Nagyteljesítményű mikrovezérlők

5. System Control block

Scherer Balázs

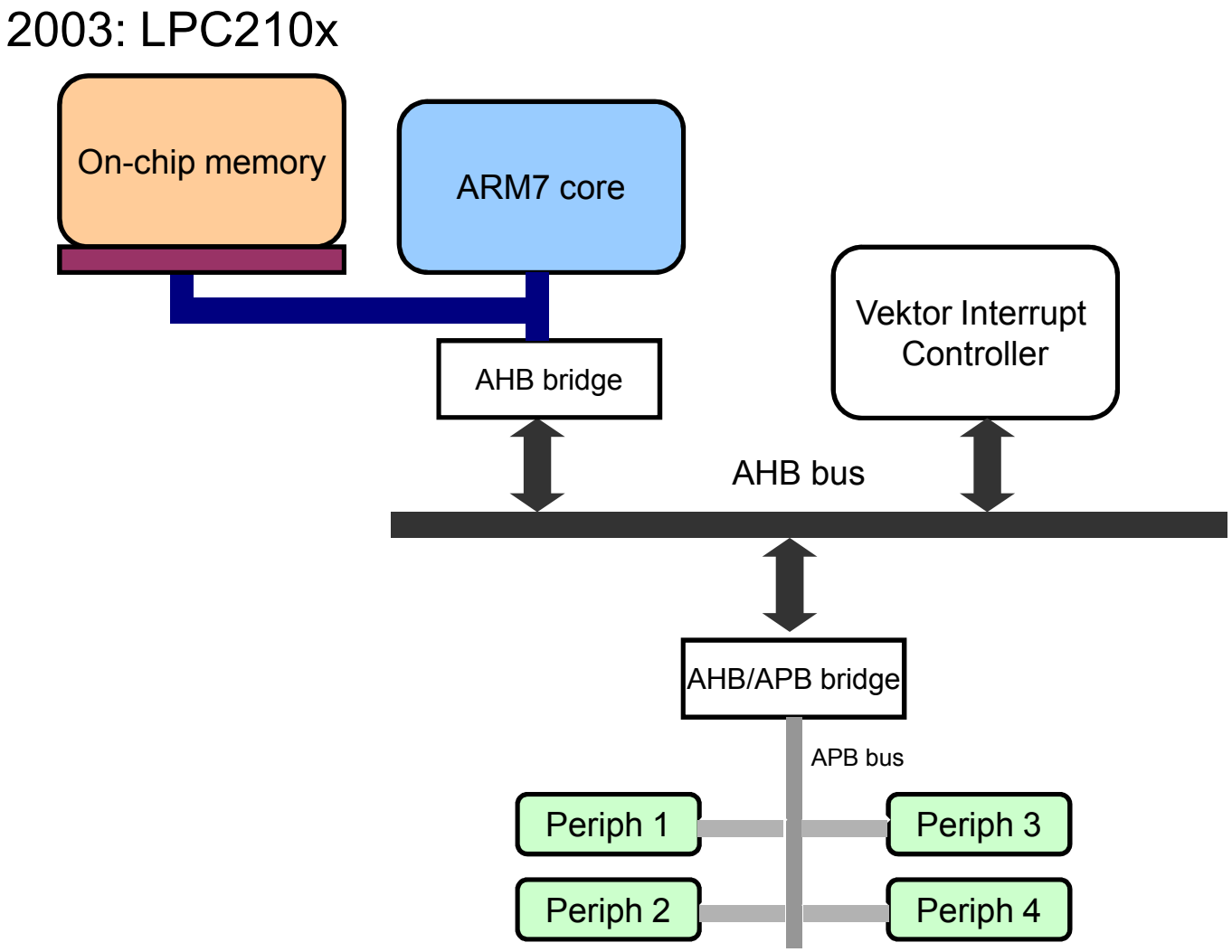
Méréstechnika és Információs Rendszerek Tanszék

Tartalom

- Az ARM7 magú vezérlők felépítésének fejlődése
- A legelterjedtebb Cortex sorozatok belső felépítése és felépítésük fejlődése
- System Control Block
  - A reset utáni elindulás folyamata
  - A Flash gyorsító modul
  - Órajel források és órajel elosztás
- CMSIS

Az ARM7 magú vezérlők felépítésének fejlődése

Az első ARM7 magú vezérlők belső felépítése



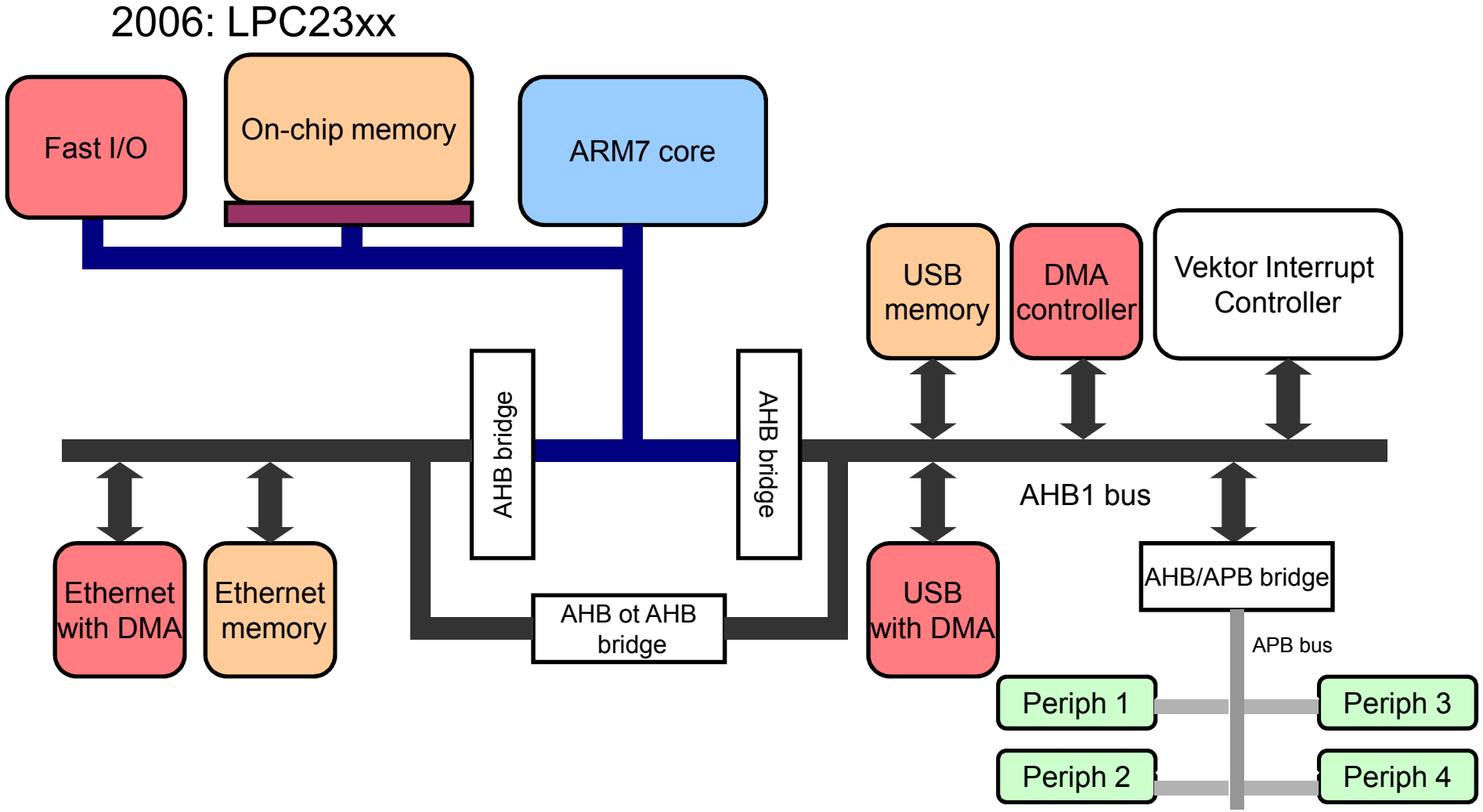
AHB vs APB

- Mindkettő az ARM Advanced Microcontroller Bus Architecture (AMBA) szabványhoz tartozik

- AHB Advanced High-performance Bus
- Van Pipelining
- Multiple master
- Burst-ös átvitel
- Full-duplex paralel komm.

- Advanced Peripheral Bus
- Nincs Pipelining
- Single master
- Kis interfész komplexitás
- Kis fogyasztás
- 32bites buszszélesség

Az utolsó szériás ARM7 magú vezérlő



A legelterjedtebb Cortex M3 sorozatok

belső felépítése és felépítésük fejlődése

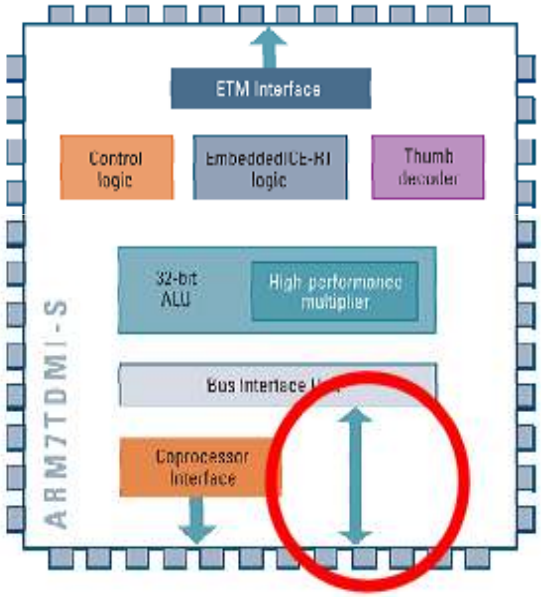
32 bites trendek 2003-2014

Flash [kbyte]

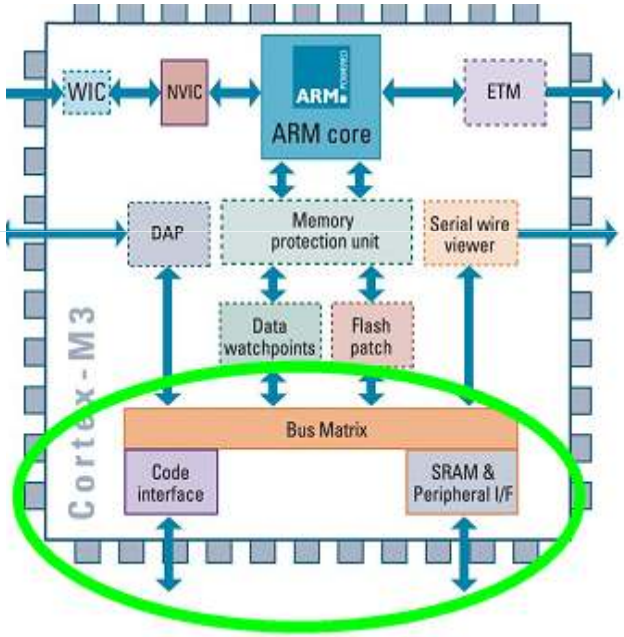
1024										
512										
256										
128										
64										
32										
16										
8										
4										
2										
1										
0,5										
	8	14-16	20	28-32-36	40-44-48	64	80-100	144	208	256

lábszám

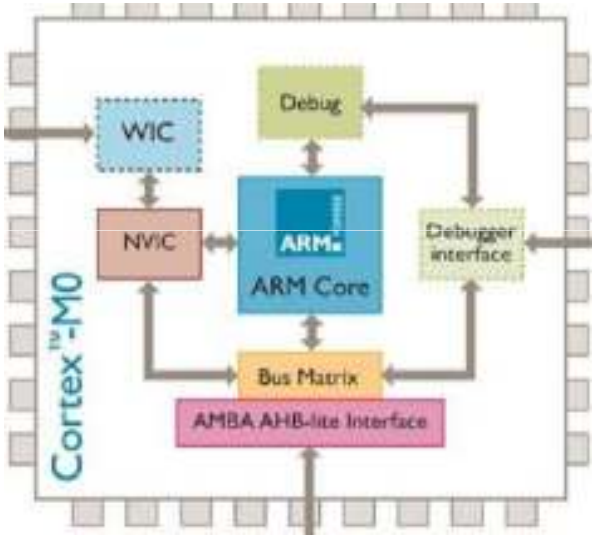
ARM7, Cortex M3, M0 összehasonlítás memória hozzáférés



ARM7TDMI



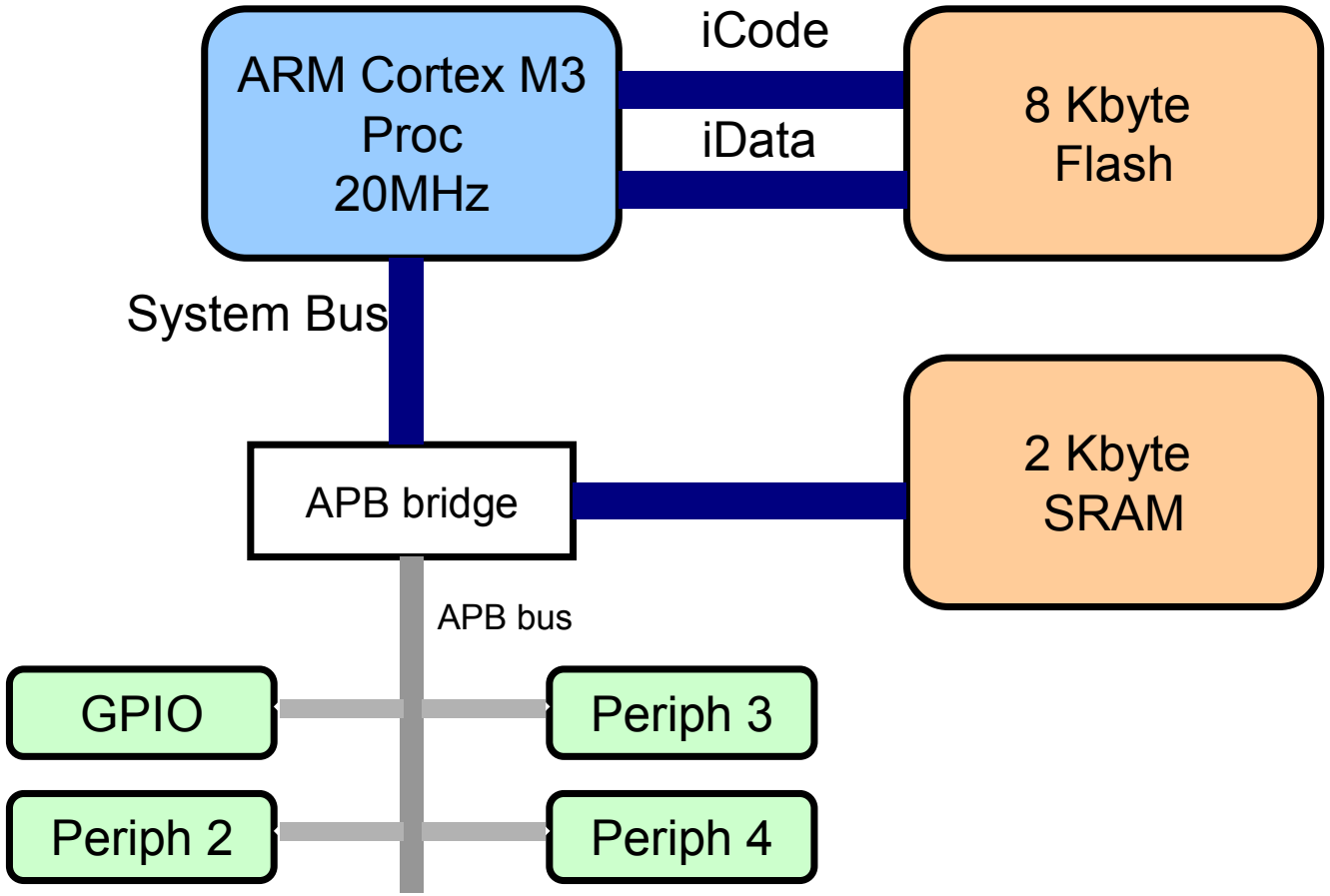
Cortex M3



Cortex M0

Első generációs Cortex M3

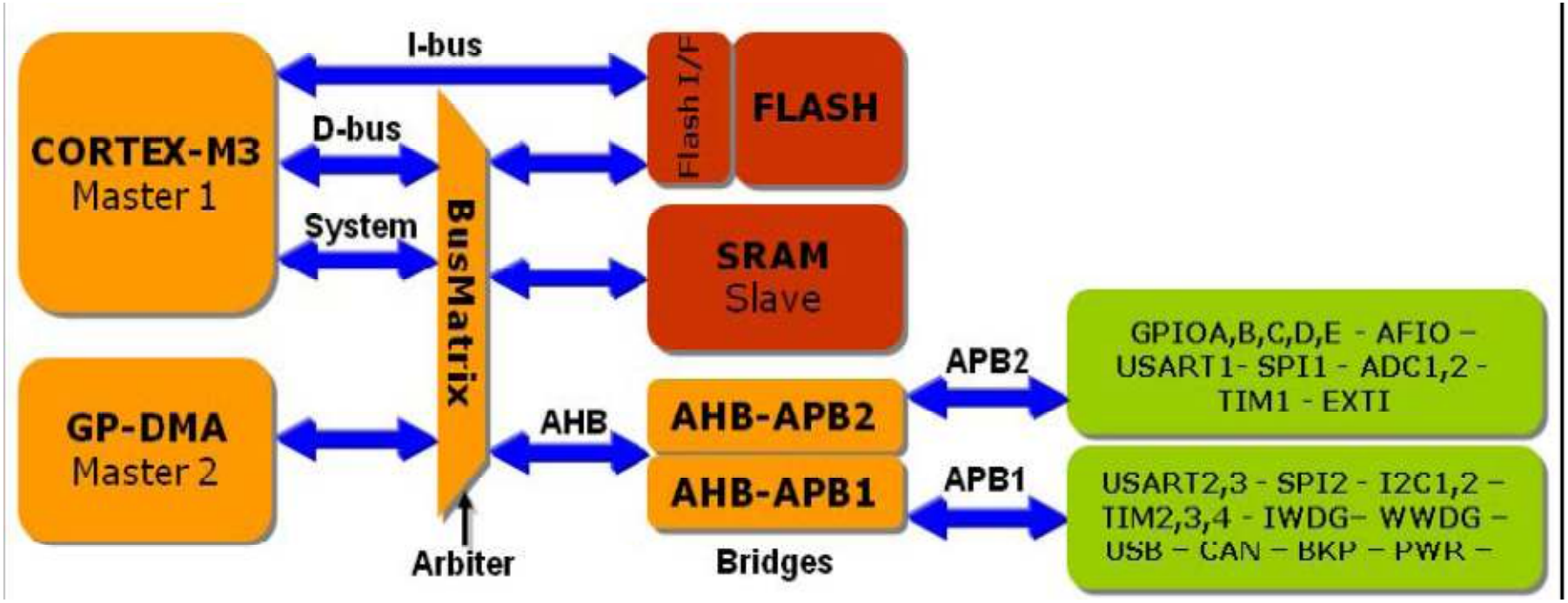
2006: Luminary LM3S102



Második generációs Cortex M3

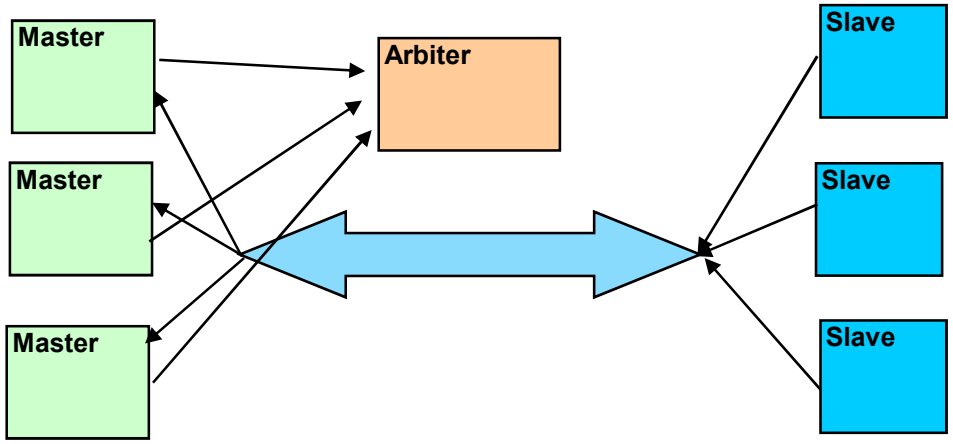
2007: STM32F103 (Max 72 MHz)

- APB1: max. 72MHz
- APB2: max. 36MHz

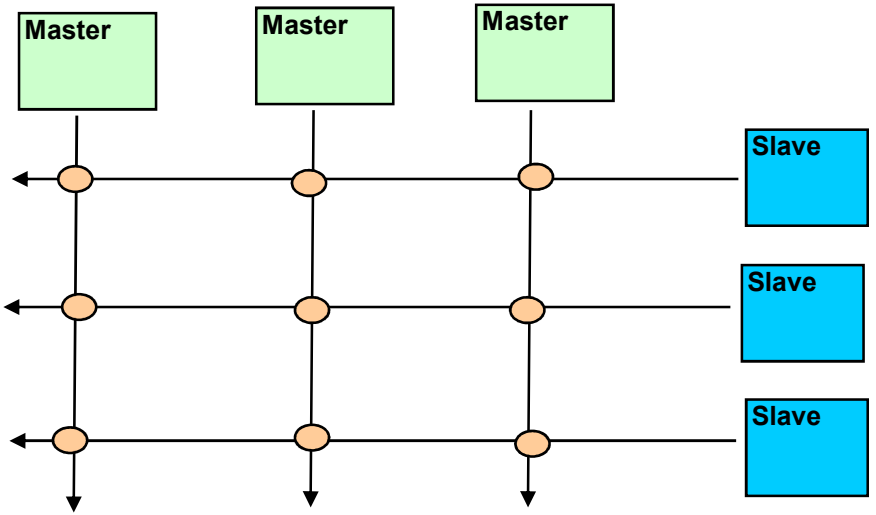


Több master a buszon

Megosztott AHB Busz

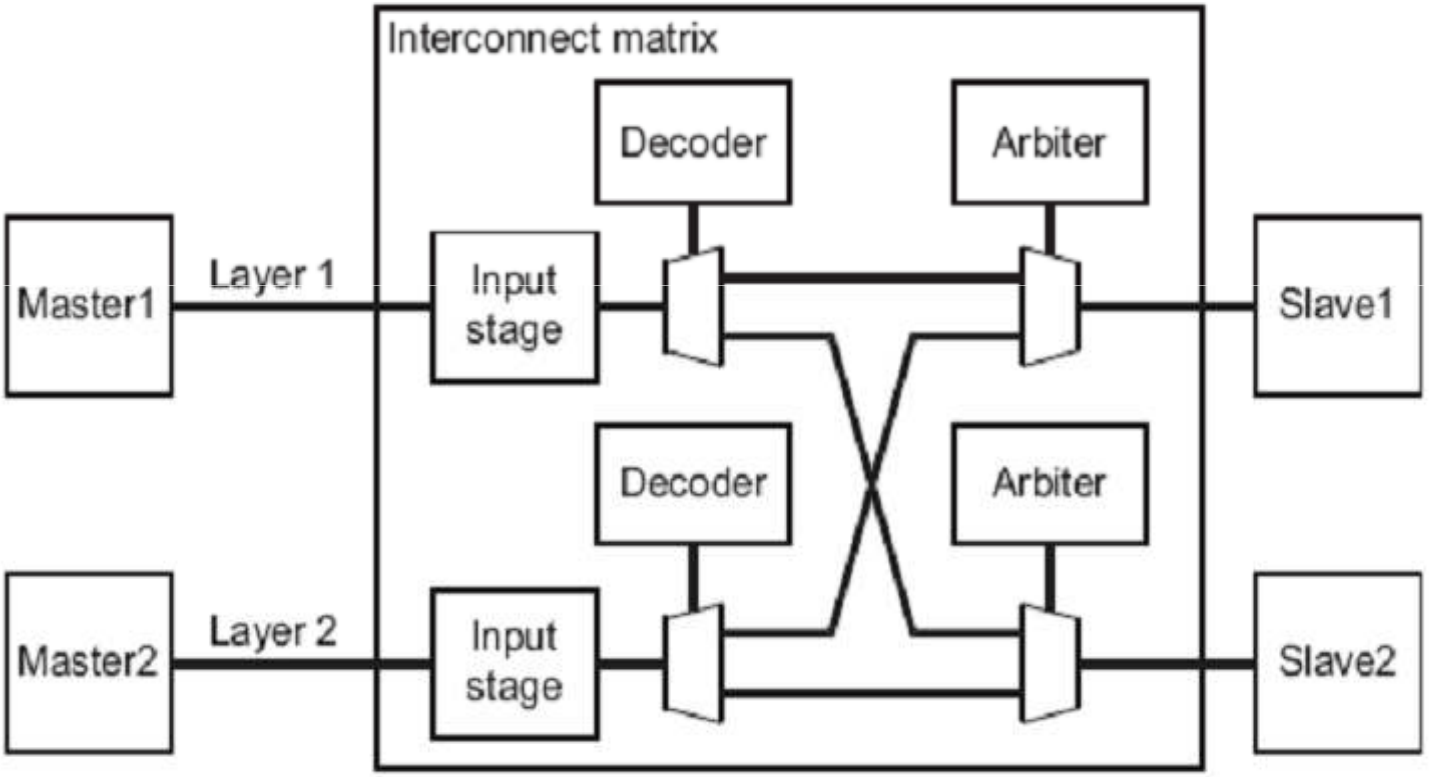


AHB Busz Matrix



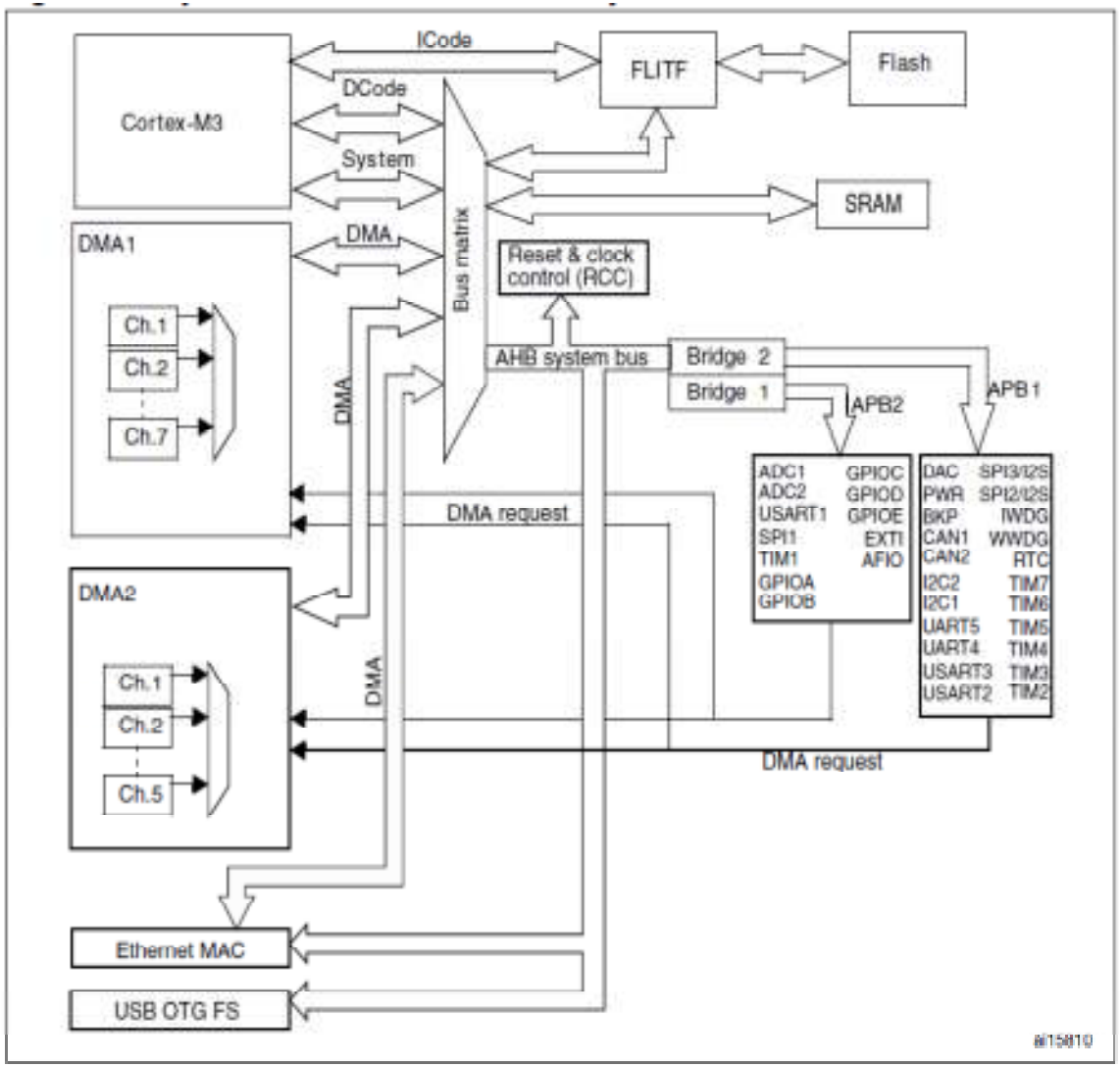
Mi történik a pontoknál

- Arbitráció: általában round-robin



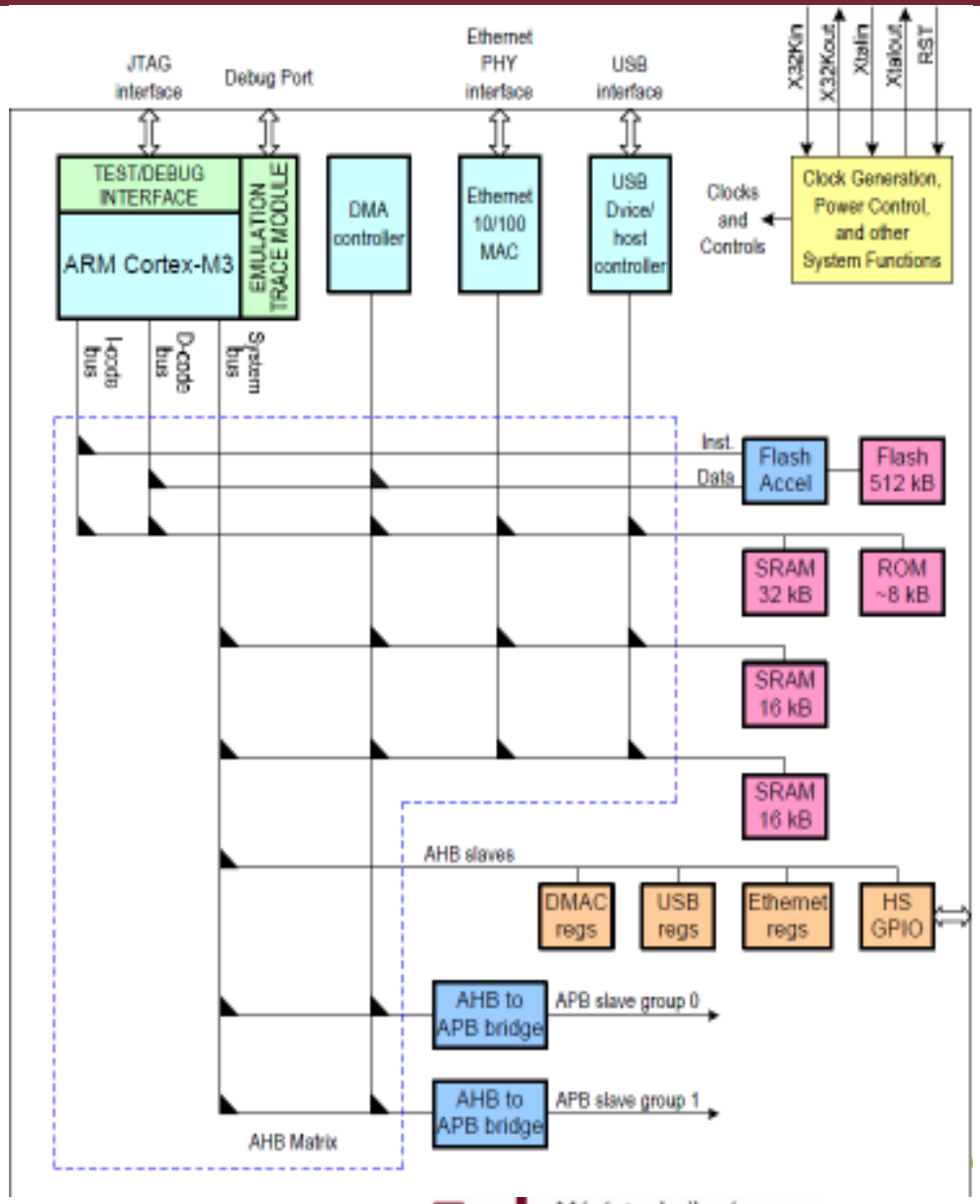
Harmadik generációs Cortex M3

2009: STM32F107 (Max 72 MHz)



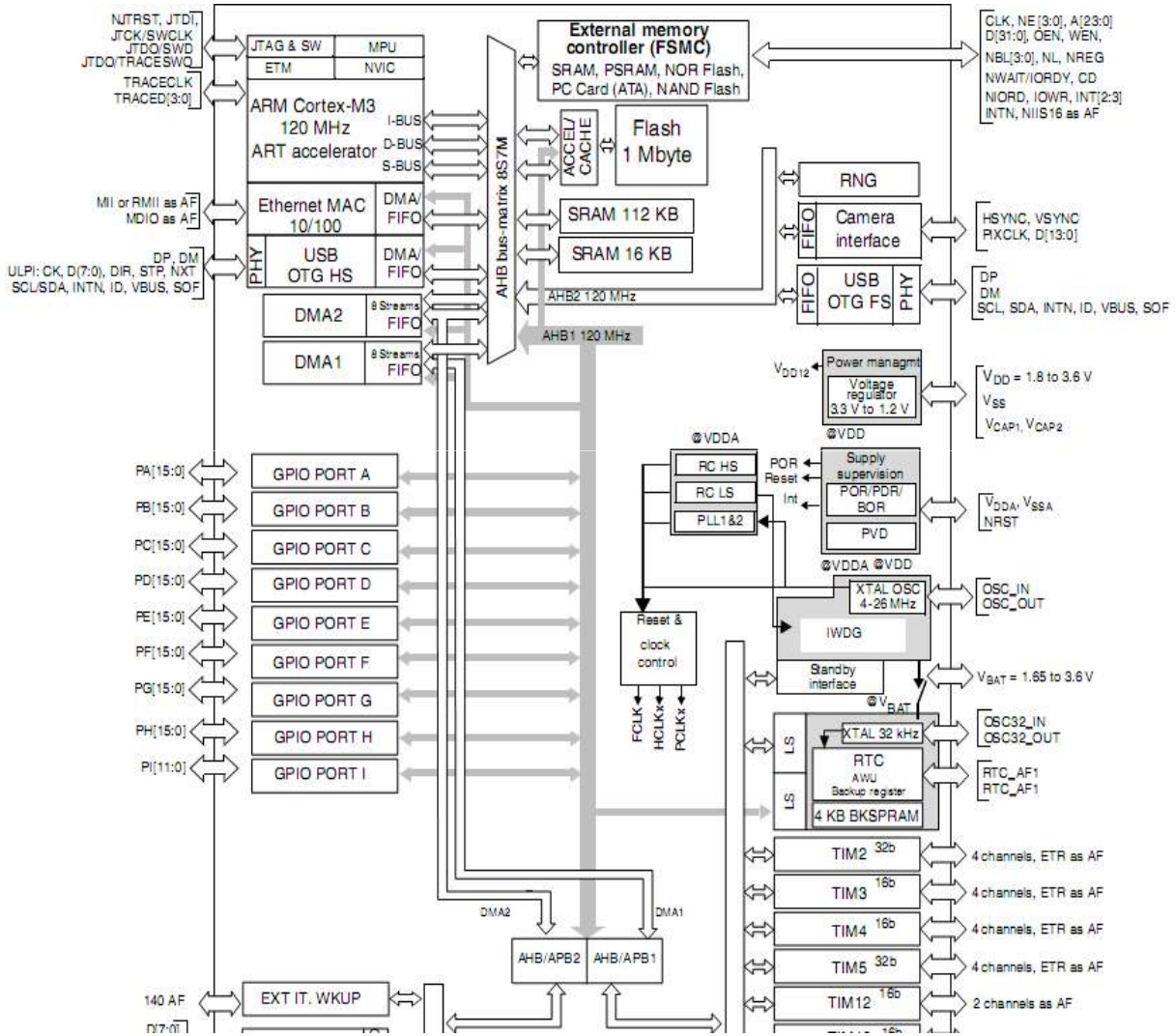
Harmadik generációs Cortex M3

- 2009: Az LPC1768 belső architektúrája



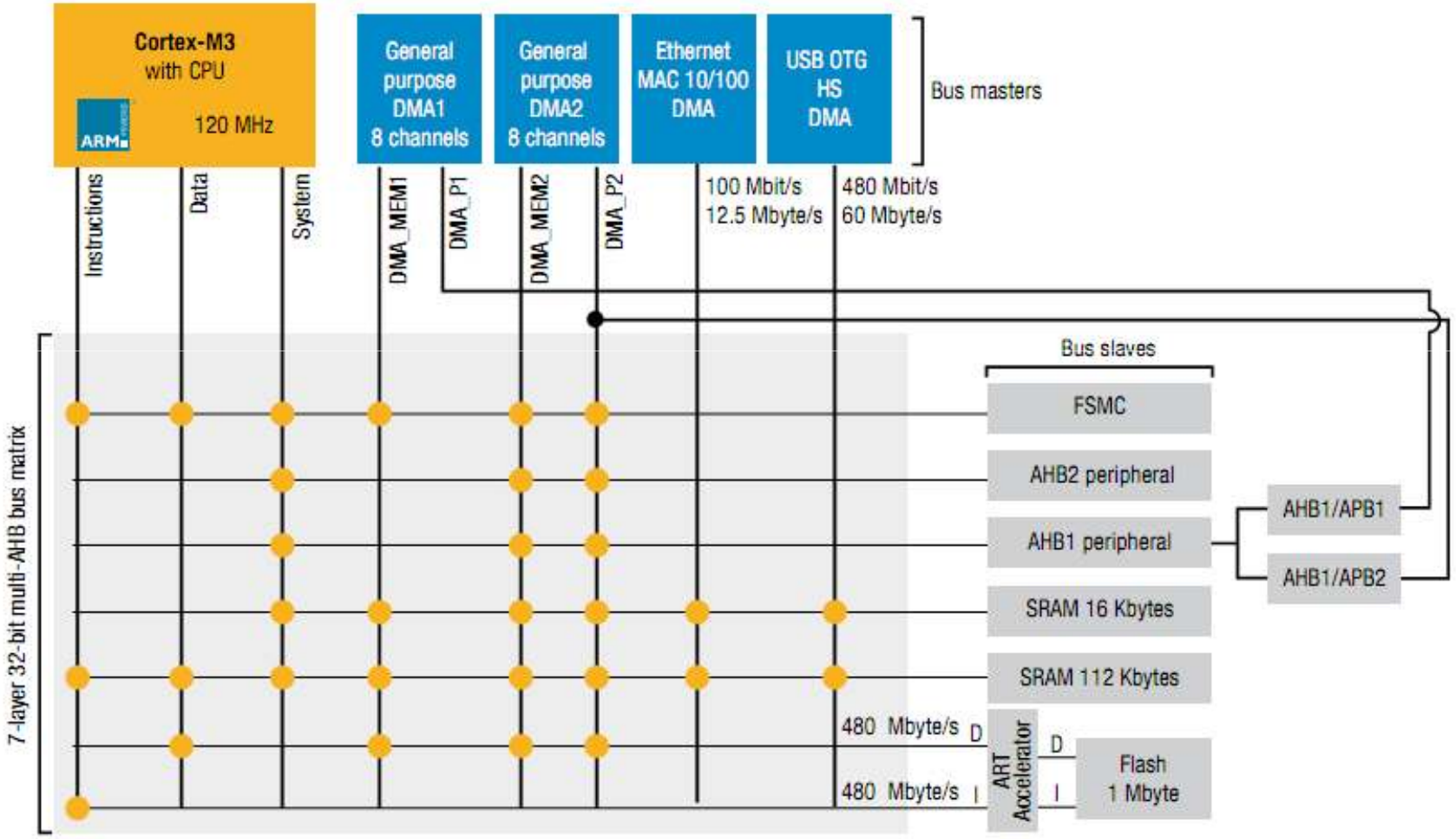
Negyedik Cortex M3 generáció

- 2010: Az STM32F2xx belső architektúrája



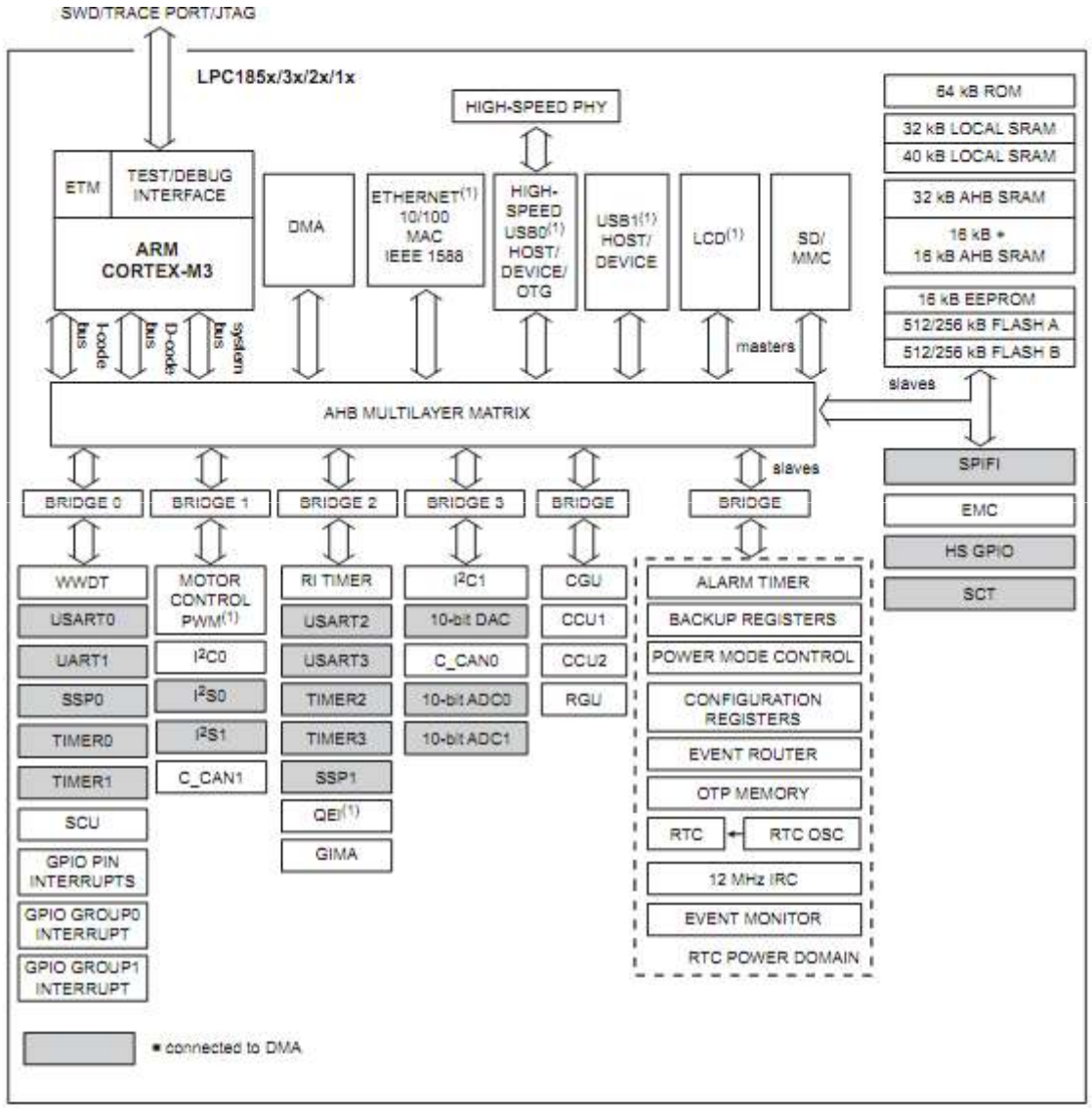
Negyedik Cortex M3 generáció

- 2010: Az STM32F2xx belső buszszerkezete



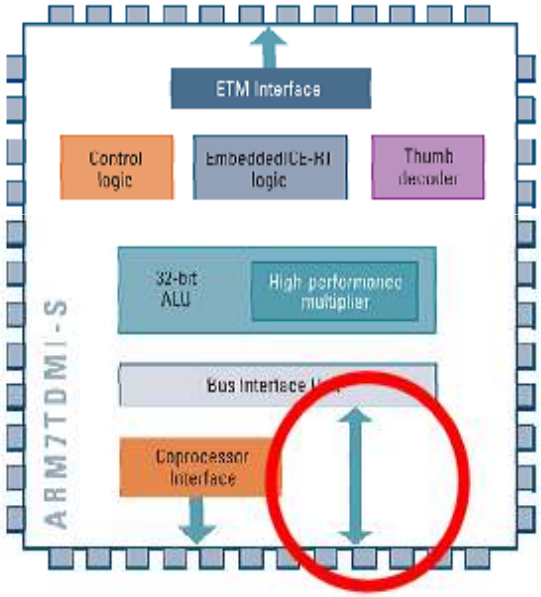
Negyedik Cortex M3 generáció

- LPC18xx

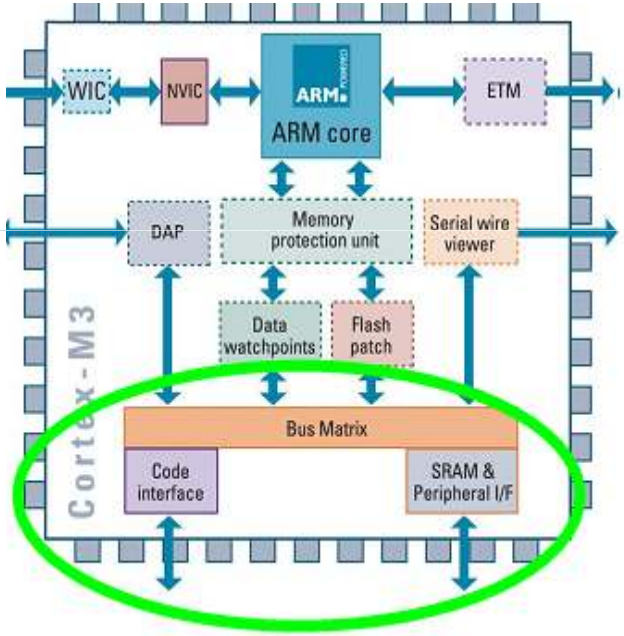


A legelterjedtebb Cortex M0 sorozatok belső felépítése

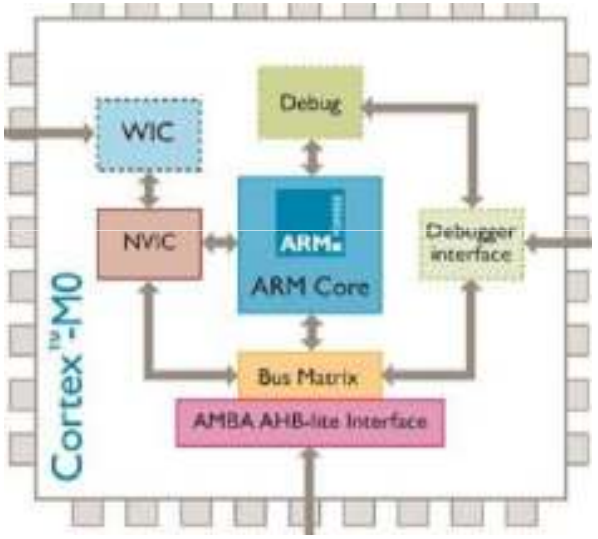
ARM7, Cortex M3, M0 összehasonlítás memória hozzáférés



ARM7TDMI



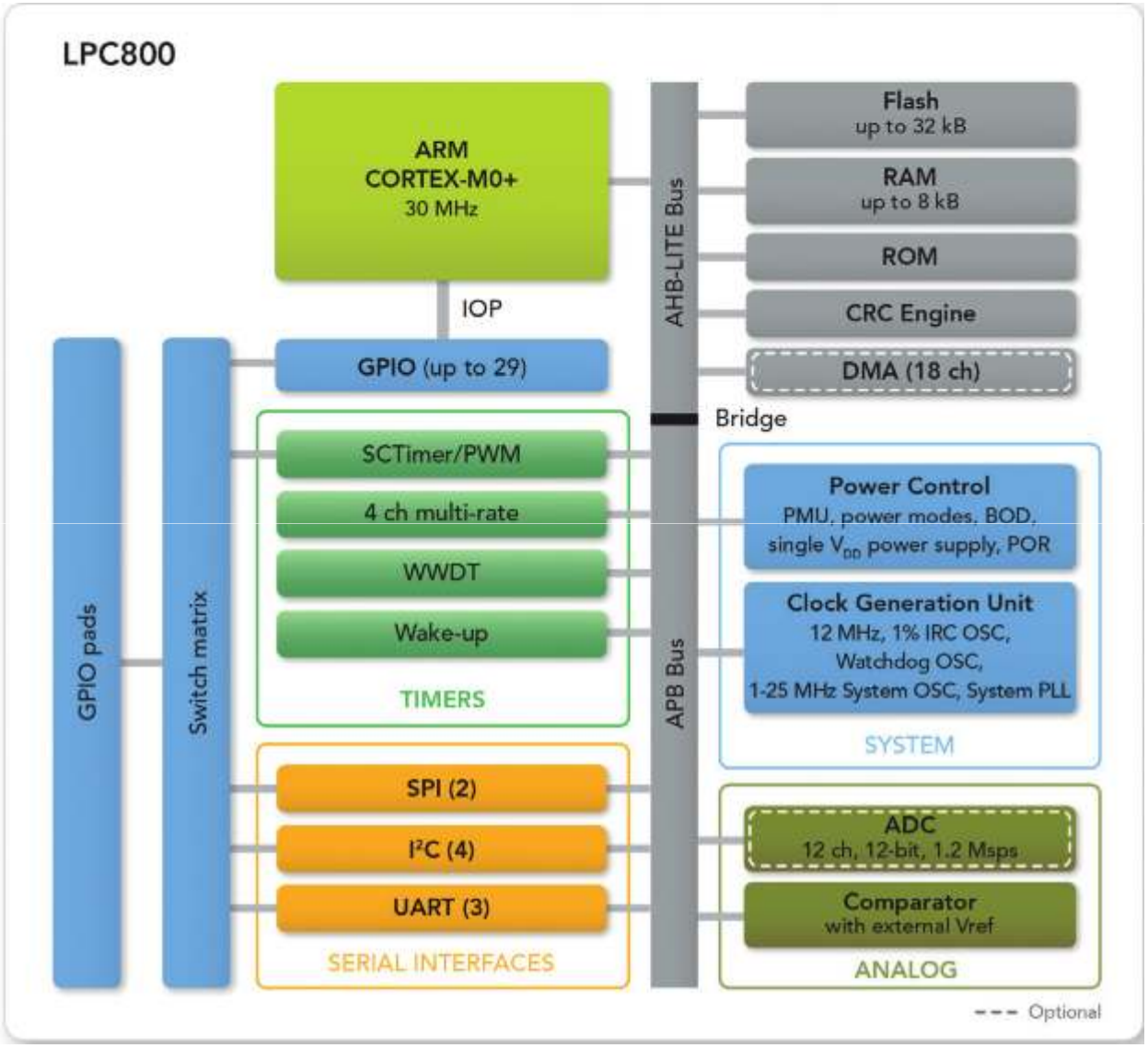
Cortex M3



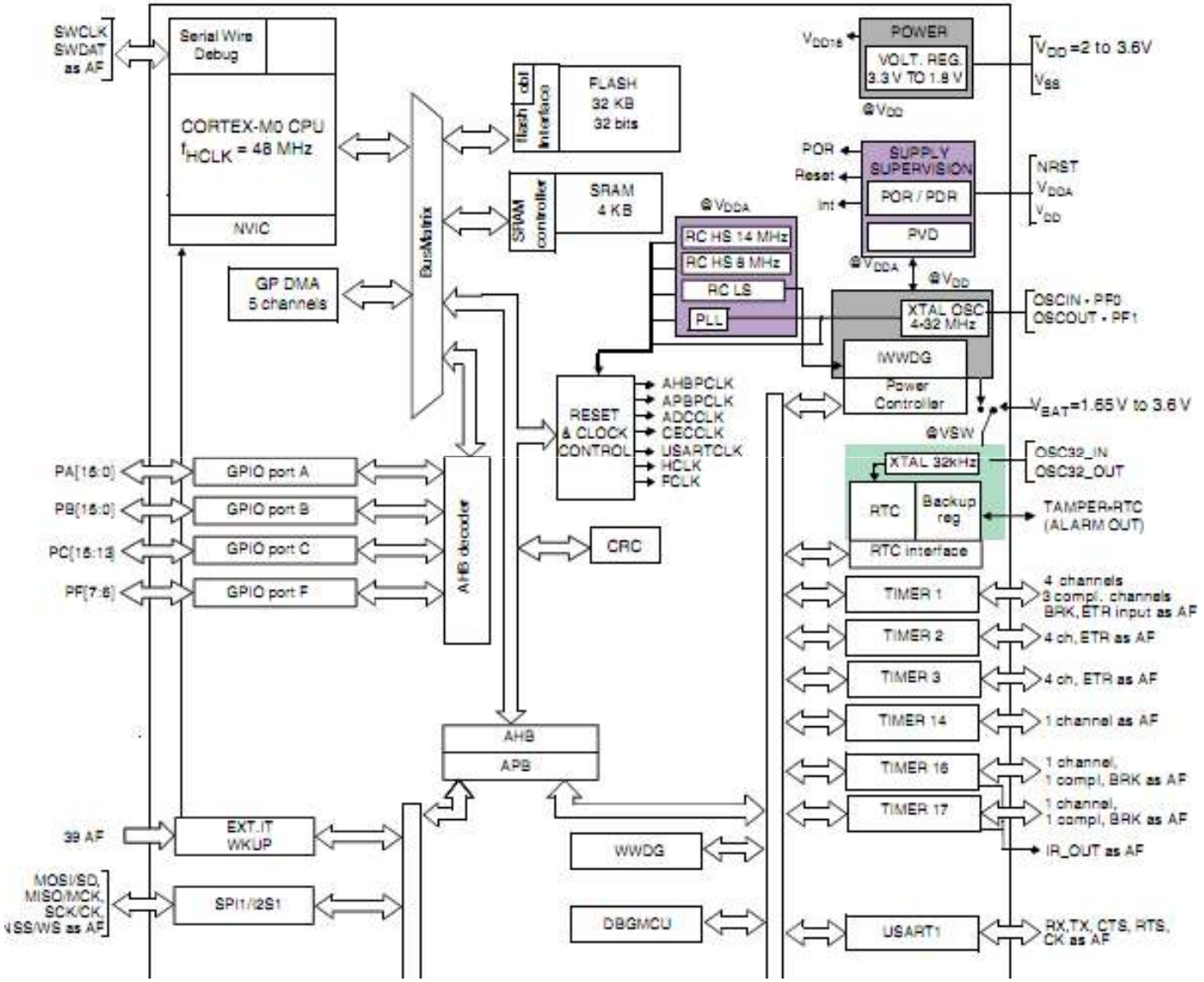
Cortex M0

LPC800

- AHB-Lite
  - Redukált bus specifikáció
  - 1 Master/layer
- Switch Matrix

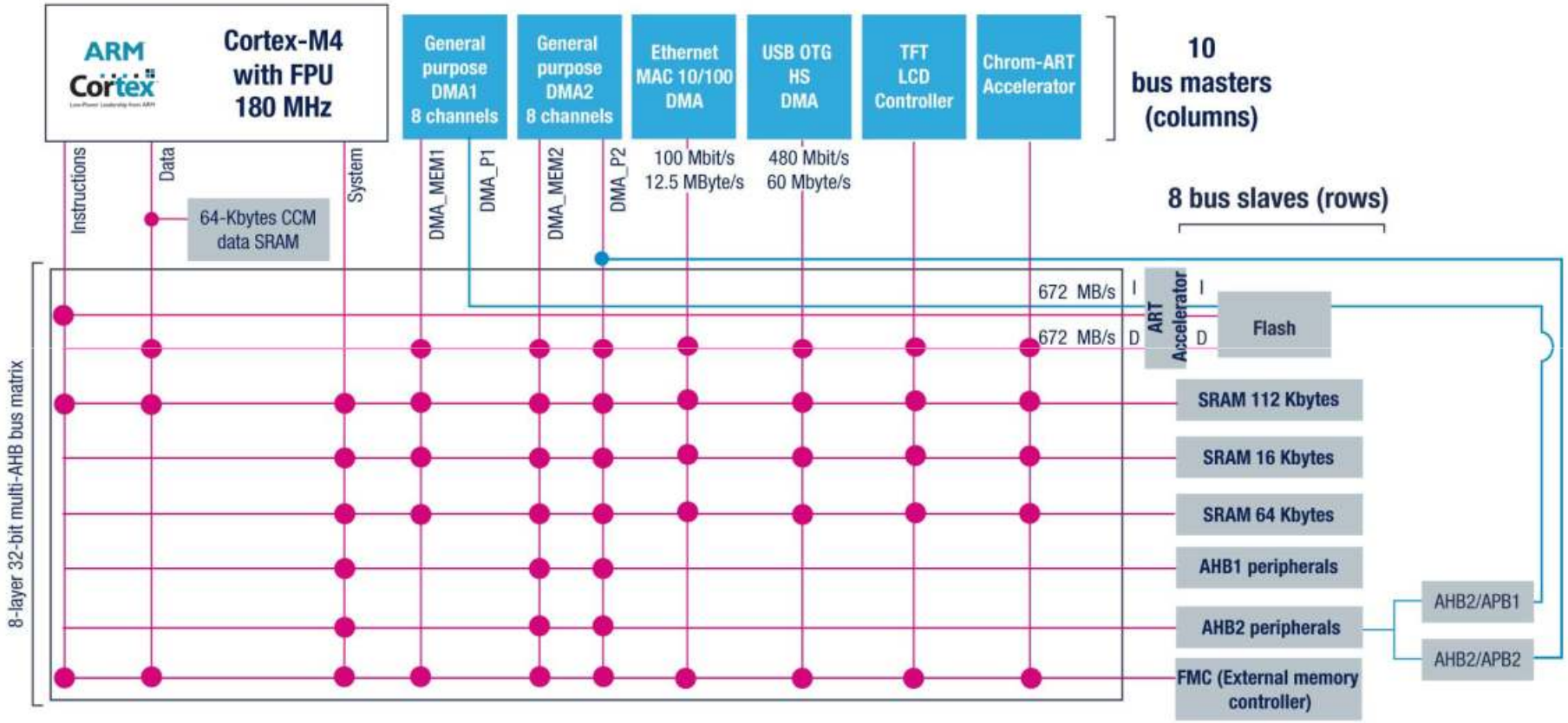


STMF0xx


SWCLK
SWDAT
as AF
Serial Wire Debug
CORTEX-M0 CPU
fHCLK = 48 MHz
NVIC
GP DMA
5 channels
BusMatrix
Flash ctrl Interface
FLASH
32 KB
32 bits
SRAM controller
SRAM
4 KB
@VDDA
RC HS 14 MHz
RC HS 8 MHz
RC LS
PLL
RESET & CLOCK CONTROL
AHBPCLK
APBPCLK
ADCCLK
CECCLK
USARTCLK
HCLK
FCLK
CRC
AHB decoder
PA[15:0]
PB[15:0]
PC[15:13]
PF[7:6]
GPIO port A
GPIO port B
GPIO port C
GPIO port F
AHB
APB
EXT.IT WKUP
39 AF
SPI1/I2S1
MOSI/SD, MISO/MCK, SCK/CK, NSS/WS as AF
WWDG
DBGMCU
VDD18
POWER
VOLT. REG. 3.3 V TO 1.8 V
@VDD
VDD=2 to 3.6V
VSS
POR
Reset
Int
SUPPLY SUPERVISION
POR / PDR
PVD
@VDDA
@VDD
NRST
VDDA
VDD
XTAL OSC 4-32 MHz
OSCIN - PF0
OSCOUT - PF1
IWWDG
Power Controller
VBAT=1.65 V to 3.6 V
@VSW
XTAL 32kHz
OSC32_IN
OSC32_OUT
RTC
Backup reg
RTC interface
TAMPER-RTC (ALARM OUT)
TIMER 1
4 channels
3 compl. channels
BRK, ETR input as AF
TIMER 2
4 ch, ETR as AF
TIMER 3
4 ch, ETR as AF
TIMER 14
1 channel as AF
TIMER 16
1 channel, 1 compl, BRK as AF
TIMER 17
1 channel, 1 compl, BRK as AF
IR_OUT as AF
USART1
RX,TX, CTS, RTS, CK as AF

A legelterjedtebb Cortex M4 sorozatok belső felépítése

STMF4xxx

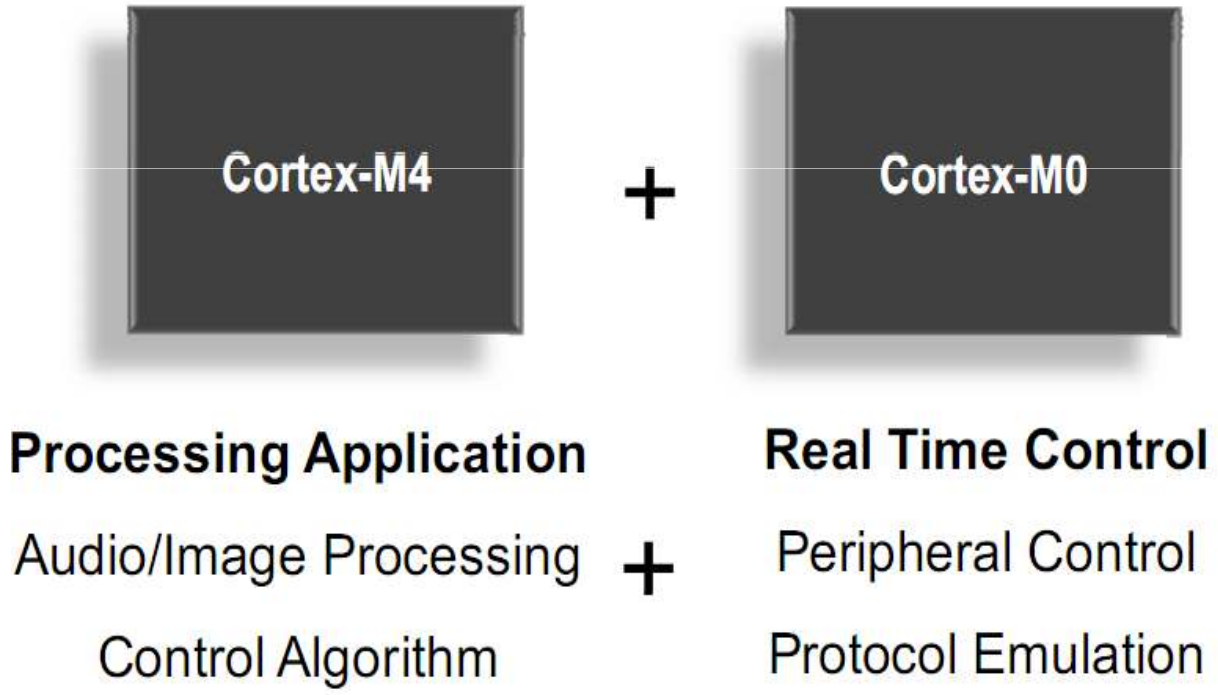

ARM Cortex
Cortex-M4 with FPU 180 MHz
General purpose DMA1 8 channels
General purpose DMA2 8 channels
Ethernet MAC 10/100 DMA
USB OTG HS DMA
TFT LCD Controller
Chrom-ART Accelerator
10 bus masters (columns)
Instructions
Data
System
DMA_MEM1
DMA_P1
DMA_MEM2
DMA_P2
100 Mbit/s 12.5 MByte/s
480 Mbit/s 60 Mbyte/s
8 bus slaves (rows)
64-Kbytes CCM data SRAM
672 MB/s
672 MB/s
ART Accelerator
Flash
SRAM 112 Kbytes
SRAM 16 Kbytes
SRAM 64 Kbytes
AHB1 peripherals
AHB2 peripherals
AHB2/APB1
AHB2/APB2
FMC (External memory controller)
8-layer 32-bit multi-AHB bus matrix

Az LPC4300 család

- Cortex-M4 alaú Digital Signal Controller
- Cortex-M0 Alrendszer a periféria funkciókra
- max. 1 MB Flash
  - Két bankos Flash
- max. 200 kbyte SRAM
- High speed USB
- Pin kompatibilis az M3 sorozattal
- További tulasjdonságok
  - 10/100 Ethernet MAC
  - LCD panel controller (max. 1024H × 768V)
  - 2x 10-bit ADCs és 10-bit DAC at 400ksps
  - 8 csatornás DMA vezérlő
  - Motor Control PWM, Quadrature Encoder
  - 4x UARTs, 2x I2C, I2S, CAN 2.0B, 2x SSP/SPI

Cortex M4, Cortex M0 együtt

- Szeparálható a feldolgozás és a Real-Time vezérlés
- Külön NVIC
- Osztott memóriarendszeren keresztüli kommunikáció

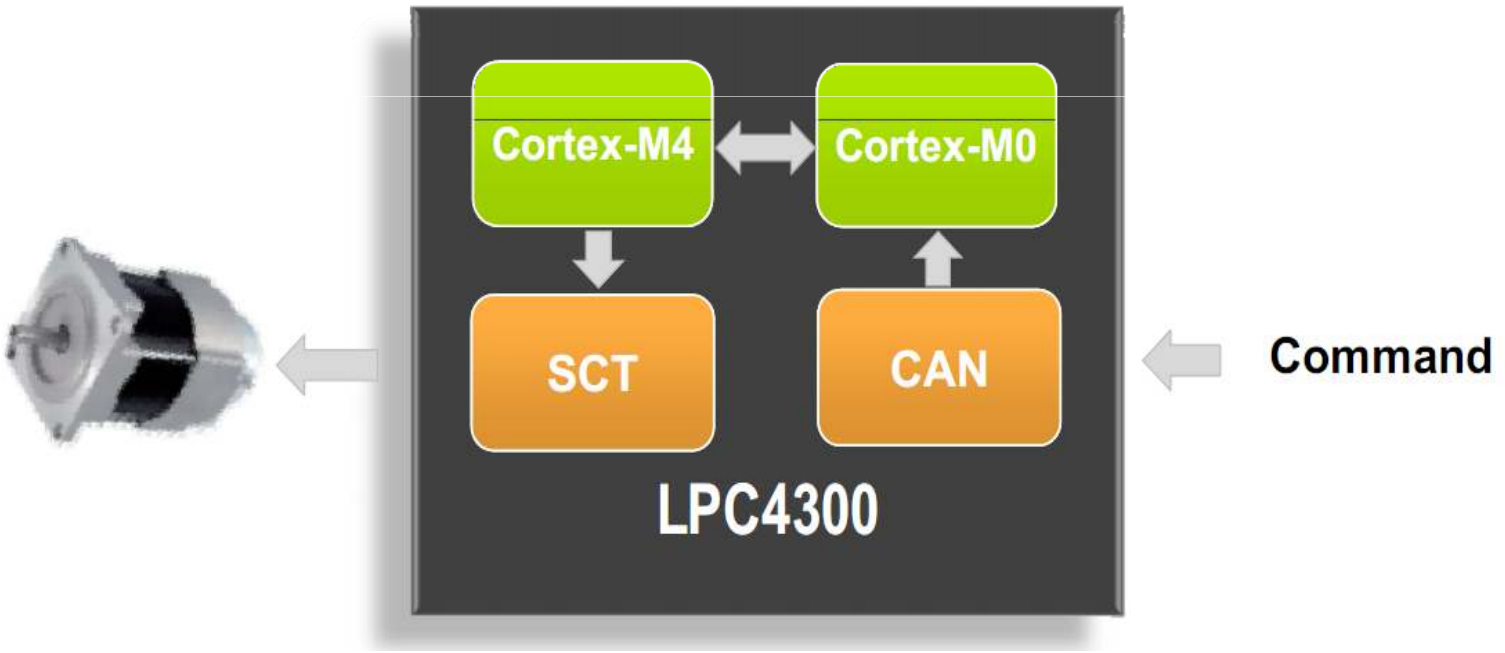


Cortex M0, M4 együttes használat példa audió feldolgozás

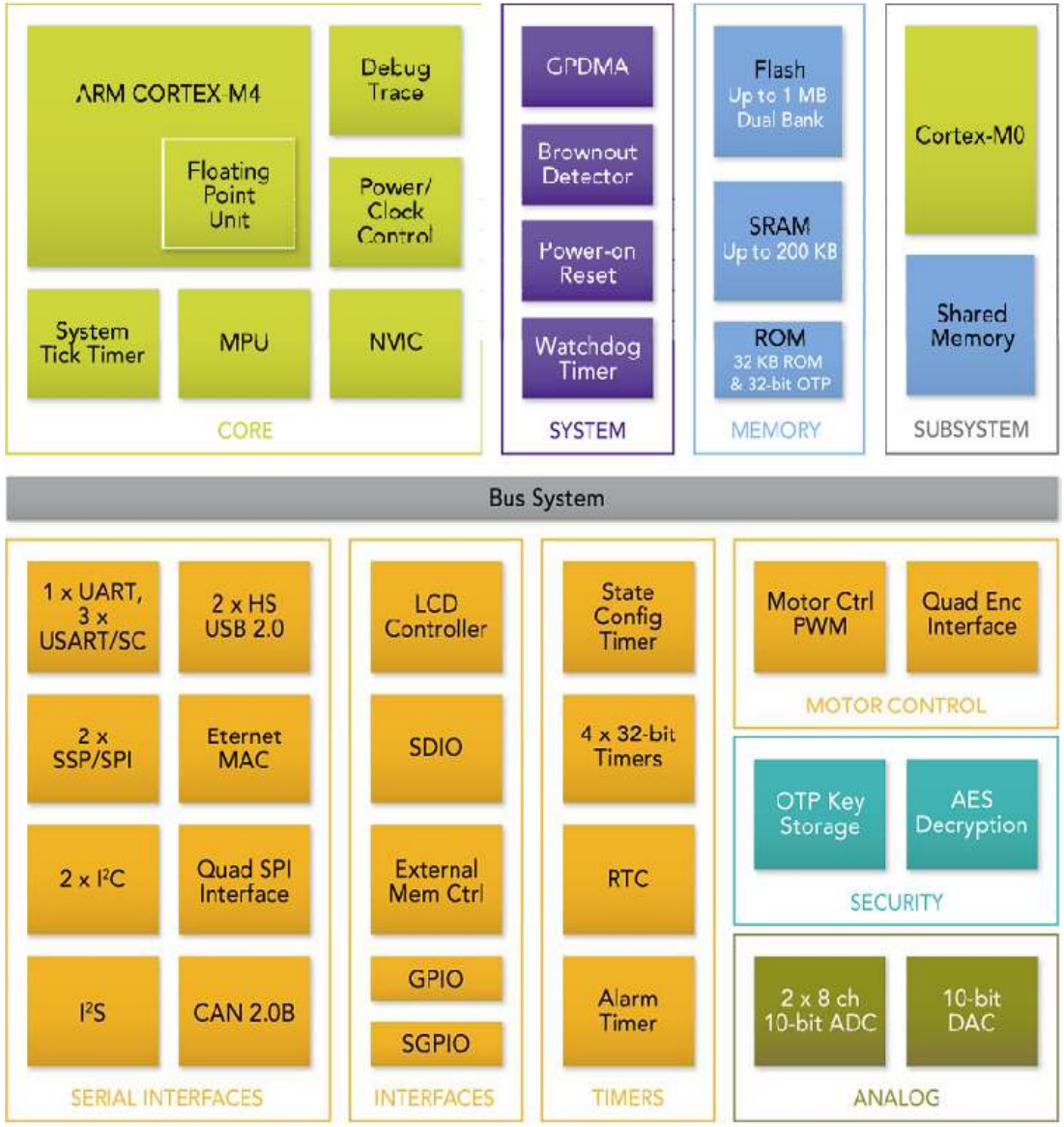
- Cortex M0: perifériakezelés: I2S, USB
- Cortex M4: teljes teljesítménnyel feldolgozás

Cortex M0, M4 motor controll

- Cortex-M4: Motor control Field Oriented Control (FOC)
- Cortex M0: CAN parancs feldolgozás

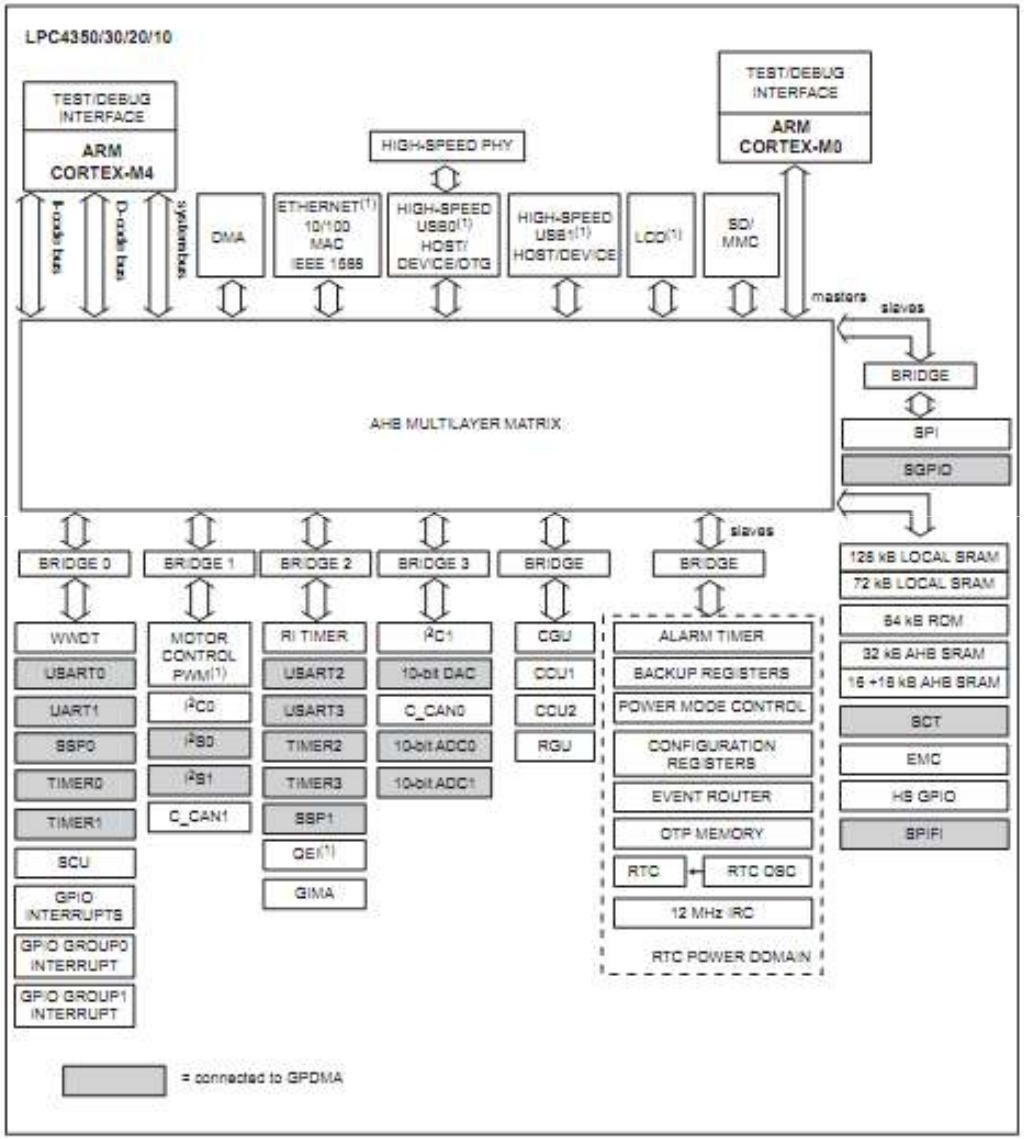


LPC4300 belső felépítés



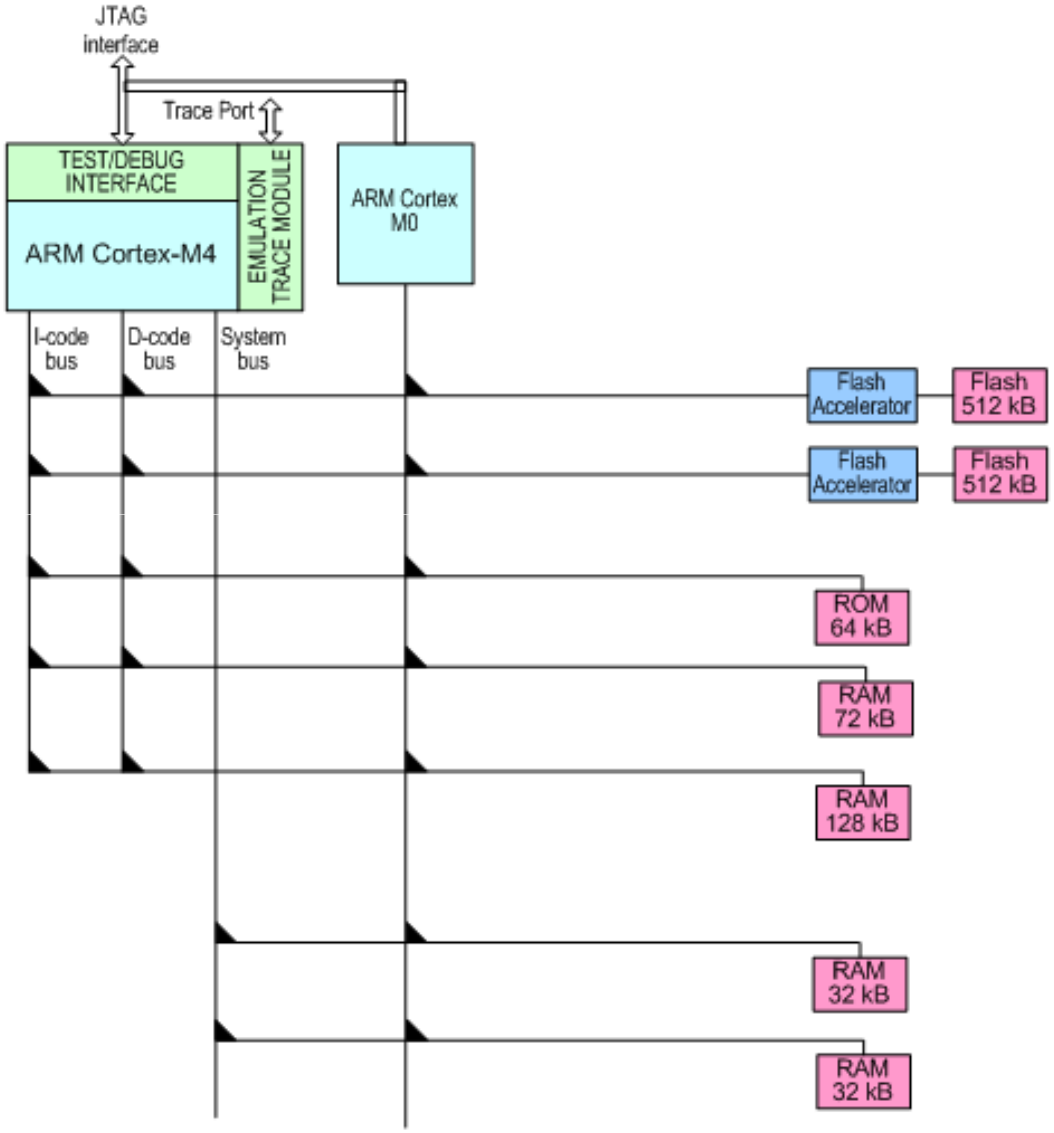
LPC43xx

- Dual Core



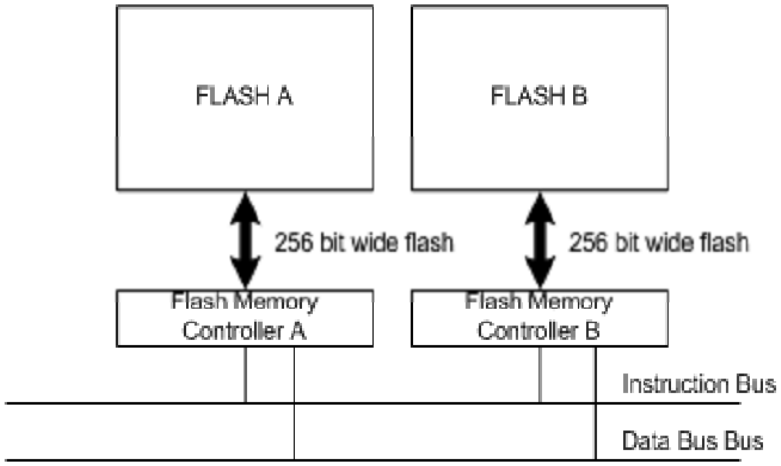
LPC43xx memória rendszer

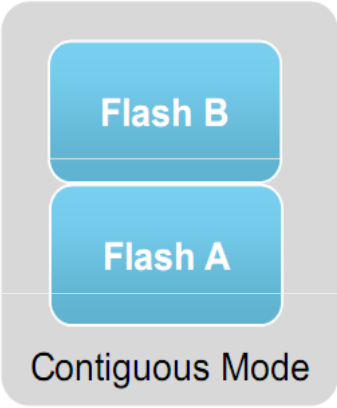
- Dual Core
  - Az M4 és M0 is tud Flash-ből kódot végrehajtani összeakadás nélkül
  - M0 a saját RAM-jából futtatni
  - ROM code Thumban van, mindkettő tudja futtatni
  - M4 MPU-ja tudja védeni az M0 által használt memóriákat

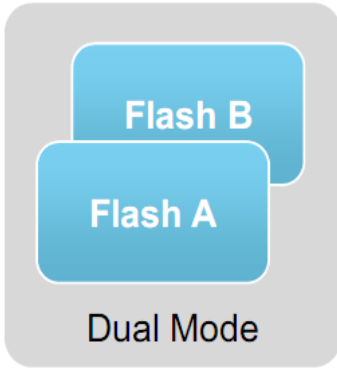


LPC4300 memória
Flash

- Két 512K byte-os flash memoria blokk
  - Lehet összefüggő 1 Mbyte-os blokként használni
- 256-bit-es memória vezérlő
  - 150MHz.

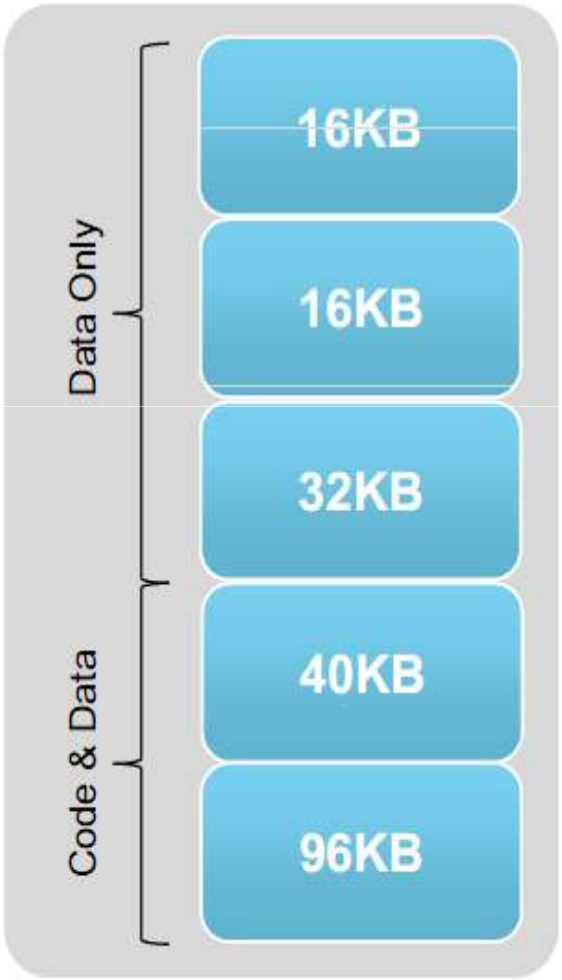






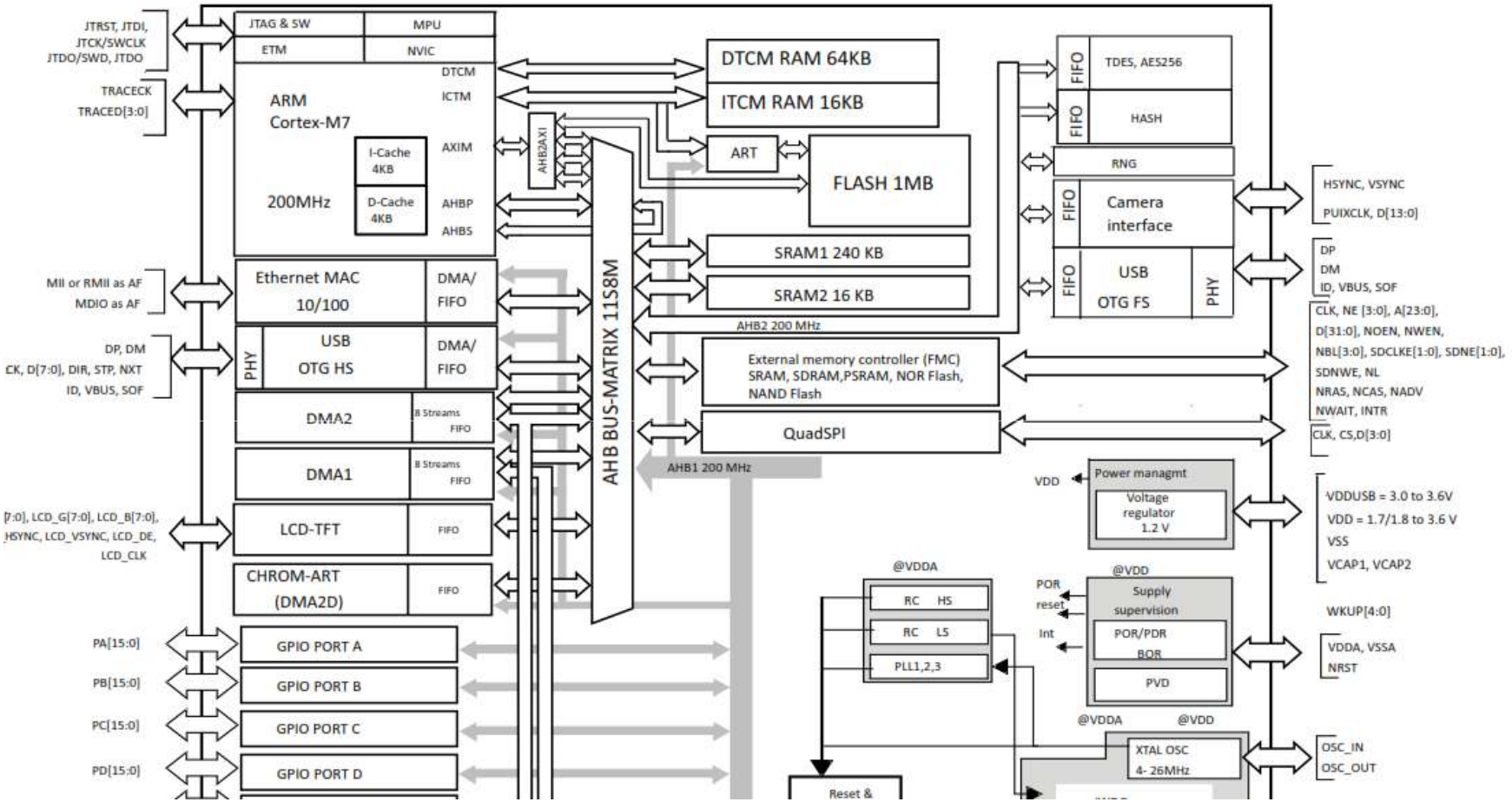
LPC4300 memória
SRAM

- max. 256KB SRAM
- Sok blokkra osztva
  - párhuzamos DMA
  - két core működés

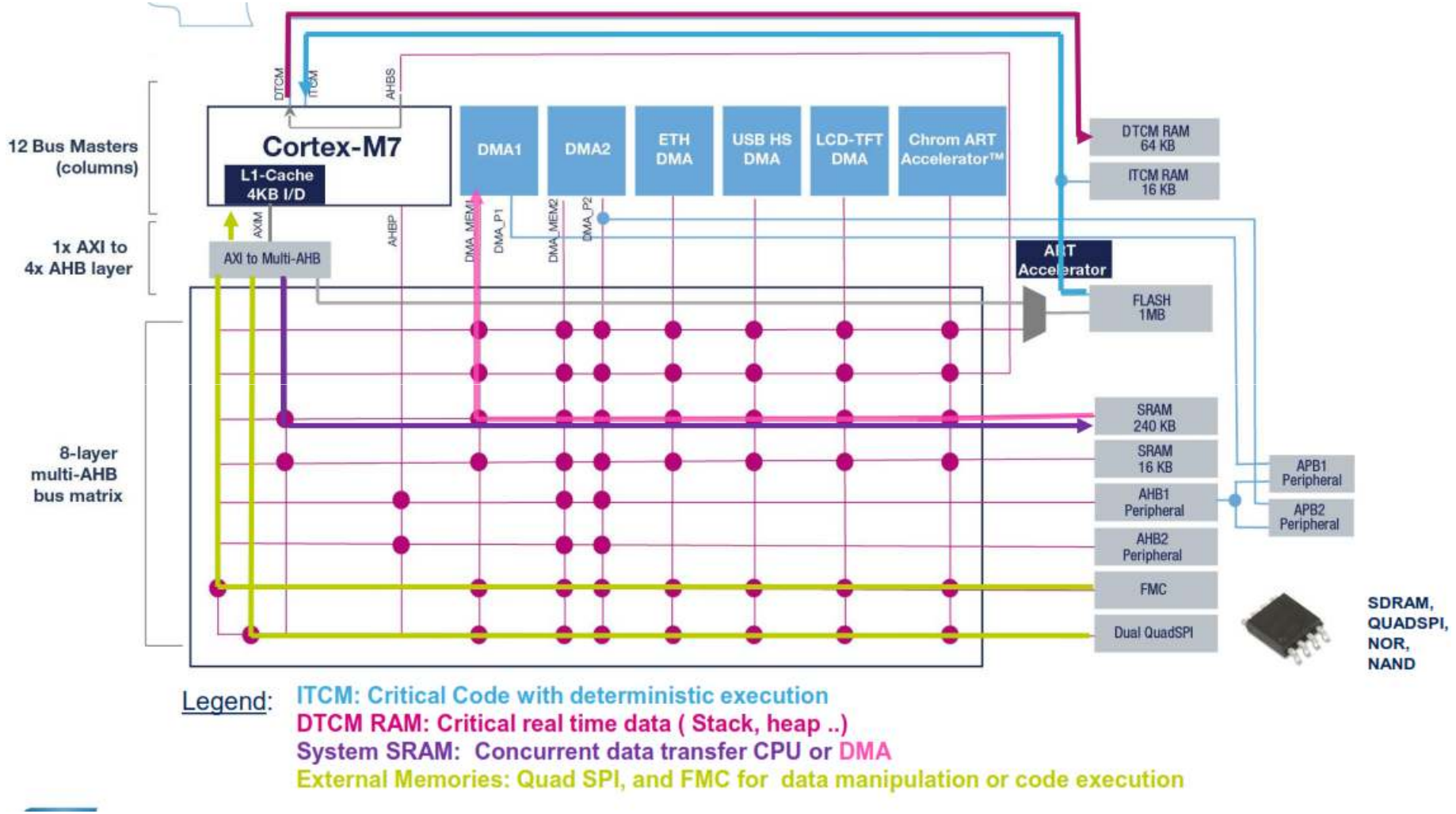


A legelterjedtebb Cortex M7 sorozatok belső felépítése

STM32F7xx belső felépítés



STM32F7xx belső felépítés



Legend:
ITCM: Critical Code with deterministic execution
DTCM RAM: Critical real time data (Stack, heap ..)
System SRAM: Concurrent data transfer CPU or DMA
External Memories: Quad SPI, and FMC for data manipulation or code execution

System Control Block

System Control block

- A rendszer órajel forrásának kiválasztása
  - Külső Quartz, Belső RC oszcillátor, Órajel quartz
- PLL (Phase Locked Loop)
  - A rendszer órajel meghatározása
- Egyes periféria buszok órajelének meghatározása
  - Core órajel és a különböző periféria órajelek közötti viszony
- Gyorsabb vezérlőknél a Flash hozzáférés szabályozása
  - Flash gyorsítás, Wait ciklusok száma stb.
- Perifériák tápellátásának engedélyezése
  - A modern vezérlőkben gyakorlatilag periféria szinten kapcsolhatóak le az egységek
- Lábak alternatív funkcióinak meghatározása

Reset

A Reset és elindulás folyamata

- A reset forrása
  - Power-on
  - Watchdog
  - Brown – out
  - Külső láb
  - Szoftveres
- A reset forrása egy regiszterből kiolvasható

- Mi történik, mi indul el a reset hatására?

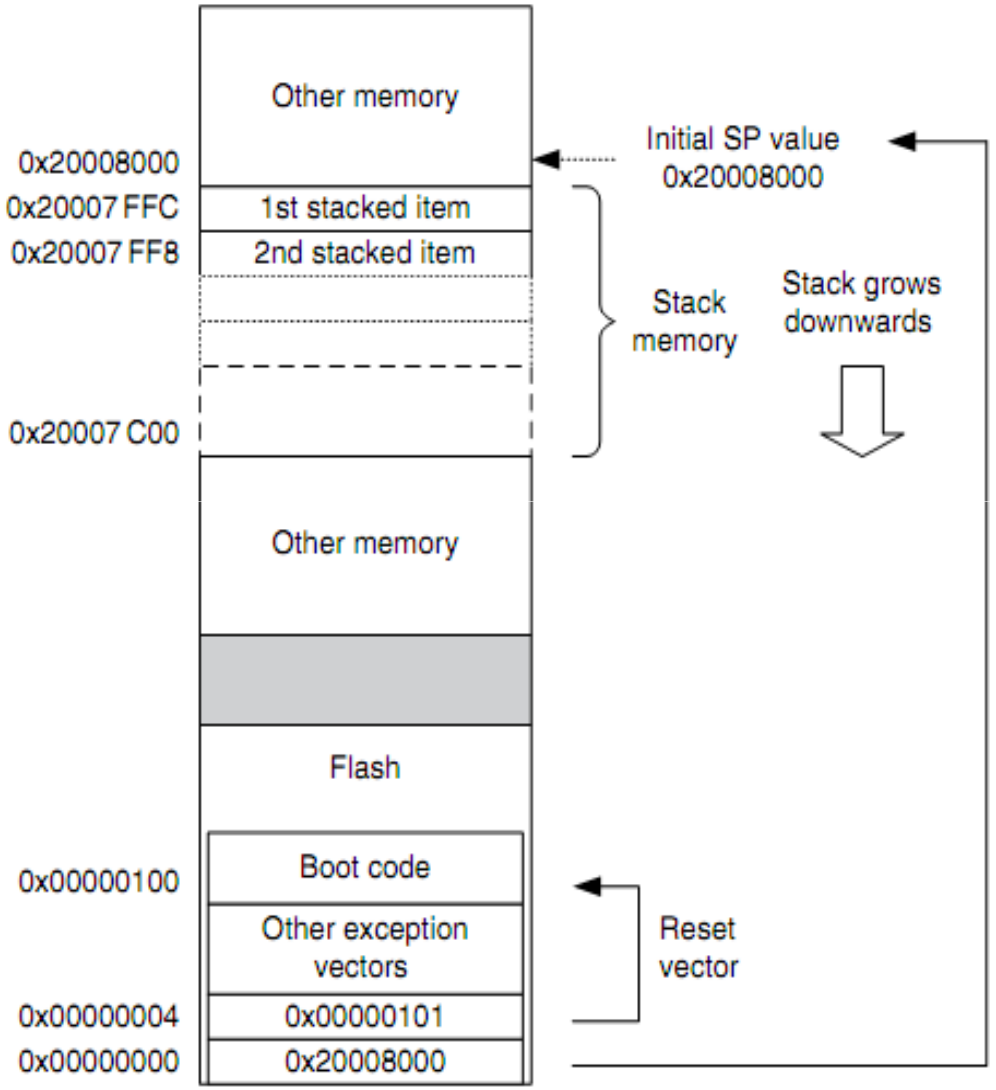
Az NVIC ugrótábla

- Az ugrótábla a címtartomány alján a 0x00000004-ről indul.
  - A 0x00000000-án a kezdő stack pointer van, hogy minél hamarabb lehessen C-t használni.

No.	Exception Type	Priority	Type of Priority	Descriptions
1	Reset	-3 (Highest)	fixed	Reset
2	NMI	-2	fixed	Non-Maskable Interrupt
3	Hard Fault	-1	fixed	Default fault if other hander not implemented
4	MemManage Fault	0	settable	MPU violation or access to illegal locations
5	Bus Fault	1	settable	Fault if AHB interface receives error
6	Usage Fault	2	settable	Exceptions due to program errors
7-10	Reserved	N.A.	N.A.	
11	SVCall	3	settable	System Service call
12	Debug Monitor	4	settable	Break points, watch points, external debug
13	Reserved	N.A.	N.A.	
14	PendSV	5	settable	Pendable request for System Device
15	SYSTICK	6	settable	System Tick Timer
16	Interrupt #0	7	settable	External Interrupt #0
.....	.....	.....	settable	.....
256	Interrupt#240	247	settable	External Interrupt #240

Gyártó specifikus

A reset utáni elindulás folyamata



Beépített boot code

- Az első on-chip flash-es ARM7-től
  - A flash programozás nem triviális
    - RAM-ba letölteni a programot, majd onnan futtatva az felprogramozni a Flash-t
  - JTAG-elés az első időkben végképp problémás volt
- Beépített bootkód, amely valamilyen módon támogatja a Flash programozását
  - Hol van ez a bootmód és hogyan indul el?

Hol helyezkedik el a boot code

- STM32F107

Table 3. Flash module organization (medium-density devices)

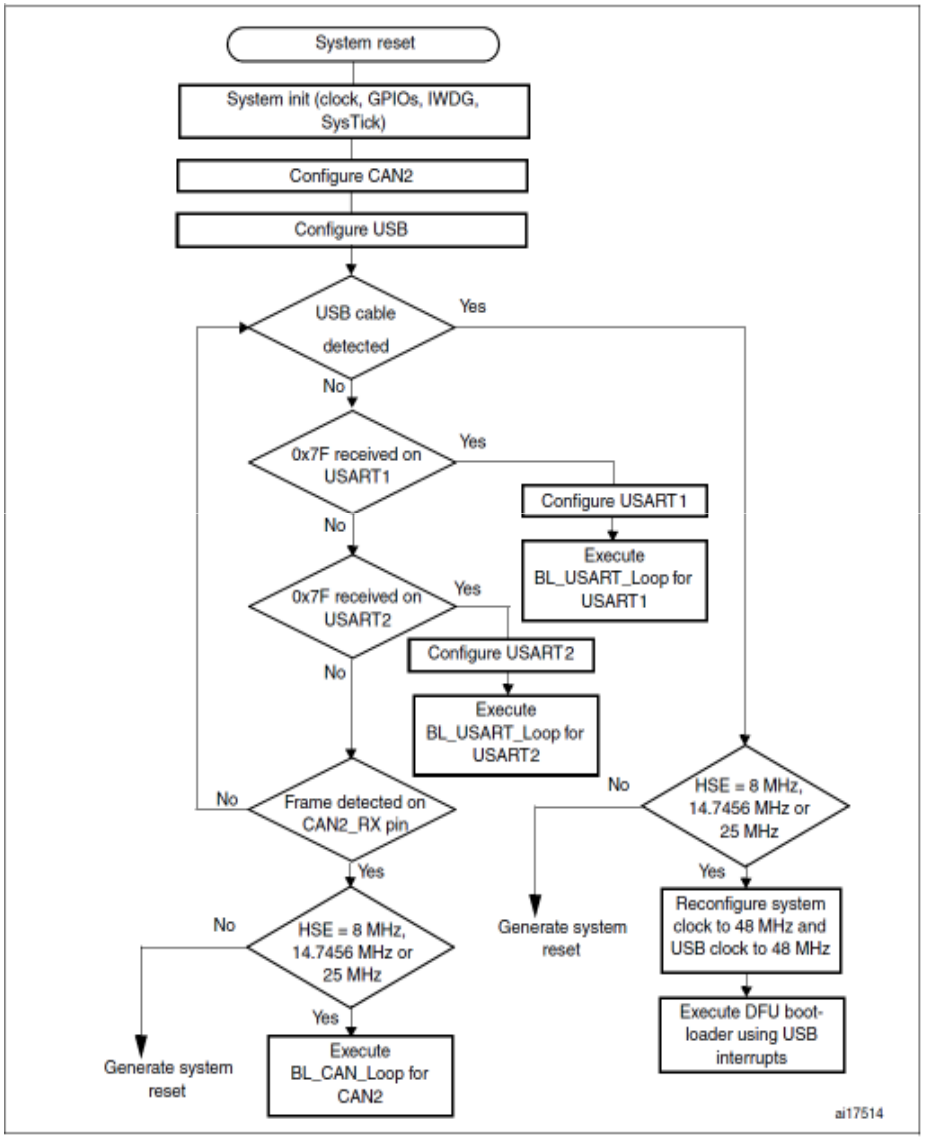
Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	. . .	. . .	. . .
	Page 127	0x0801 FC00 - 0x0801 FFFF	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16

STM32F107 Boot konfiguráció

- Két külső láb függvénye

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main Flash memory	Main Flash memory is selected as boot space
0	1	System memory	System memory is selected as boot space
1	1	Embedded SRAM	Embedded SRAM is selected as boot space

STM32F107 viselkedése bootnál



Flash gyorsító modul

Flash memória

- A Flash memóriák kisebb helyet foglalnak el, mint a RAM-ok, de lassabbak.
- Flash memória hozzáférési idők
  - 30ns - 50ns (33 –25 MHz)
- Ez kevés a 60, 72, 120, 180, 2xx MHz-ről való működéshez
- Megoldások
  - Futtassunk kódot RAM-ból
    - RAM drága és sokat fogyaszt.
  - Emeljük meg Flash hozzáférés bitszámát,
    - 64bit, 128 bit
    - Valamilyen interfészt igényel

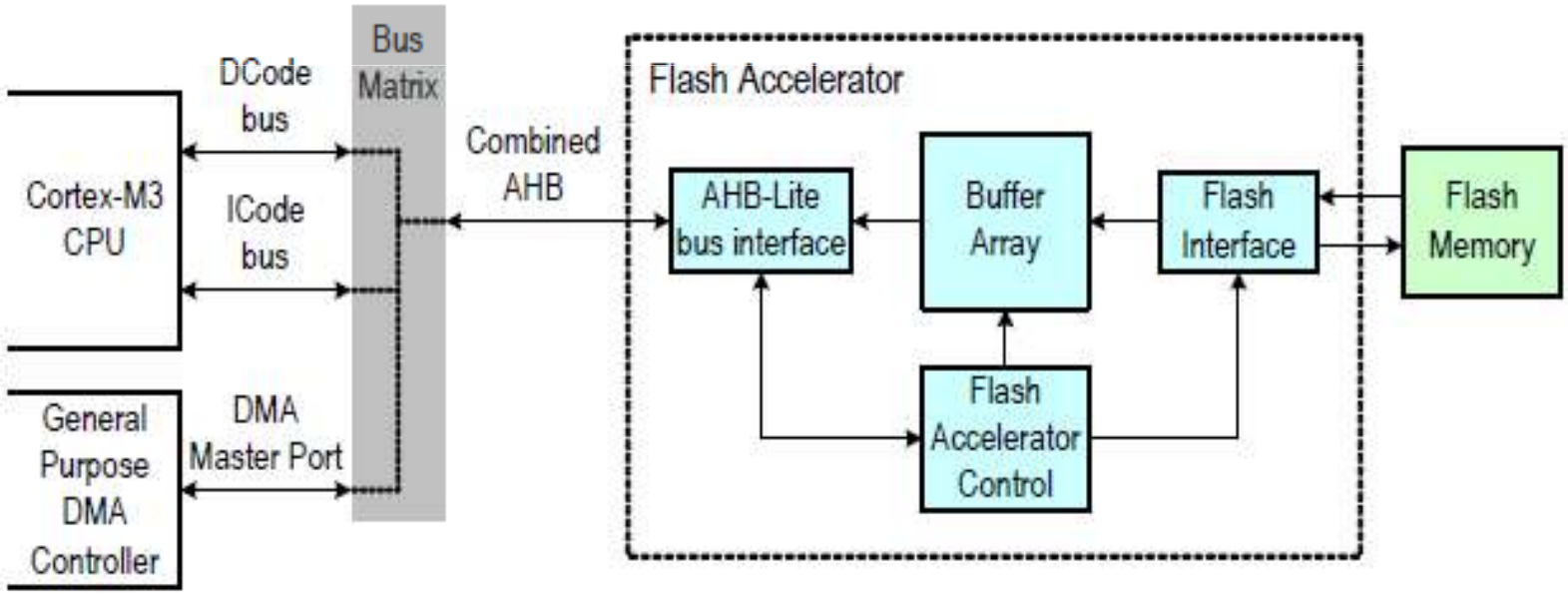
Az STM32F10x megoldása 2009

- 2 db 64 bites prefetch buffer
- A wait ciklusok számát fel kell programozni.

zero wait state, if 0 < SYSCLK ≤ 24 MHz
one wait state, if 24 MHz < SYSCLK ≤ 48 MHz
two wait states, if 48 MHz < SYSCLK ≤ 72 MHz

Az LPC1768 megoldása 2010

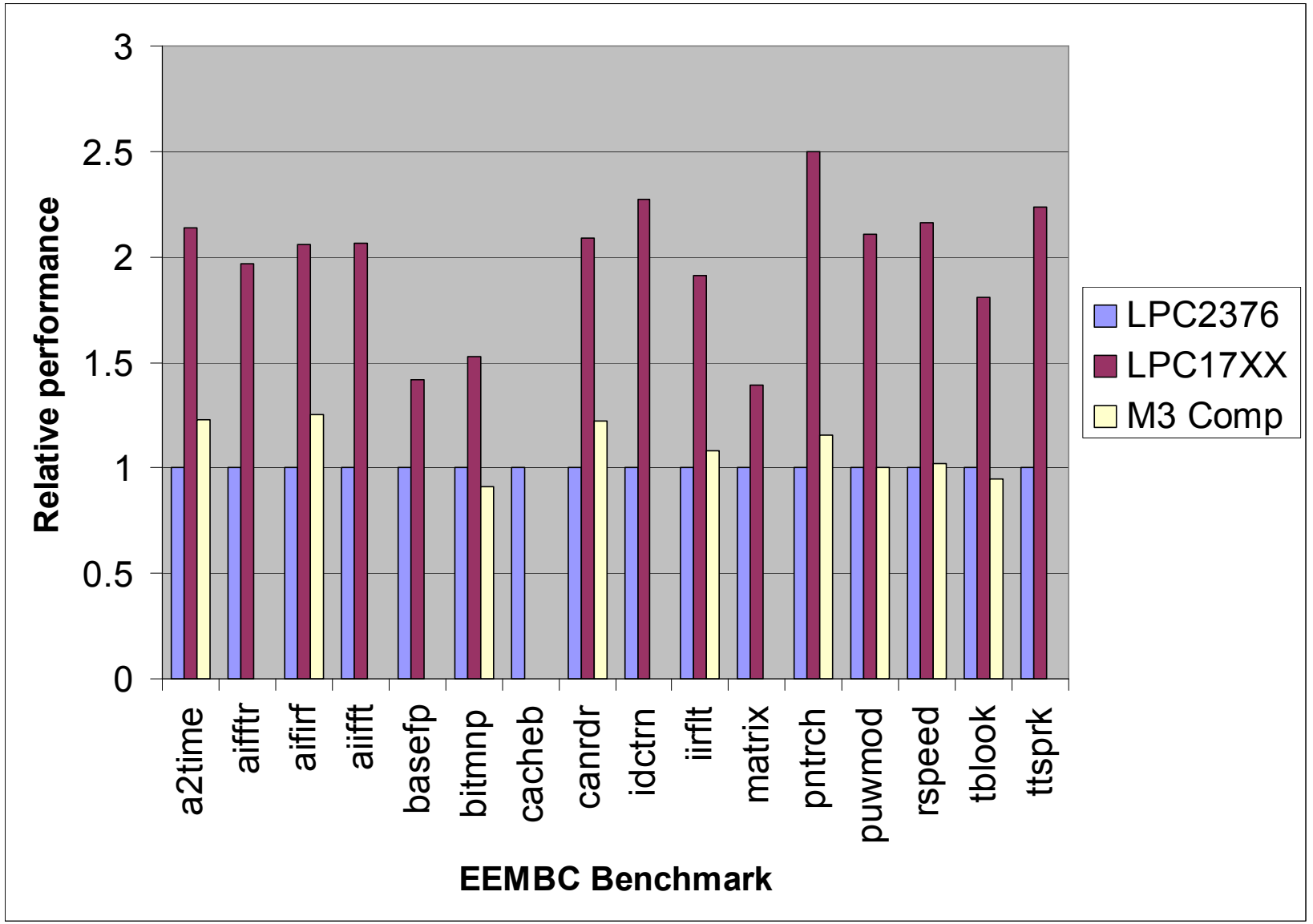
- 8 darab 128 bites szó
- nem csak utasítások, de adatokat is fel tud hozni
- wait ciklusok számát fel kell programozni



LPC1768 megoldás eredményei

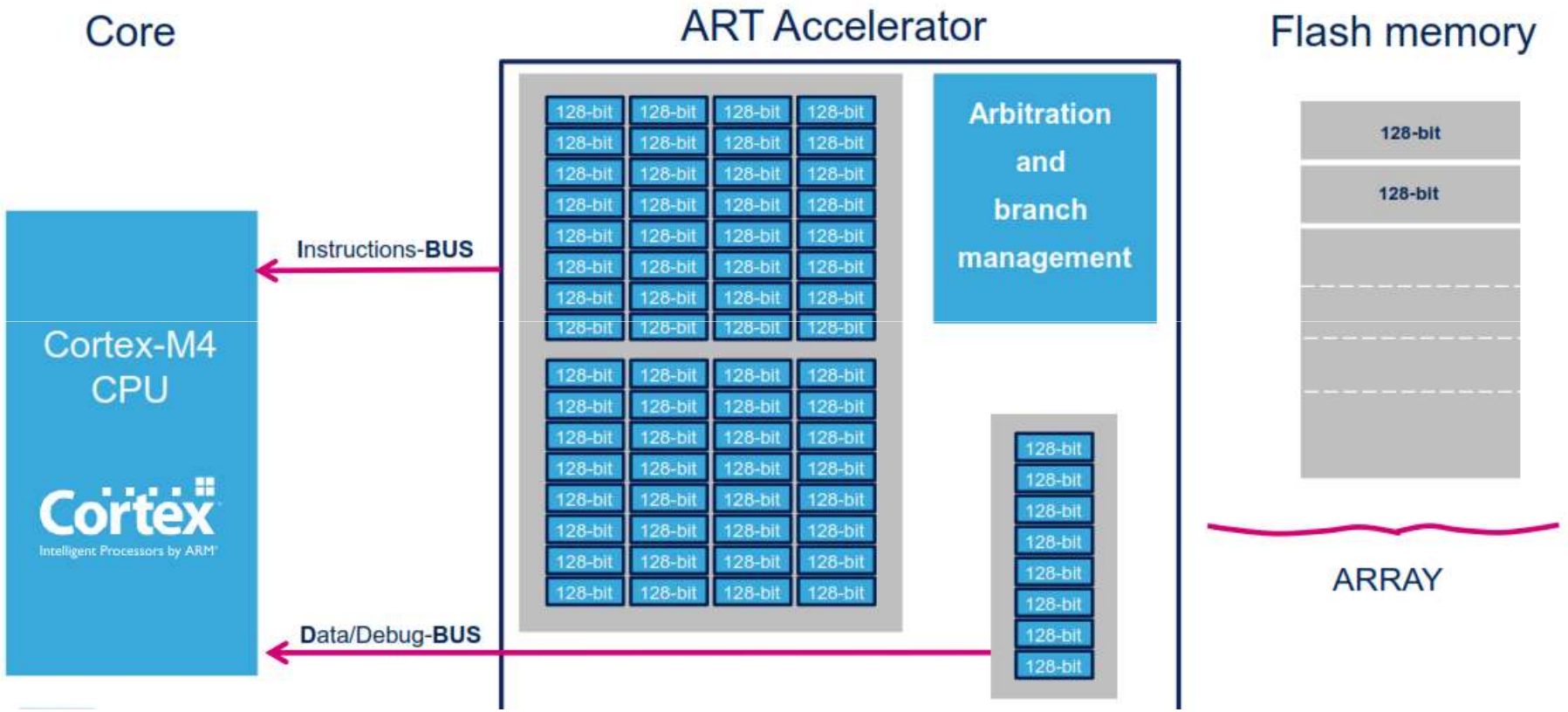
- RAM-ból való kód végrehajtással összevetve
  - A végrehajtási sebesség 16% -on belül marad
  - A fogyasztás 25%-al kevesebb
- A régi ARM7-es verziónál használthoz képest
  - Sima 128 bites Flash interfész
  - 45% teljesítmény növekedés

Benchmark eredmények

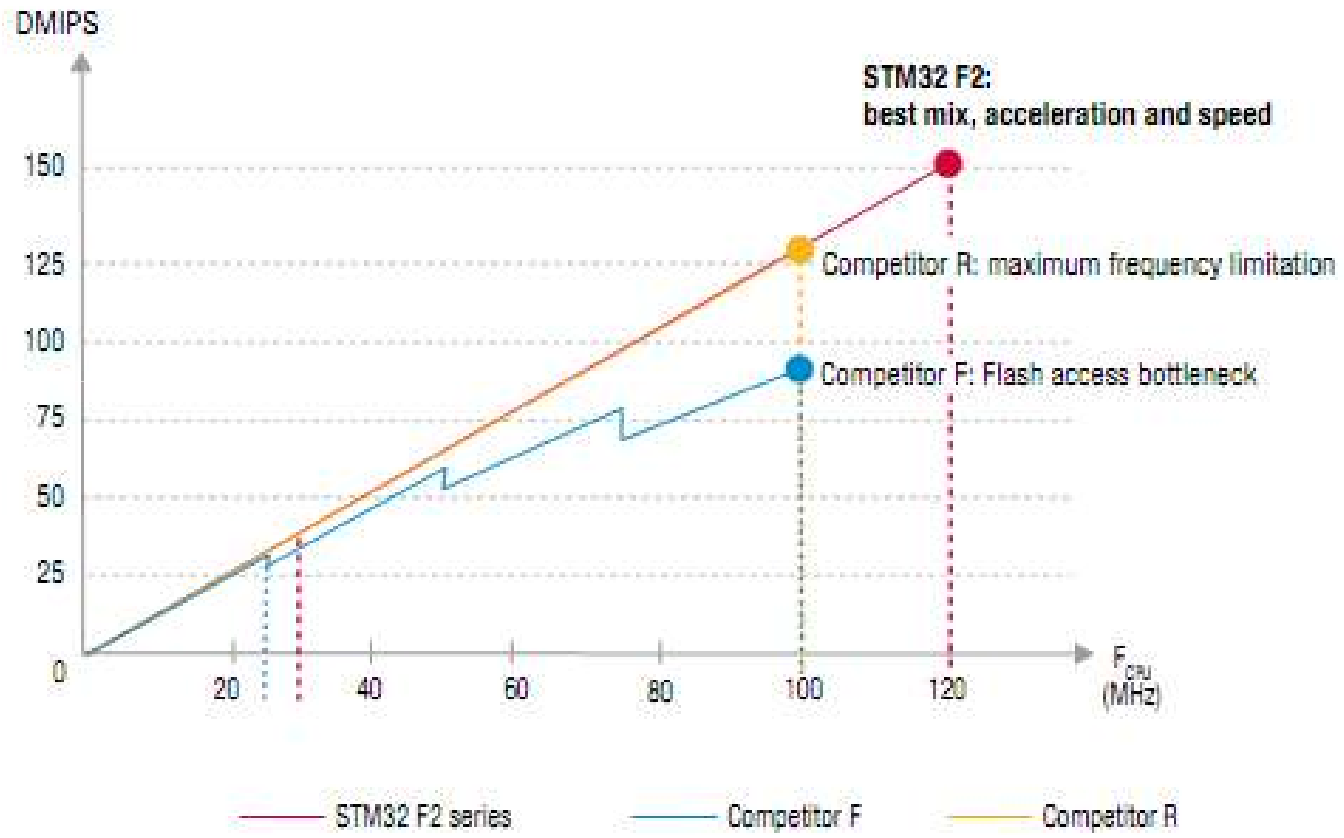


STM32F2xx/STM32F4xx megoldása

Branch manegement és Cache újdonság



STM32F2xx / STM32F4xx megoldása



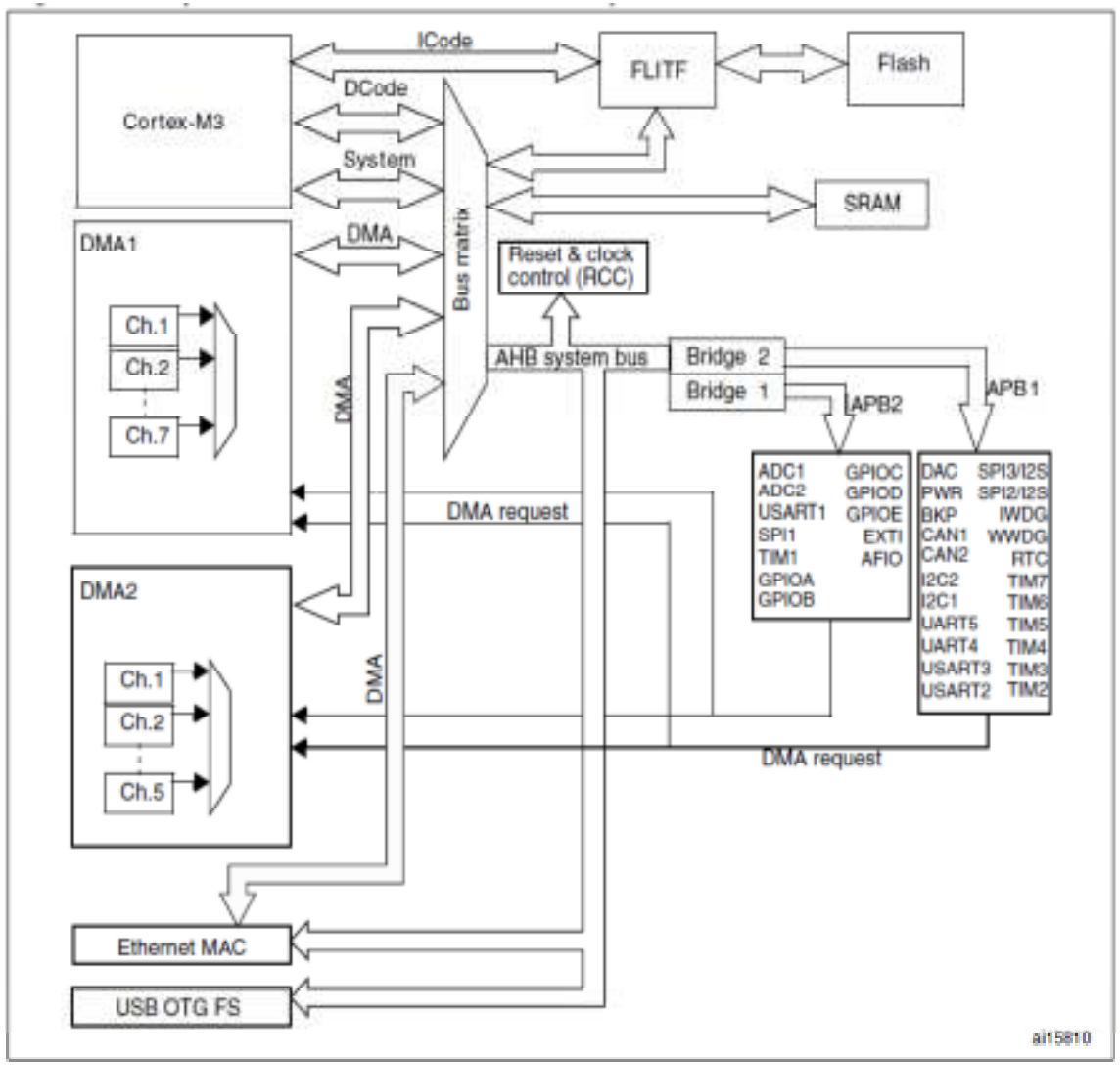
Órajel elosztás

Órajel szétosztás problémái

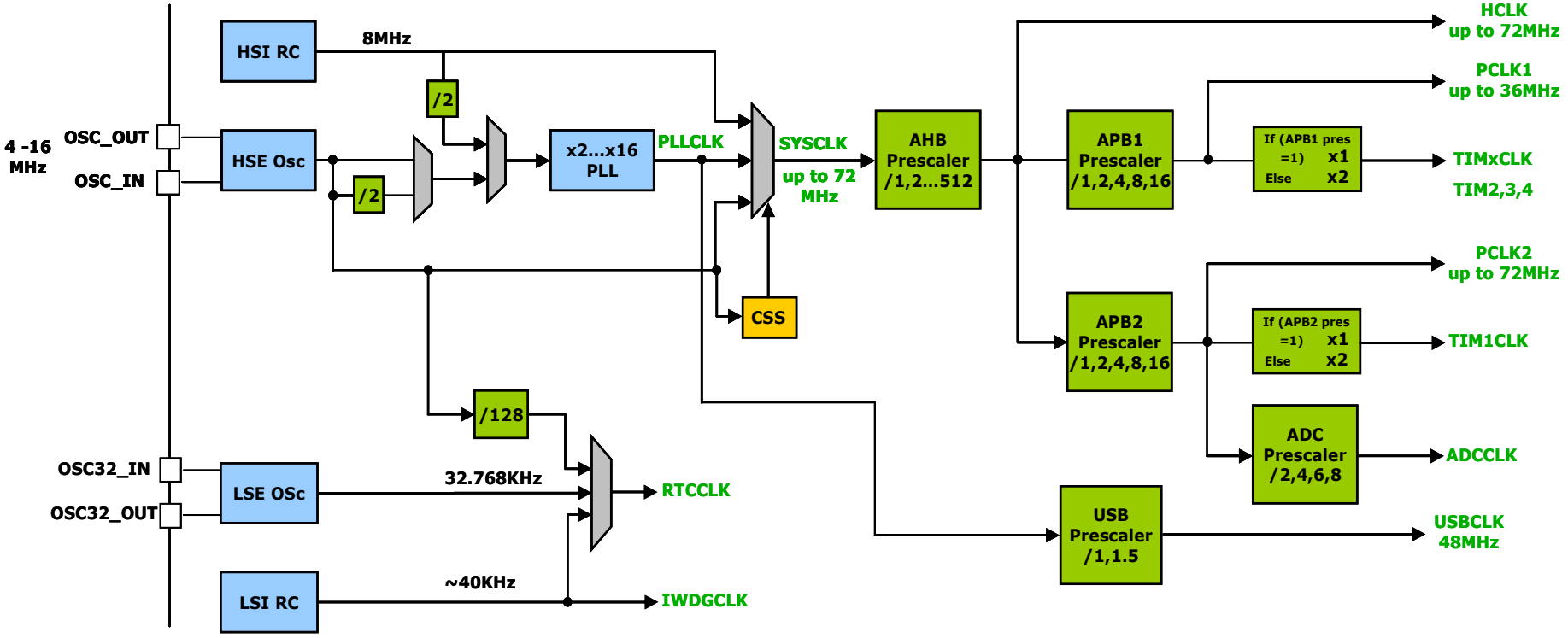
- Követelmény sokféle alapórajel
  - Quartz
    - Pontos, stabil, drága
  - RC osc
    - Pontatlan, olcsó
- Sokféle periféria igény
  - Alap perifériablokkok
  - I/O lábak
  - Ethernet
  - USB

Cortex M3 magú vezérlő belső felépítése

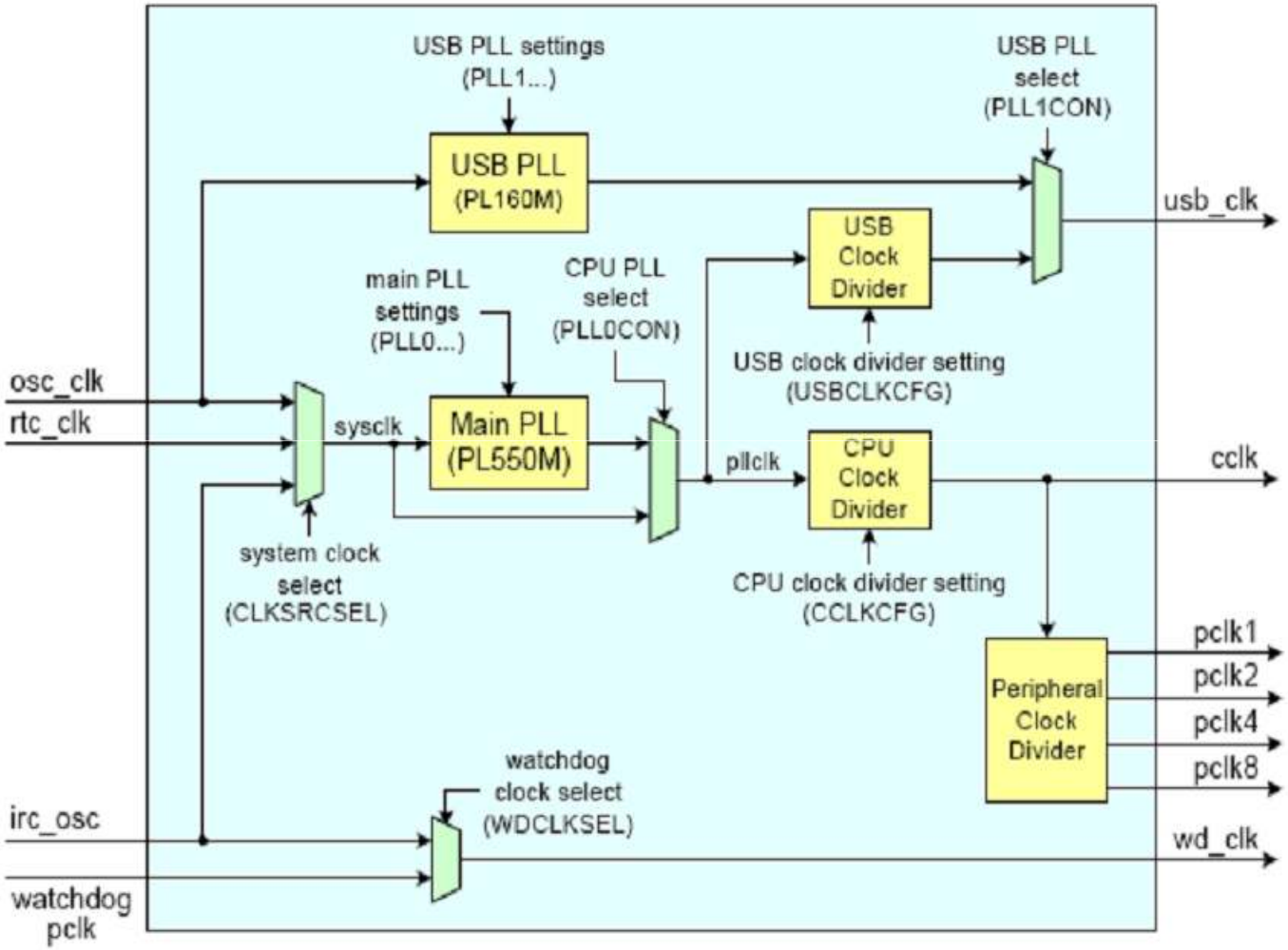
2009: STM32F107 (Max 72 MHz)



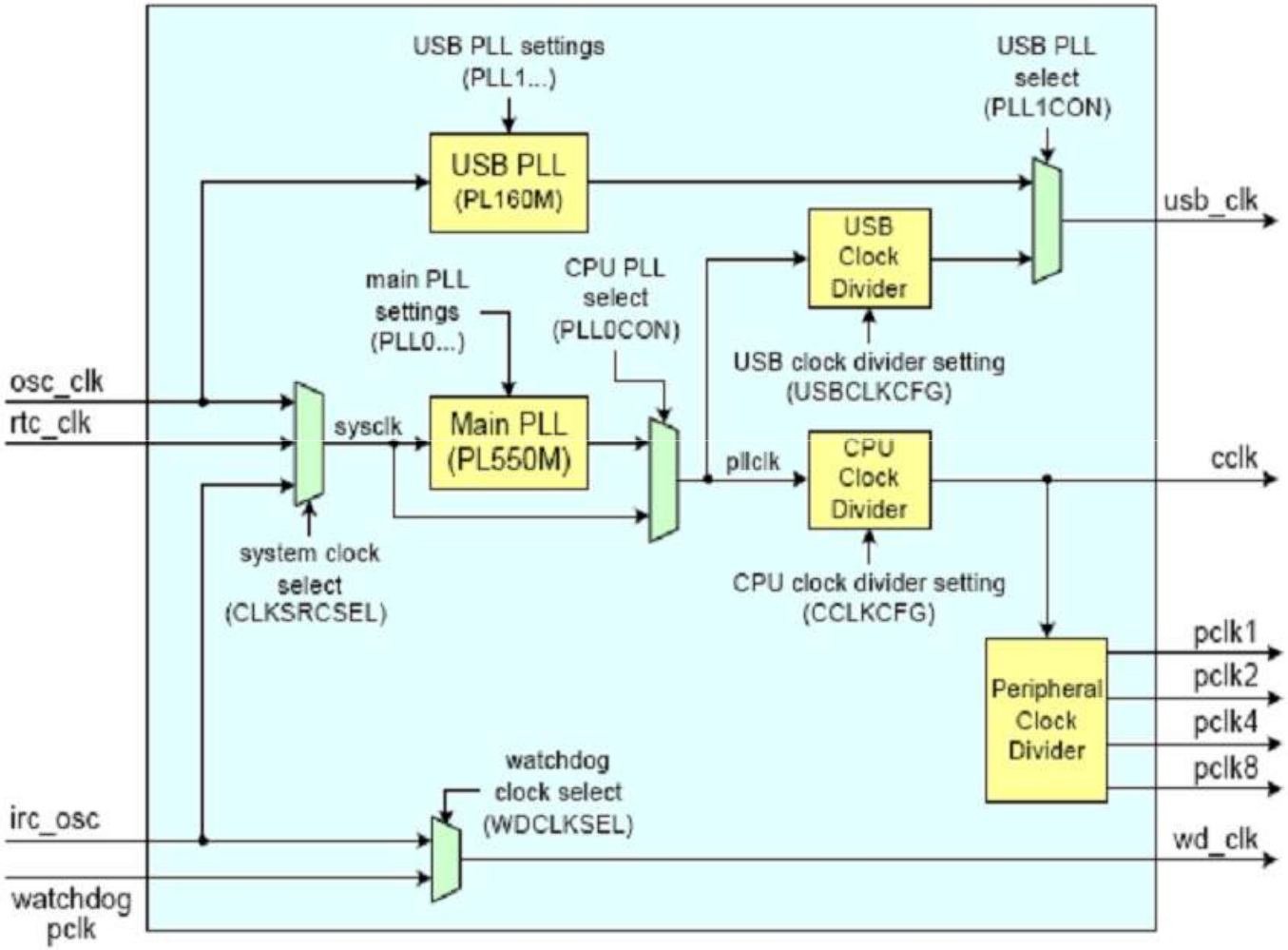
Az STM32F107 megoldása



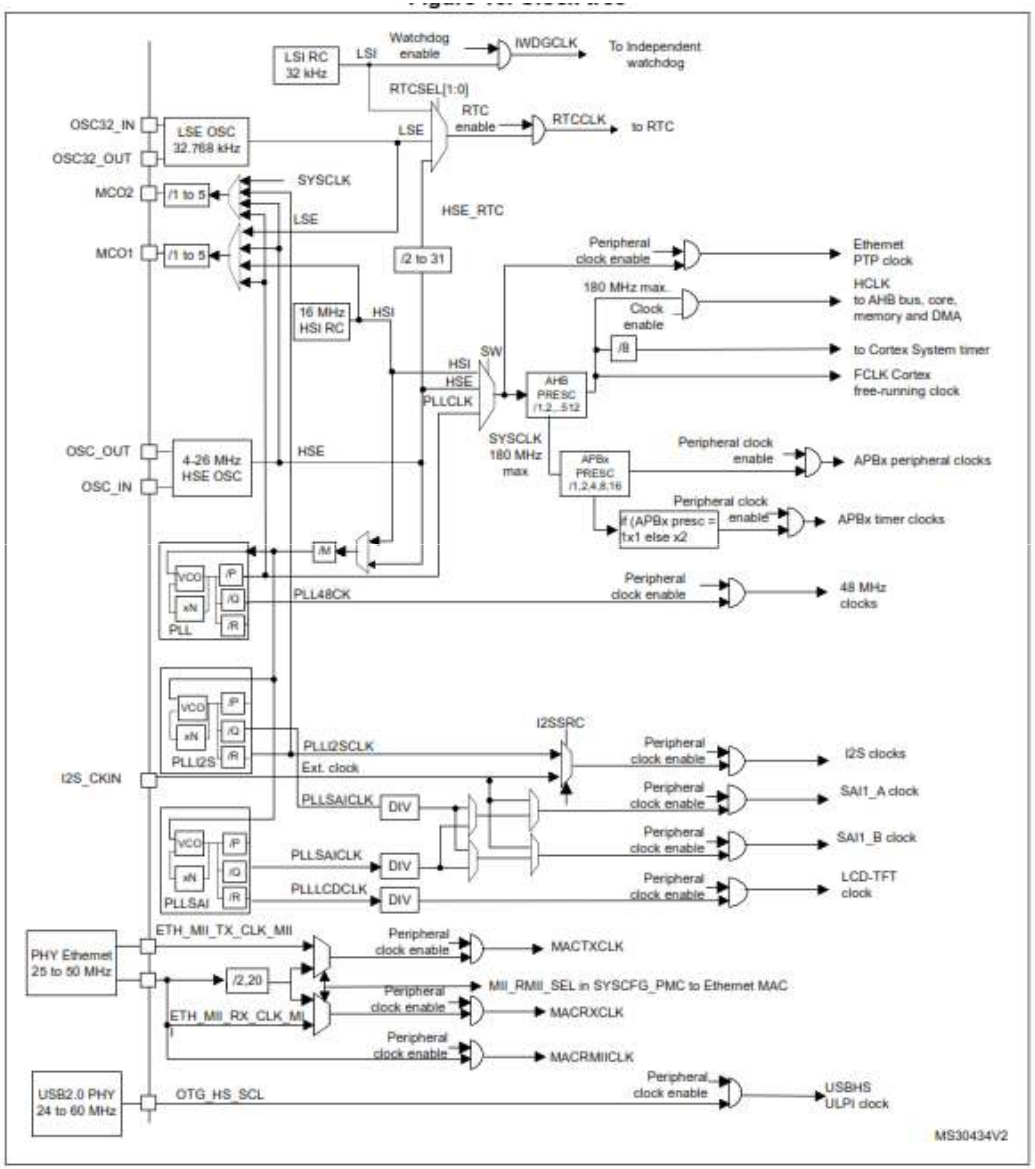
Az LPC1768 megoldása



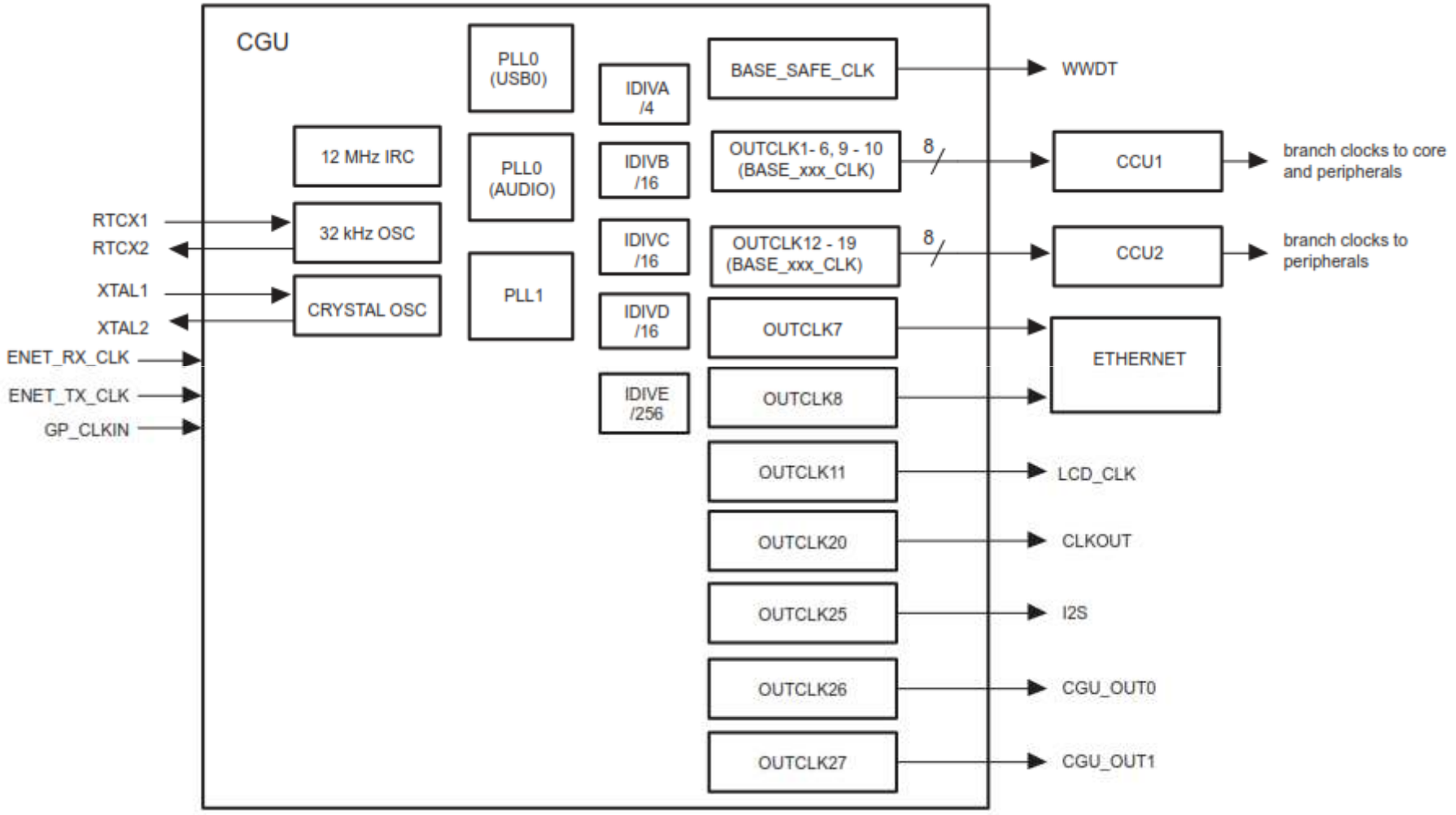
Az LPC1768 megoldása



STM32F4xx


LSI RC 32 kHz
LSI
Watchdog enable
IWDGCLK
To Independent watchdog
RTCSEL[1:0]
RTC enable
RTCCLK
to RTC
OSC32_IN
OSC32_OUT
LSE OSC 32.768 kHz
LSE
MCO2
/1 to 5
SYSCLK
MCO1
/1 to 5
HSE_RTC
/2 to 31
Peripheral clock enable
Ethernet PTP clock
180 MHz max.
Clock enable
HCLK to AHB bus, core, memory and DMA
/8
to Cortex System timer
FCLK Cortex free-running clock
16 MHz HSI RC
HSI
SW
HSI
HSE
PLLCLK
AHB PRESC /1,2...512
SYSCLK 180 MHz max
OSC_OUT
OSC_IN
4-26 MHz HSE OSC
HSE
APBx PRESC /1,2,4,8,16
Peripheral clock enable
APBx peripheral clocks
if (APBx presc = 1)x1 else x2
APBx timer clocks
/M
VCO
xN
/P
/Q
/R
PLL
PLL48CK
Peripheral clock enable
48 MHz clocks
PLLI2S
I2SSRC
PLLI2SCLK
Ext. clock
I2S_CKIN
Peripheral clock enable
I2S clocks
PLLSAICLK
DIV
SAI1_A clock
SAI1_B clock
PLLLCDCLK
LCD-TFT clock
PLLSAI
PHY Ethernet 25 to 50 MHz
ETH_MII_TX_CLK_MII
/2,20
ETH_MII_RX_CLK_MII
MACTXCLK
MII_RMII_SEL in SYSCFG_PMC to Ethernet MAC
MACRXCLK
MACRMIICLK
USB2.0 PHY 24 to 60 MHz
OTG_HS_SCL
USBHS ULPI clock
MS30434V2

Az LPC43xx megoldása



Órajel szétosztás problémái

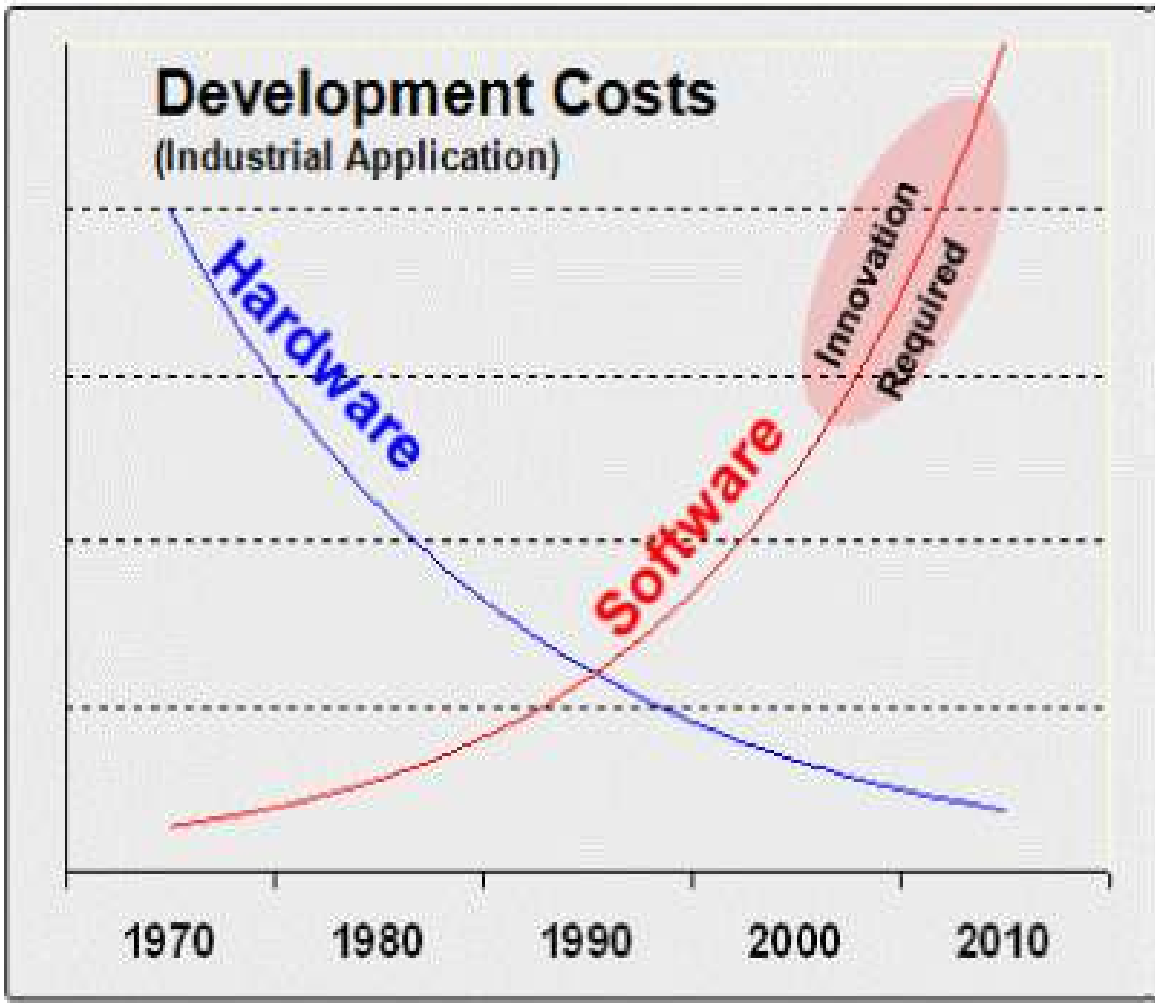
- Egy egyszerű LED villogtatás is minimum 1 órajel engedélyezést igényel, de akár 3-4-et is igényelhet

CMSIS

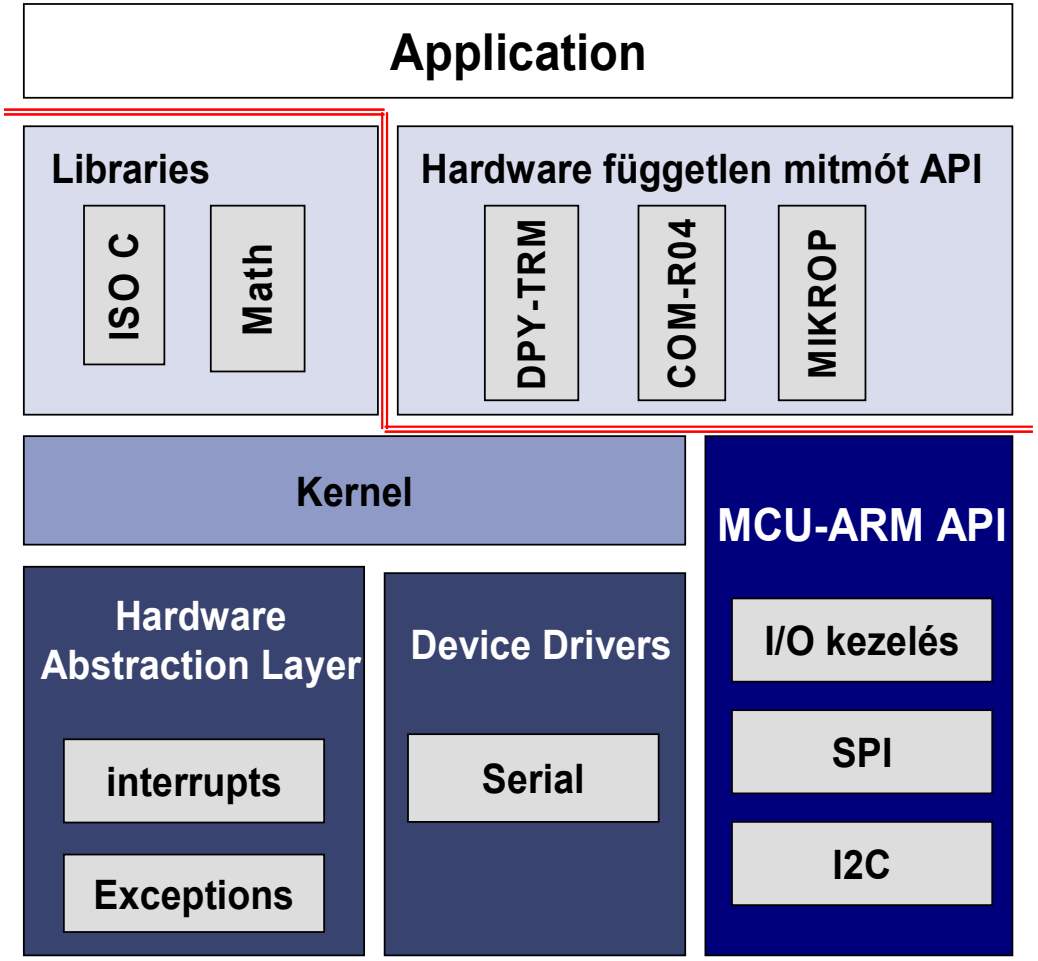
Cortex Microcontroller

Software Interface Standard

Fejlesztési költségek alakulása



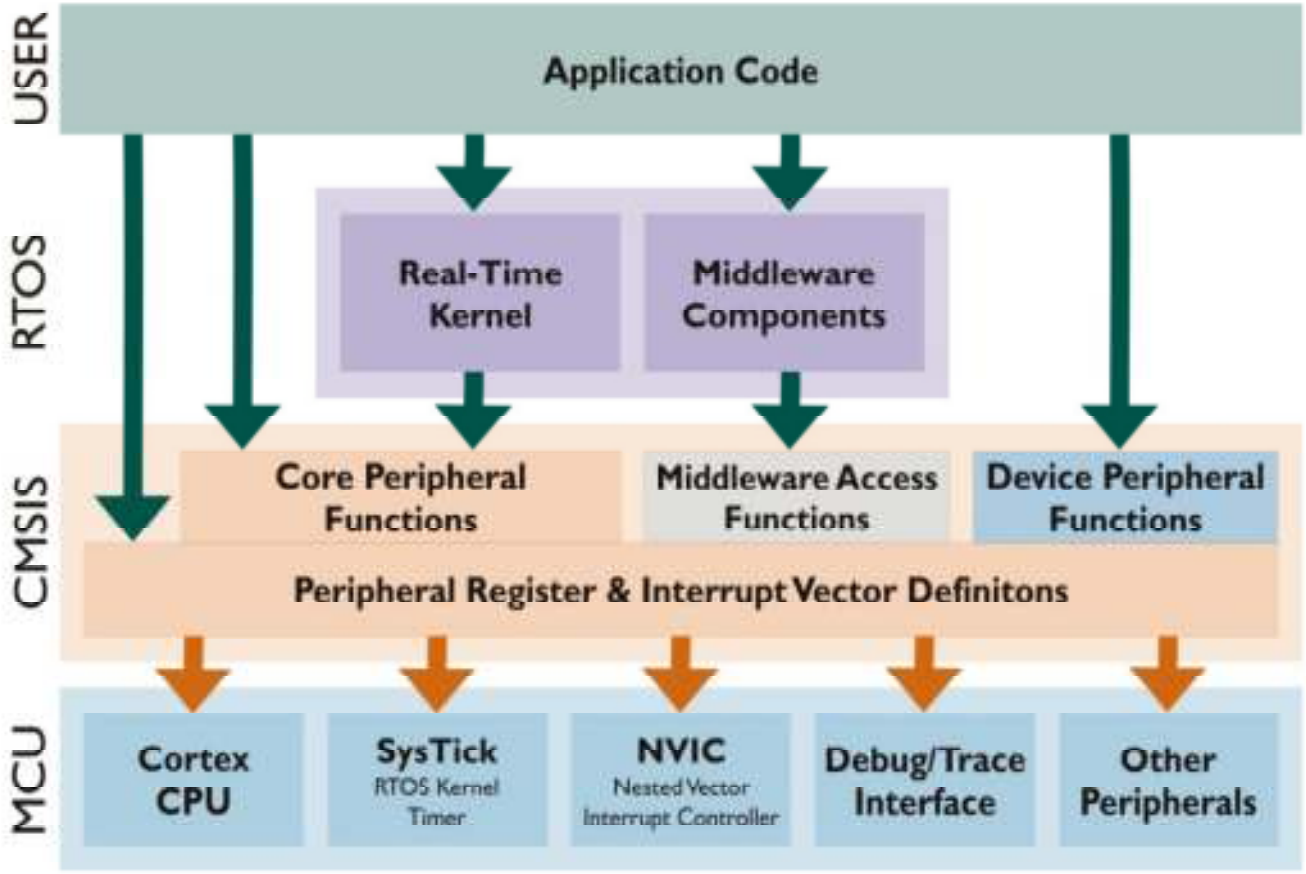
Egy általános beágyazott rendszer SW architektúrája



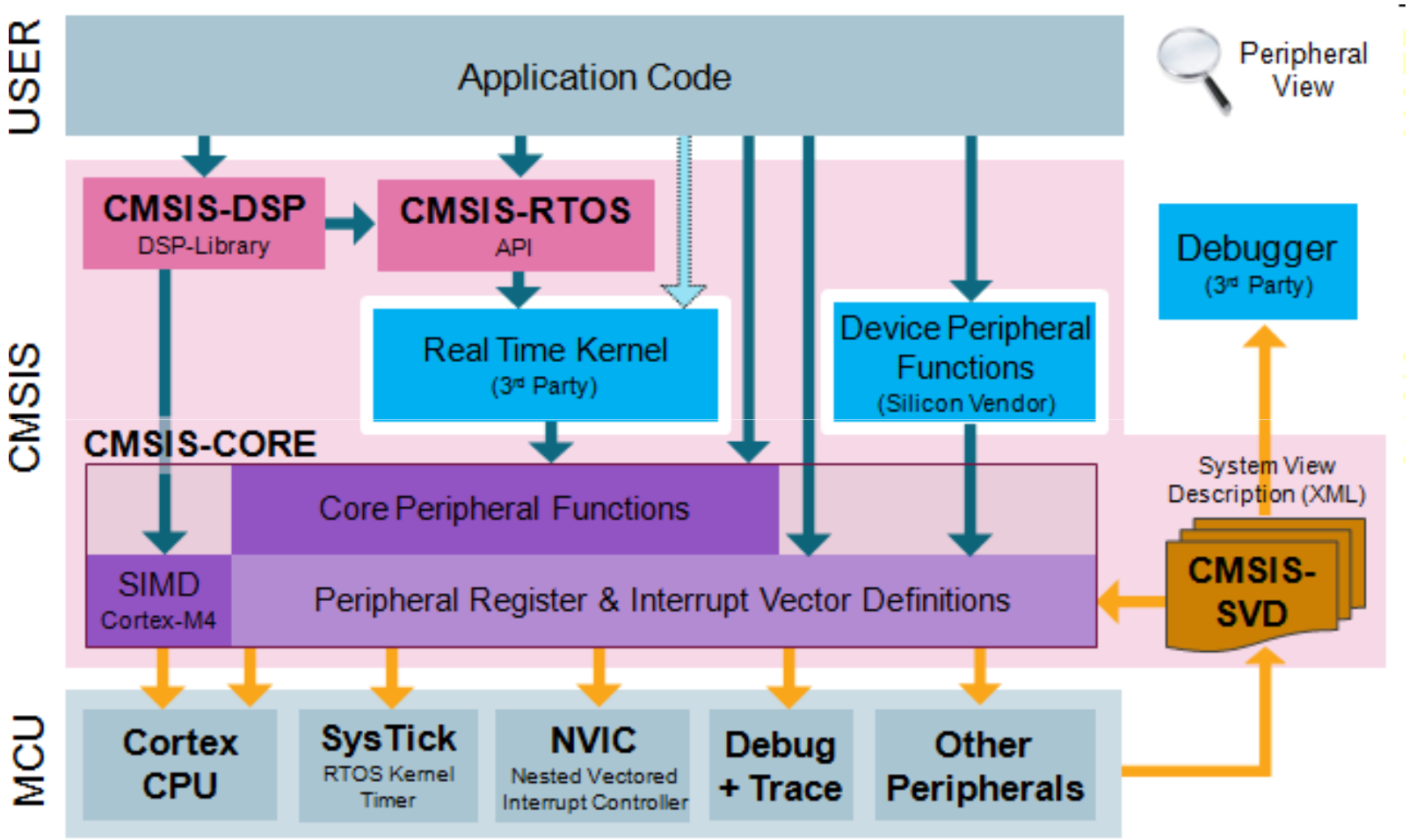
Felhasználói project

Előre fordított eCos distribúció

CMSIS szerkezete (v1.3)



CMSIS szerkezete (v3)



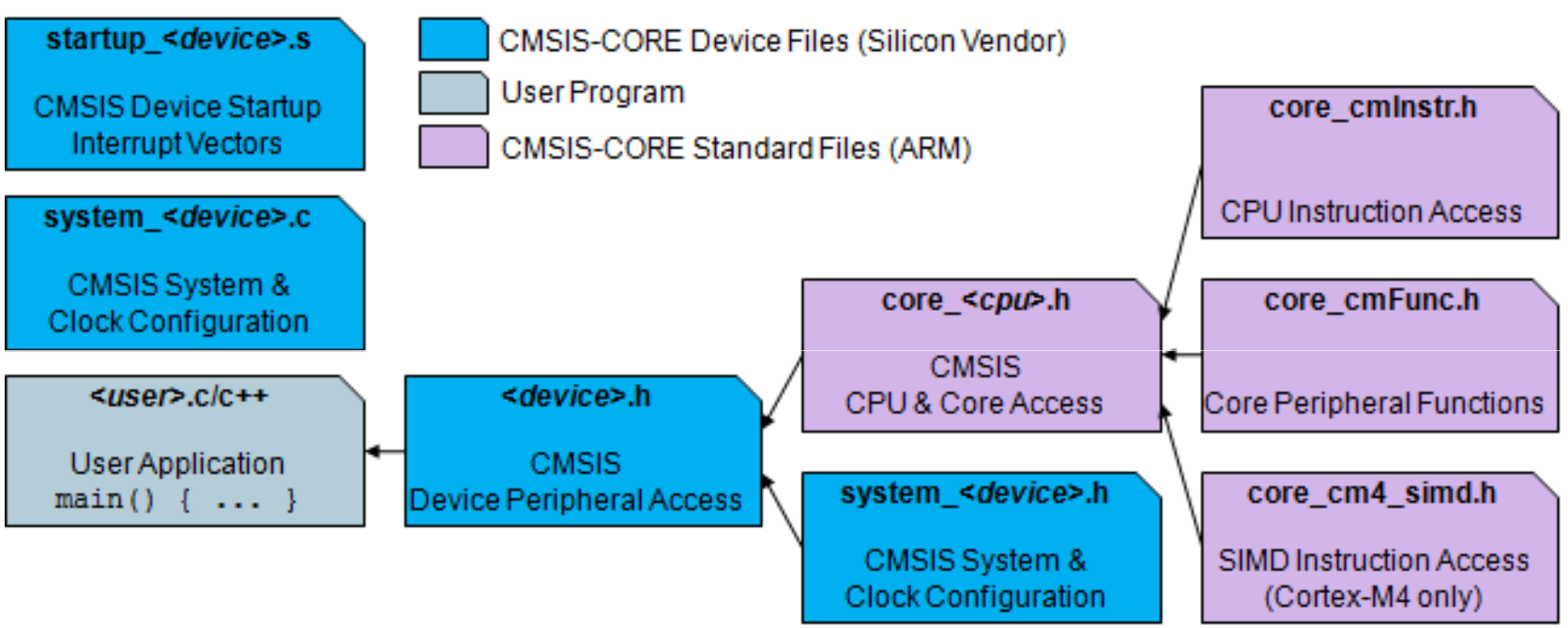
CMSIS Core

- **Hardware Abstraction Layer (HAL):** Az összes Cortex M variánshoz egy standardizált periféria és regiszter készlet kezelés a SysTick, NVIC, System Control Block, MPU, FPU registerszterekhez
- **Rendszer kivétel nevek:** A rendszer kivételekhez, interruptokhoz való hozzáférés deffiniálása
- **Header file szervezés definiciók:** Elnevezési konvenciók az mikrokontroller specifikus interruptok-hoz és header file-okhoz
- **Rendszer indítás:** Mikrovezérlő független inicializáló függvény interfész. Standardized SystemInit() függvény
- **Speciális utasítások támogatása**
- Globális változó a **system clock** frekvencia meghatározására

A CMSIS file-jai és könyvtárstruktúrája

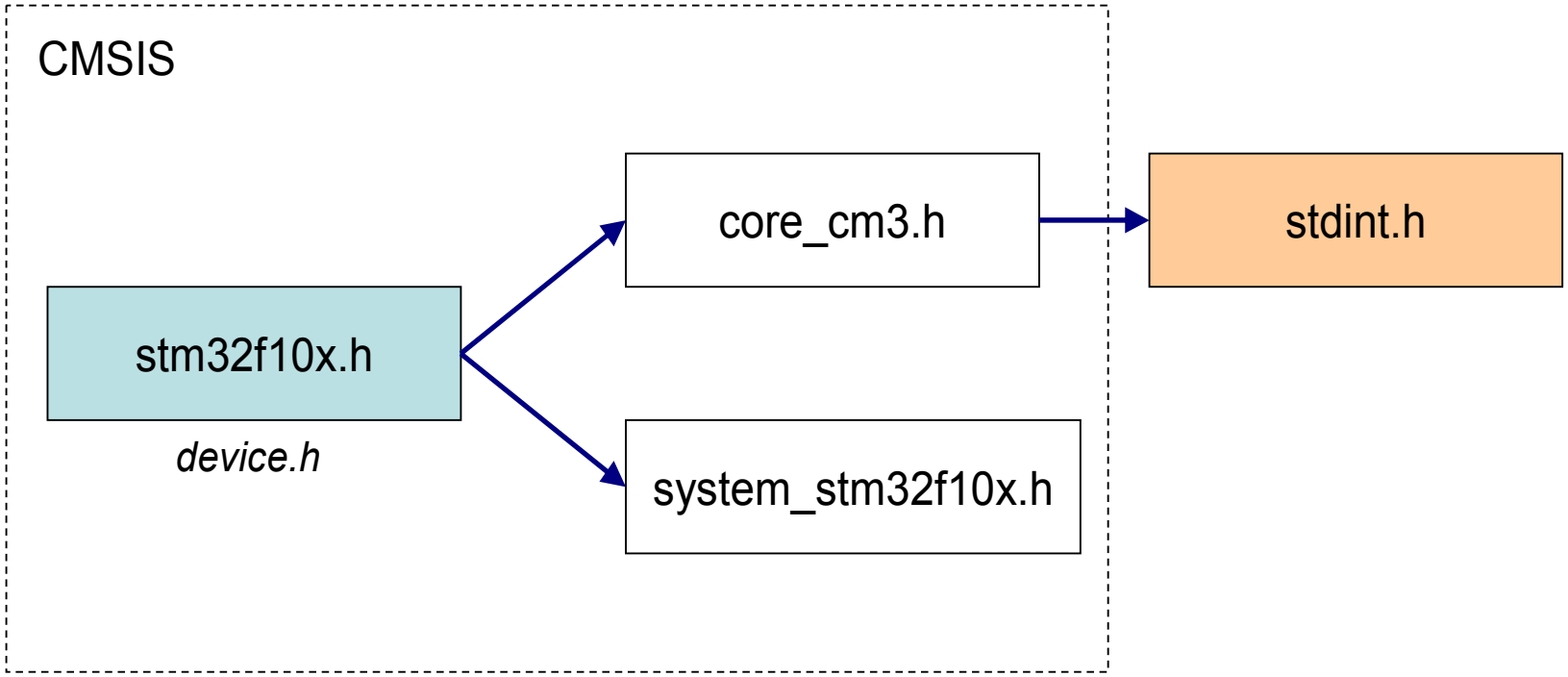
File	Szolgáltató	Leírás
device.h	Mikrovezérlő gyártó	A mikrovezérlő perifériáinak definítciója. Ez a file include-olhatja az összes többi mikrovezérlőhöz tartozó firmware headert.
core_cm0.h	ARM (fordítófüggő részekkel)	A Cortex-M0 alapperiférіáinak és regisztereinek definítciója
core_cm3.h	ARM (fordítófüggő részekkel)	A Cortex-M3 alapperiférіáinak és regisztereinek definítciója
core_cm0.c	ARM (fordítófüggő részekkel)	A Cortex-M0 alapperiférіáinak és regisztereinek kezelőfüggvényei
core_cm3.c	ARM (fordítófüggő részekkel)	A Cortex-M3 alapperiférіáinak és regisztereinek kezelőfüggvényei
startup_*device*	ARM, de a mikrovezérlő, vagy compiler gyártó adoptálhatja és testreszabhatja	A Cortex-M mikrovezérlő startupkódja és a teljes mikrovezérlő függő ugrótábla
system_*device*	ARM, de a mikrovezérlő gyártó, adoptálhatja és testreszabhatja	Mikrovezérlő függő inicializációs rész. Tipikusan a PLL és egyébb órajelforrások inicializálása történik itt.

CMSIS core struktúra



A device.h szerepe

- Az egyetlen szükséges include file a felhasználó számára (az induláshoz)



A *startup_device* file

- Fordító függő
- Startup Code, ugrótábla
- a GCC megvalósításnál a ugrótáblákhoz úgynevezett ***weak*** pragmákat használnak

```
DCD       USART1_IRQHandler,              /* USART1 */
```

Majd az így definiált nevet egy ***weak*** pragmával ráhuzalozzuk egy default handlerre:

```
#pragma weak USART1_IRQHandler = Default_Handler
```

A system_device.c

- Minden Cortex M3 vezérlő azonos módon elindítható legyen
- Minimum szolgáltatások

Függvény definició	Leírás
void SystemInit (void)	A mirovezérlő elindulásához szükséges rutinok végrehajtása. Tipikusan ez a függvény konfigurálja a PLL-t. Ez a függvény tipikusan a startup_device-ból híódik meg, de lehet, hogy a felhasználónak kell meghívnia.
void SystemCoreClockUpdate (void)	Beállítja a PLL-t a SystemCoreClock változó értékének megfelelően.

Változó definítció	Leírás
uint32_t SystemCoreClock	A rendszer órajel frekvenciáját tartalmazza.

A CMSIS coding guildlines-ai

- A CMSIS szabvány a MISRA 2004-es guildline-jainak betartásával készül.
- A CMSIS az adattípusokként az **<stdint.h>** -ba definiált típusokat használja
- A kifejezéseket tartalmazó #define-okat zárójelezni kell.
- A *Core Peripheral Access Layer (**CPAL**)* összes függvénye re-entráns kell, hogy legyen.
- A *Core Peripheral Access Layer (**CPAL**)* nem tartalmaz blokkoló kódot.
- Az összes interupt kezelő függvények a **_IRQHandler** utótaggal kell záródnia.

A CMSIS coding guildlines-ai 2.

- Nagybetűvel jelöl minden core, vagy periféria regisztert, illetve assemly utasítást.
- Az ún. ***CamelCase*** jelölést használja a perifériakezelő függvények és interupt függvények számára.
- PERIFÉRIA_ perfixet kell használni a perifériához tartozó függvényekhez (tehát a függvény neve elött nagybetűvel fel kell tüntetni a periféria nevét).
- Doxygen kommentekkel kell minden függvényt ellátni
  - A minimális komment a következő
    - Egy soros **brief** leírás
    - Kifejtett paraméter komment a **param**-al
    - Kifejtett visszatérési érték komment a **ret** paraméterrel.
    - Kifejtett leírás a függvény működéséről.