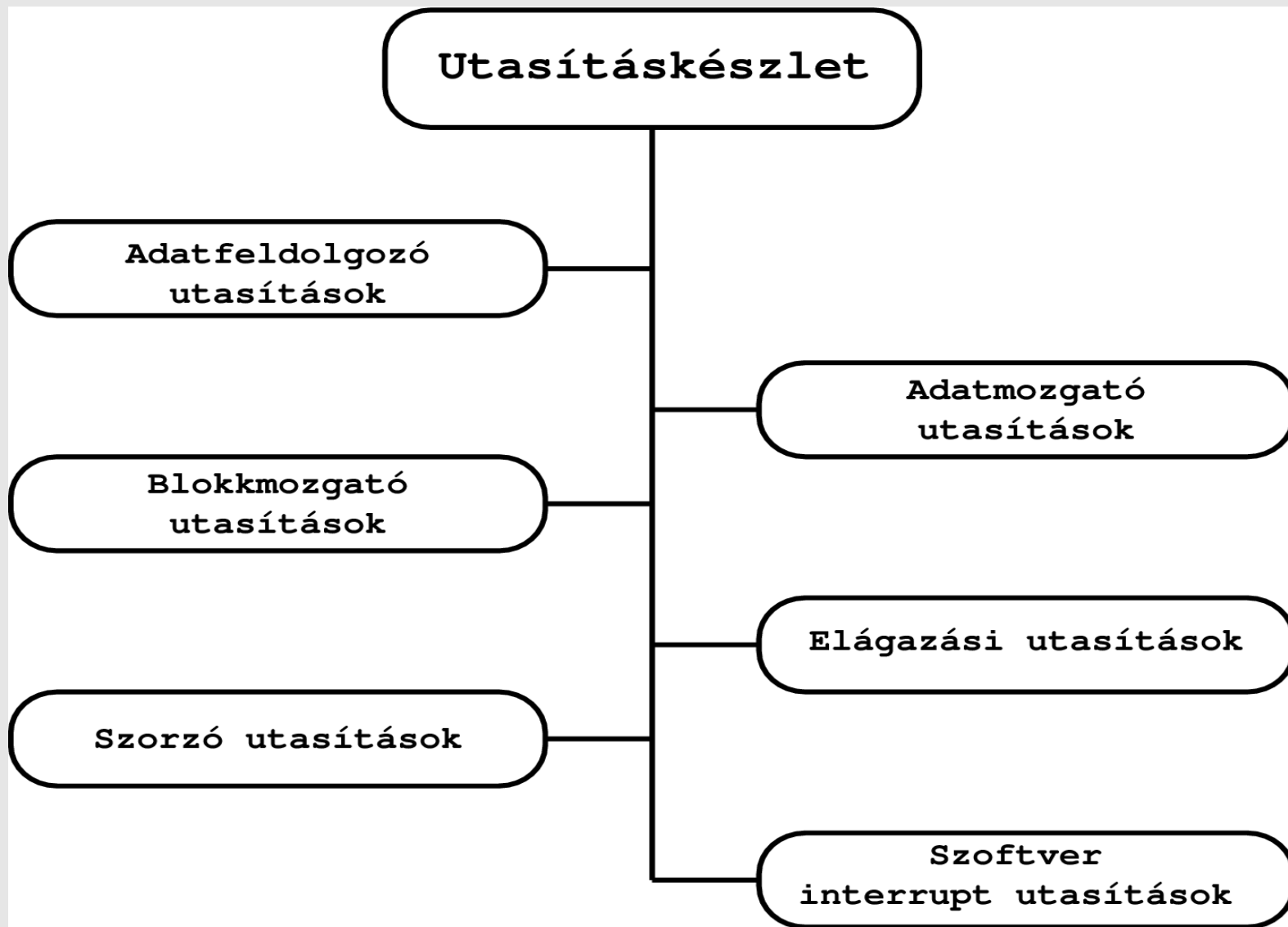


# ARM Cortex M3

## Bit-banding



# Utasításkészlet felépítése



# Thumb és Thumb2

## Thumb

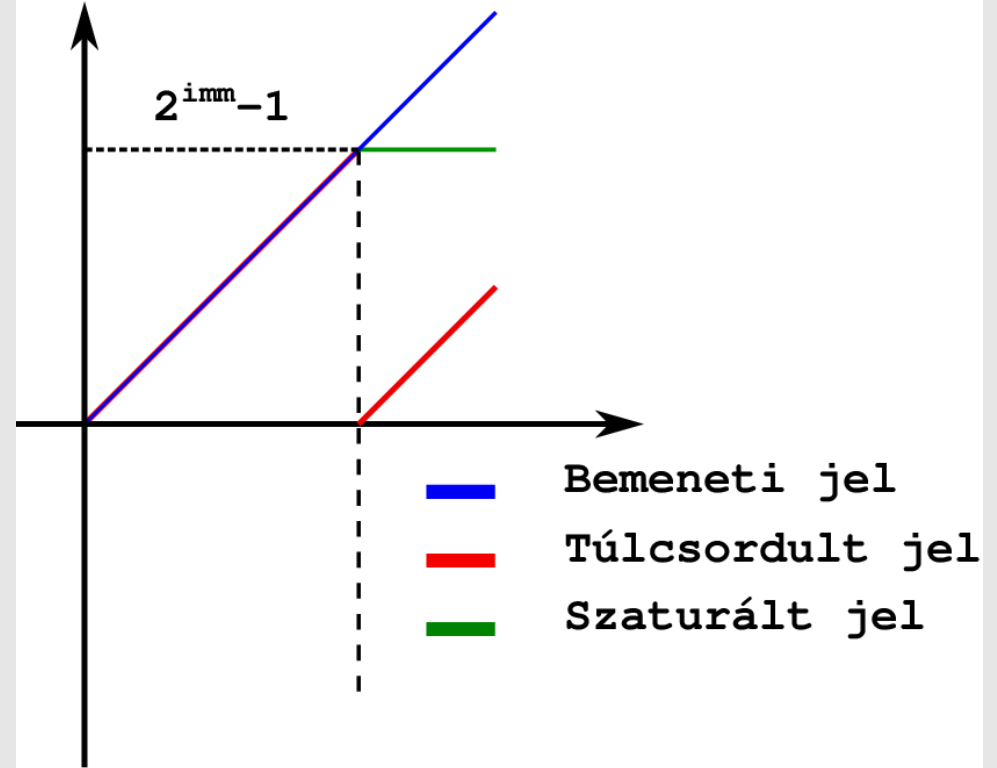
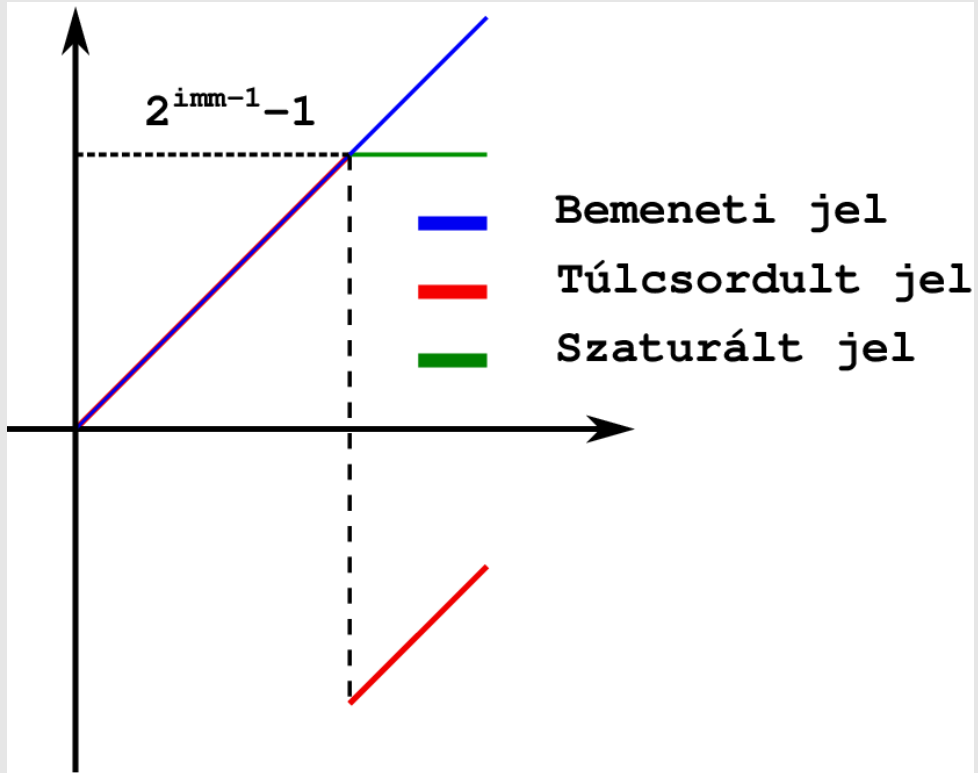
- a 32 bites ARM utasításkészlet részhalmaza
- minden utasítás 16 bites
- lefedi az eredeti ARM utasításkészlet 65%-át

## Thumb2

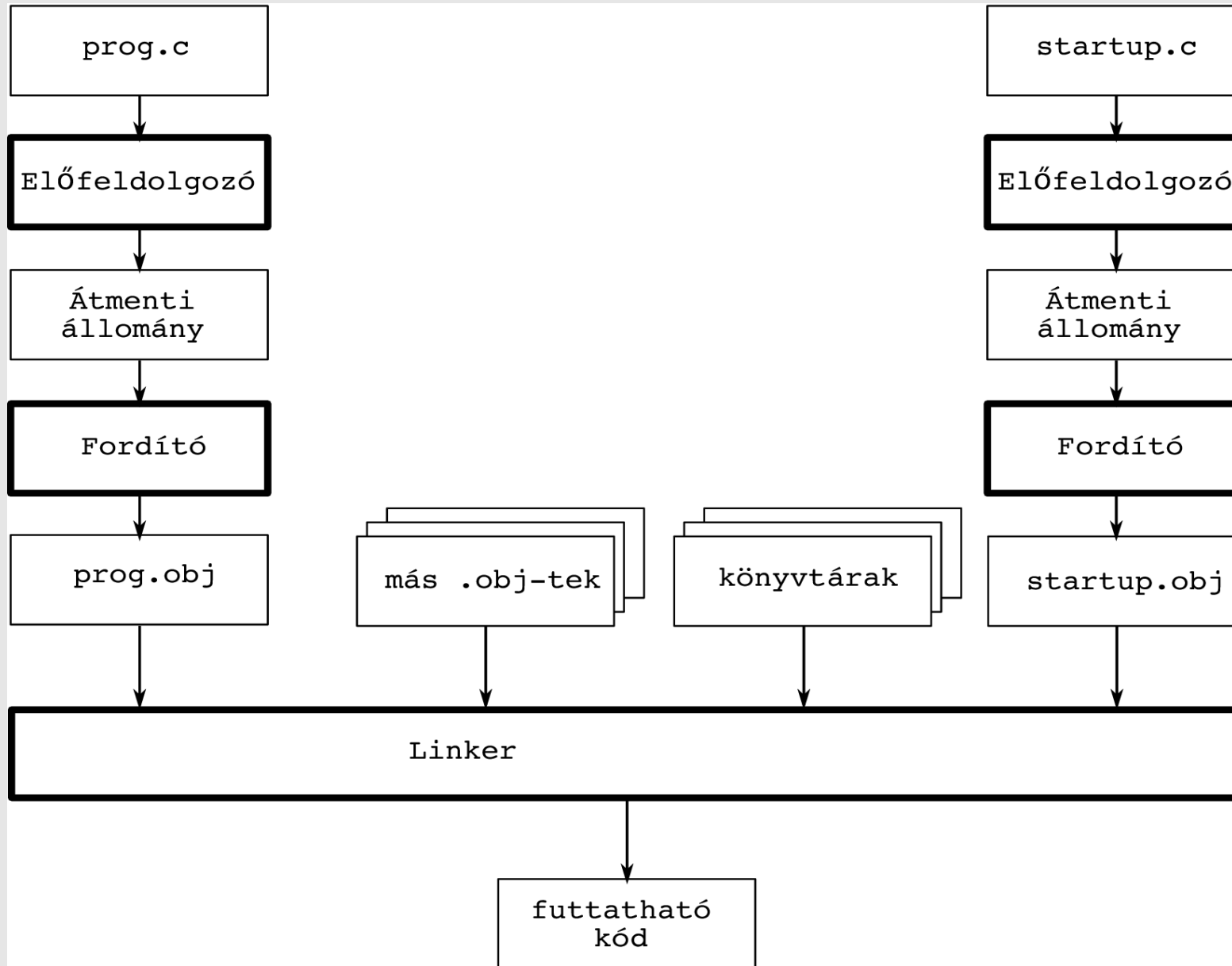
- szintén a a 32 bites ARM utasításkészlet részhalmaza
- lefedi a Thumb utasításokat és számos bővítése van:
  - támogatja a kivételkezelést
  - támogatja a koprocesszor elérést
  - DSP utasítások
  - feltételes “szerkezetek”



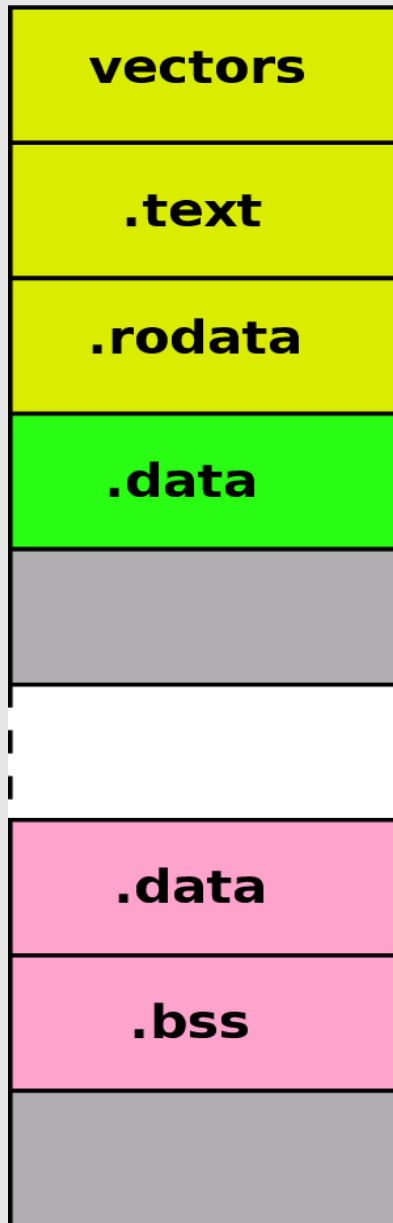
# Érdekes



# Fordítás és startup kód



# Szegmesek



 load és runtime

 load

 runtime

**vectors** IT vektorok

**.text** program szegmens

**.rodata** csak olvasható statikus adatok

**.data** inicializált statikus adatok

**.bss** nem inicializált statikus adatok

```
void ResetISR(void) ;  
static void NmiSR(void) ;  
static void FaultISR(void) ;  
static void IntDefaultHandler(void) ;  
extern int main(void) ;
```

```
extern unsigned long _etext ;  
extern unsigned long _data ;  
extern unsigned long _edata ;  
extern unsigned long _bss ;  
extern unsigned long _ebss ;
```

```

static unsigned long pulStack[64];

__attribute__((section(".isr_vector")))
void (* const g_pfnVectors[])(void) =
{
    (void (*)(void))((unsigned long)pulStack + sizeof(pulStack)),
                                // The initial stack pointer
    ResetISR,                   // The reset handler
    NmiISR,                     // The NMI handler
    FaultISR,                   // The hard fault handler
    IntDefaultHandler,          // The MPU fault handler
    IntDefaultHandler,          // The bus fault handler
    IntDefaultHandler,          // The usage fault handler
    0,                          // Reserved
    0,                          // Reserved
    0,                          // Reserved
    0,                          // Reserved
    IntDefaultHandler,          // SVCall handler
    IntDefaultHandler,          // Debug monitor handler
    0,                          // Reserved
    IntDefaultHandler,          // The PendSV handler
    IntDefaultHandler,          // The SysTick handler
    IntDefaultHandler,          // GPIO Port A
    IntDefaultHandler,          // GPIO Port B
    .
    .
    .

```

```

void ResetISR(void)
{
    unsigned long *pulSrc, *pulDest;

    pulSrc = &_etext;
    for(pulDest = &_data; pulDest < &_edata; )
    {
        *pulDest++ = *pulSrc++;
    }

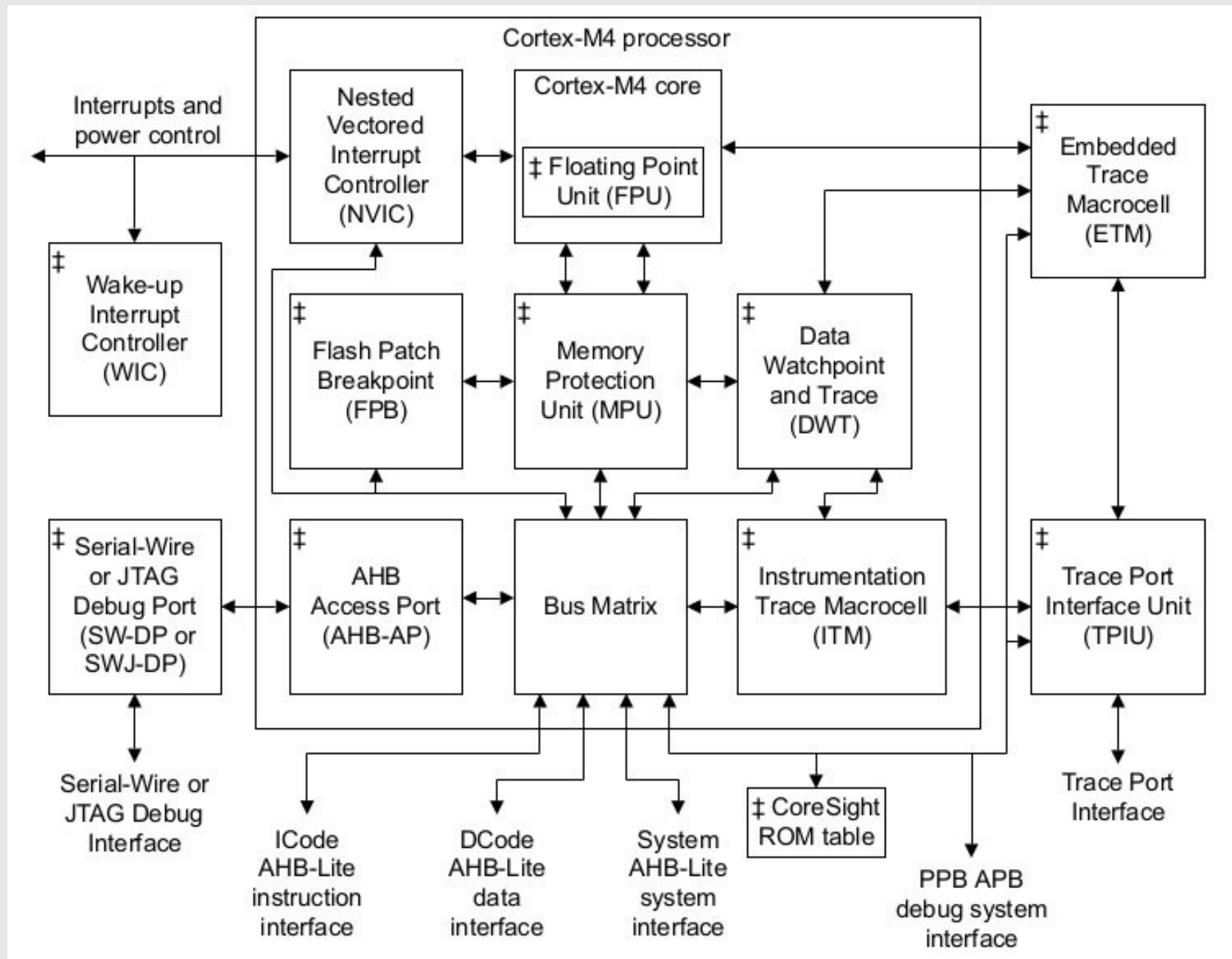
    __asm("    ldr      r0, =_bss\n"
        "    ldr      r1, =_ebss\n"
        "    mov      r2, #0\n"
        "    .thumb_func\n"
        "zero_loop:\n"
        "            cmp      r0, r1\n"
        "            it      lt\n"
        "            strlt    r2, [r0], #4\n"
        "            blt      zero_loop");
}

```

# ARM Cortex M4

- ARMv7 architektúra
- teljes Thum és Thumb2 utasításkészlet
- 1 ciklusos 32 bites hardver szorzás
- 2-12 ciklusos 32 bites hardver osztás
- 16/32 bites DSP MAC
- egyszeres pontosságú lebegőpontos bővítés
- 3 lépcsős utasítás pipeline ugrás figyeléssel
- 1-240 megszakítás
- 12 órajeles IT késleltetés
- 8 területre memória védelem
- beépített sleep mód
- 1.25 DMIPS/MHz

# ARM Cortex M4





# TIVA TM4C1294NCPDT

