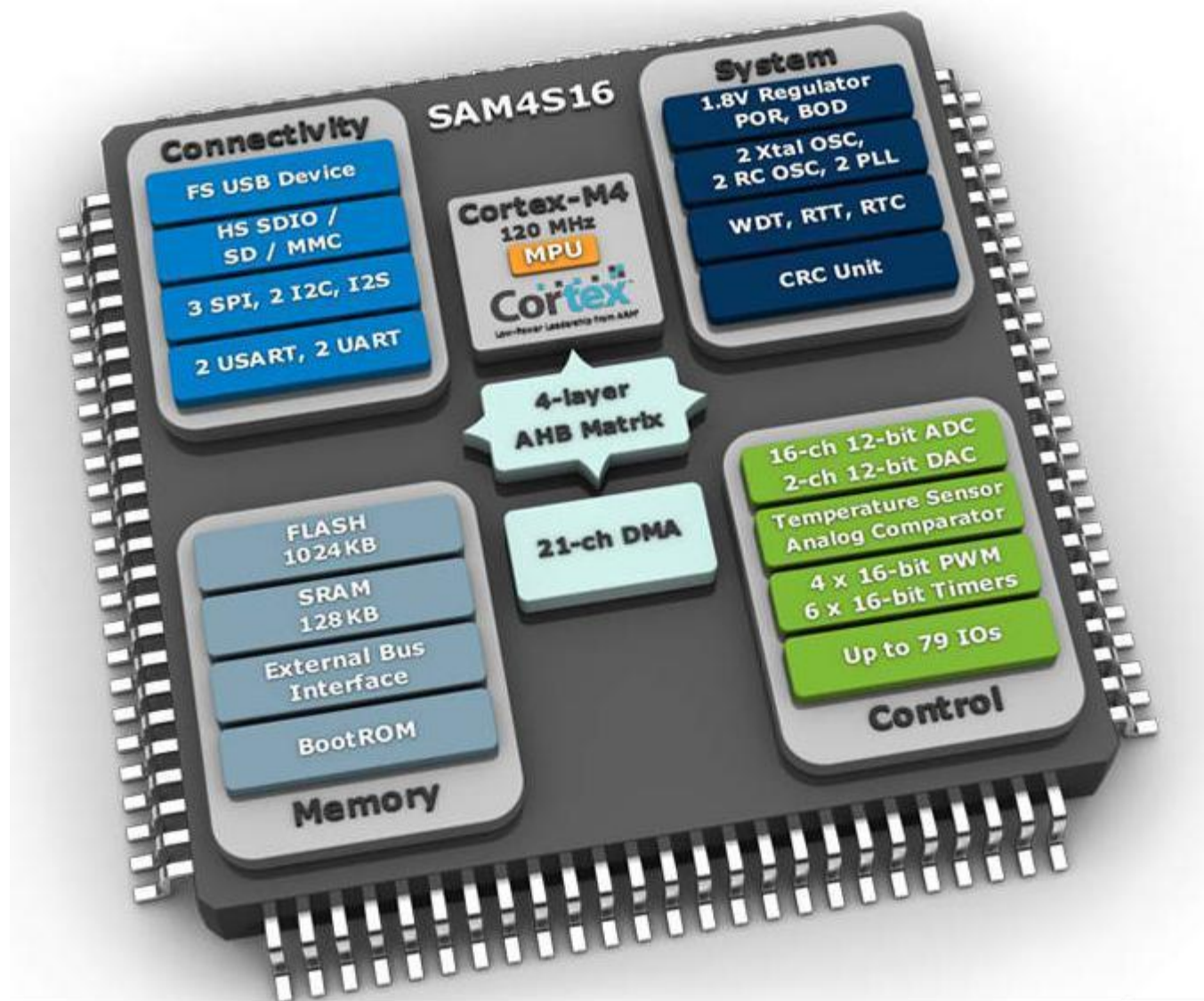




Nucleul ARM Cortex M

- I. Clasificarea.
- II. Blocul de regiștri.
- III. Harta de memorie.

I. Processore ARM Cortex - M



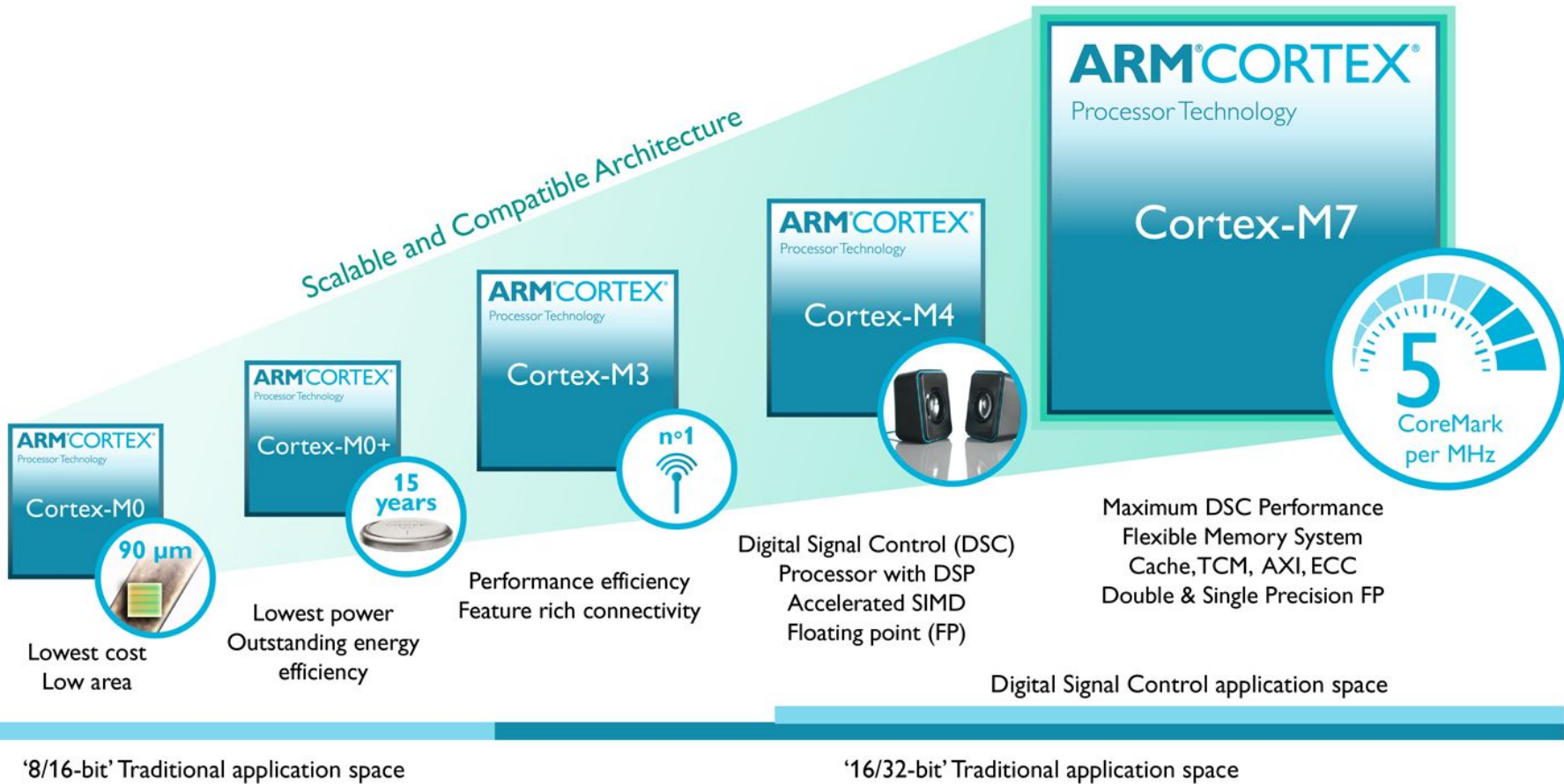
Procesorul Cortex-M face parte din familia de procesoare bazate pe arhitectura RISC ce este dezvoltată de *ARM Holdings*.

Indexul "M" specifică ca acest procesor se utilizează în microcontrolere.

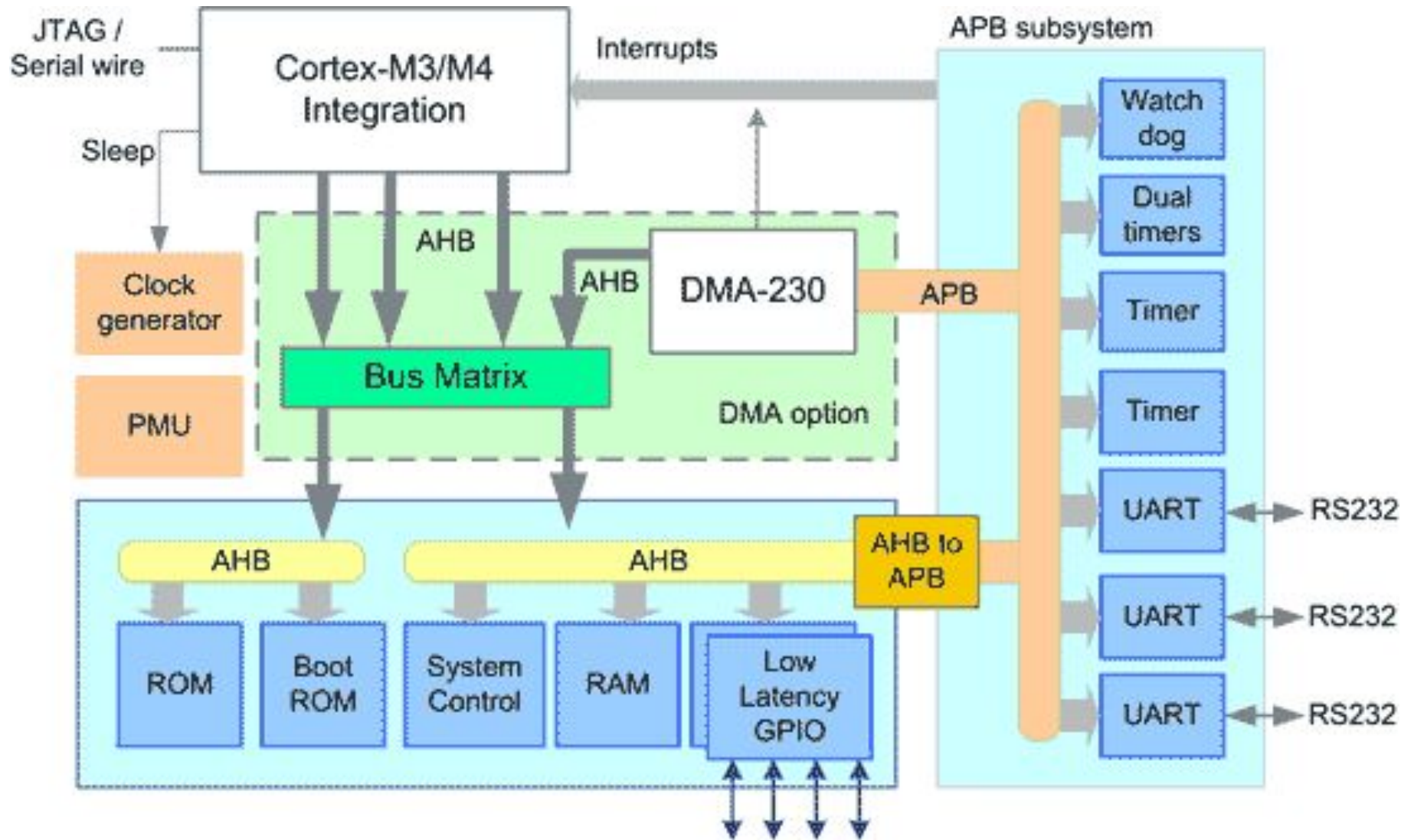
**La moment sunt nomenclatura procesoarelor Cortex-M include următoarele procesoare:
M0, M0+, M1, M3, M4, M7**

Toate procesoarele Cortex-M suportă setul de instrucțiuni *Thumb-2*

I. Structura nucleului ARM

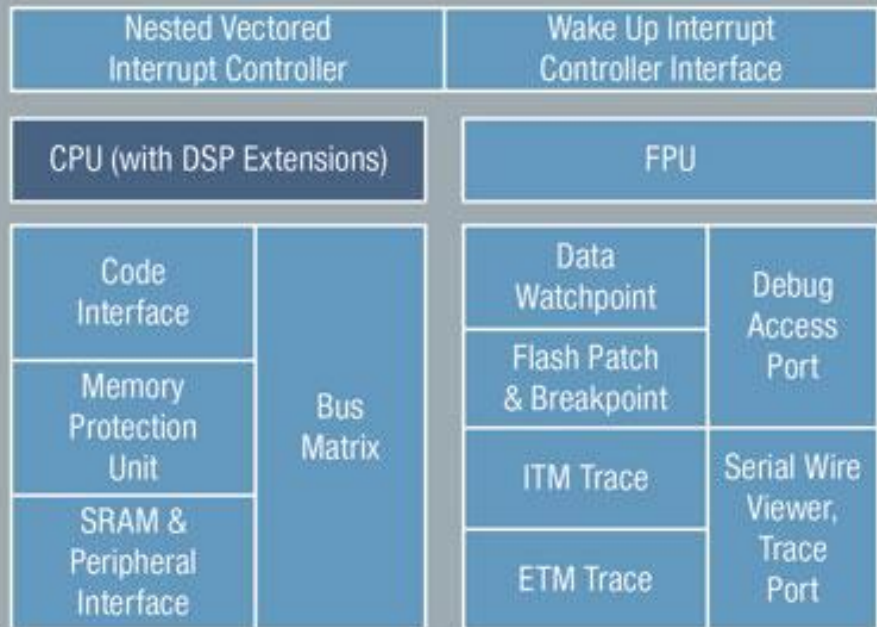


I. Structura nucleului ARM

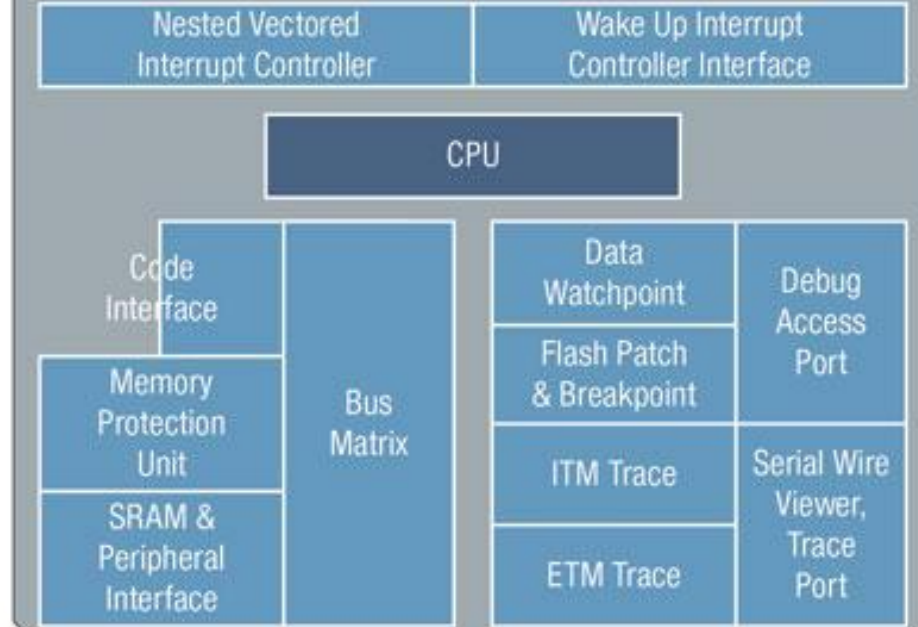


I. Structura nucleului ARM

Cortex-M4



Cortex-M3



ARM[®] Cortex[®]-M3

Nested Vectored
Interrupt Controller

Wake Up Interrupt
Controller Interface

CPU

Code
Interface

Memory
Protection
Unit

SRAM &
Peripheral
Interface

Bus
Matrix

Data
Watchpoint

Flash Patch
& Breakpoint

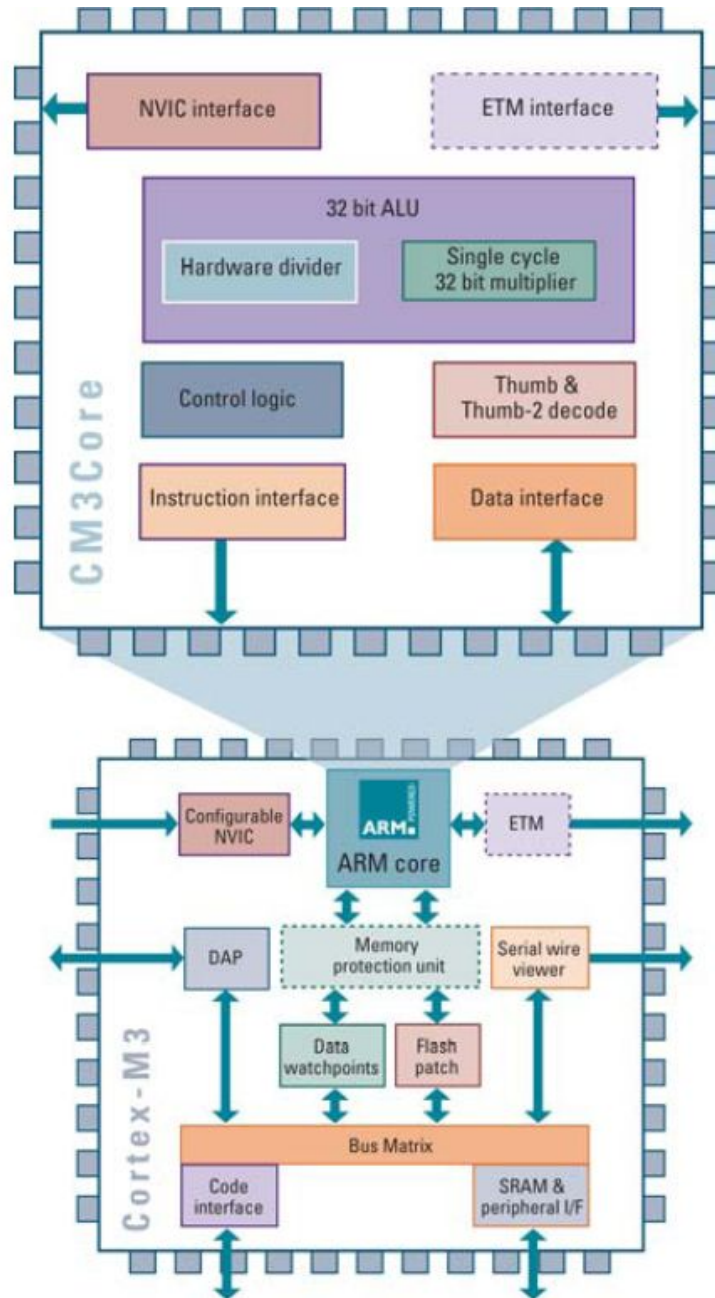
ITM Trace

ETM Trace

Debug
Access
Port

Serial
Wire
Viewer,
Trace Port

I. Structura microcontrolerului cu nucleul Cortex-M

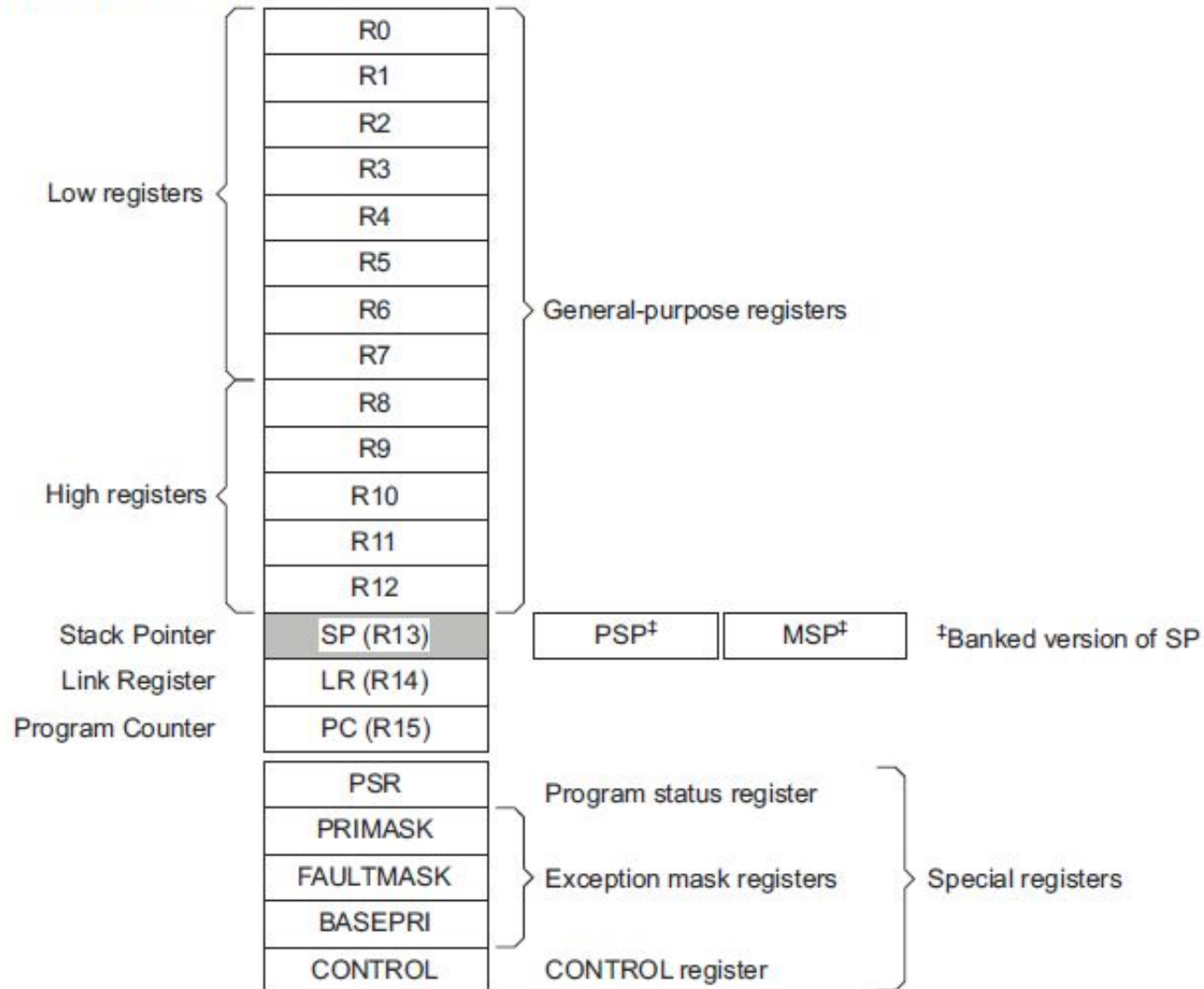


- **Arhitectura ARMv7-M**
- **Setul de instrucțiuni Thumb și Thumb-2**
- **Inmulțirea -1 cicluri, Împărțirea -2..12 cicluri**
- **Aritmetica cu saturație**
- **Memorie Bit Banding**
- **Pipeline cu 3 nivele**
- **Pîna la 240 vectori de intrerupere + NMI**
- **256 nivele de prioritate a întreruperilor**
- **Latența intrării in intrerupere 12 cicluri**

II. Blocul de regiștri

Core registers

The processor core registers are:

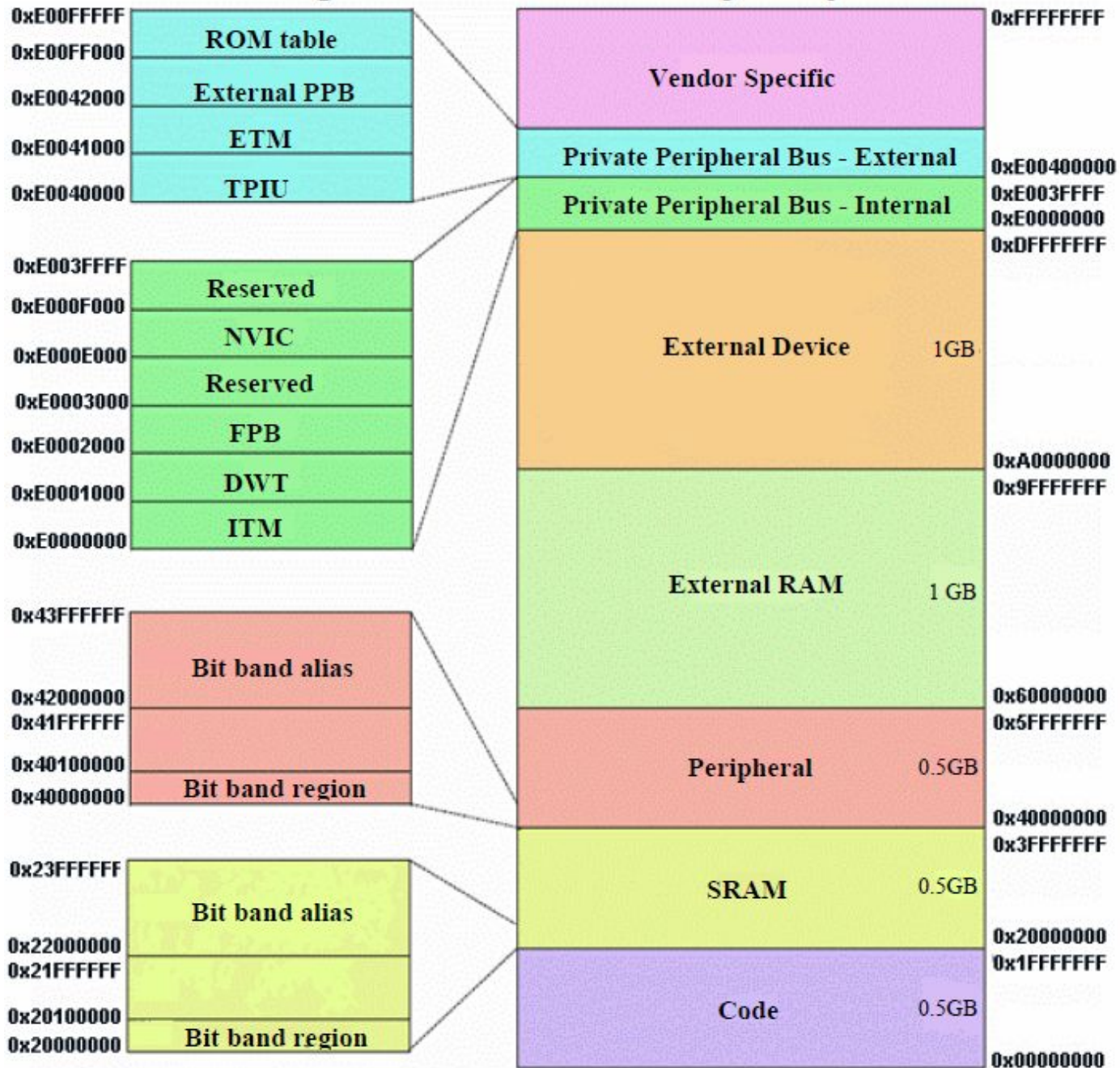


II. Blocul de regiștri

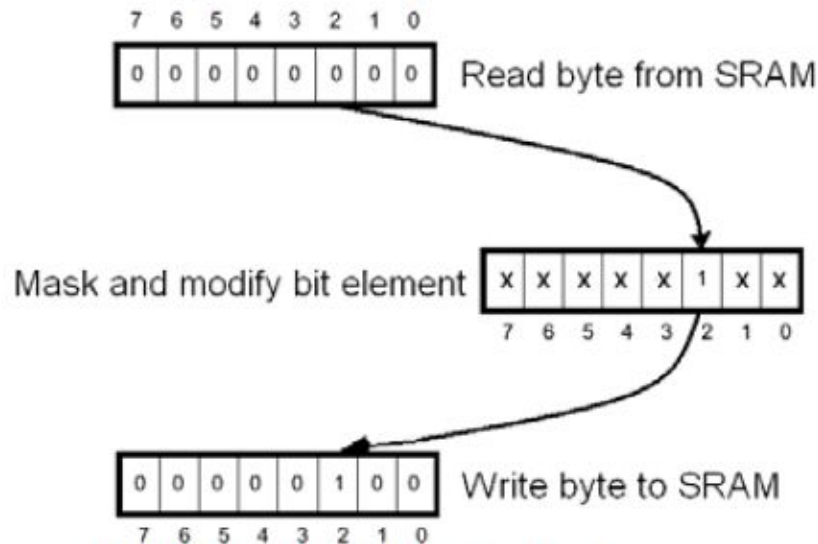
Name	Type ⁽¹⁾	Required privilege ⁽²⁾	Reset value	Description
R0-R12	RW	Either	Unknown	"General-purpose registers" on page 43
MSP	RW	Privileged	See description	"Stack Pointer" on page 43
PSP	RW	Either	Unknown	"Stack Pointer" on page 43
LR	RW	Either	0xFFFFFFFF	"Link Register" on page 43
PC	RW	Either	See description	"Program Counter" on page 43
PSR	RW	Privileged	0x01000000	"Program Status Register" on page 44
ASPR	RW	Either	0x00000000	"Application Program Status Register" on page 45
IPSR	RO	Privileged	0x00000000	"Interrupt Program Status Register" on page 46
EPSR	RO	Privileged	0x01000000	"Execution Program Status Register" on page 46
PRIMASK	RW	Privileged	0x00000000	"Priority Mask Register" on page 47
FAULTMASK	RW	Privileged	0x00000000	"Fault Mask Register" on page 48
BASEPRI	RW	Privileged	0x00000000	"Base Priority Mask Register" on page 49
CONTROL	RW	Privileged	0x00000000	"CONTROL register" on page 50

1. Describes access type during program execution in thread mode and Handler mode. Debug access can differ.
2. An entry of Either means privileged and unprivileged software can access the register.

III. Harta de Memorie

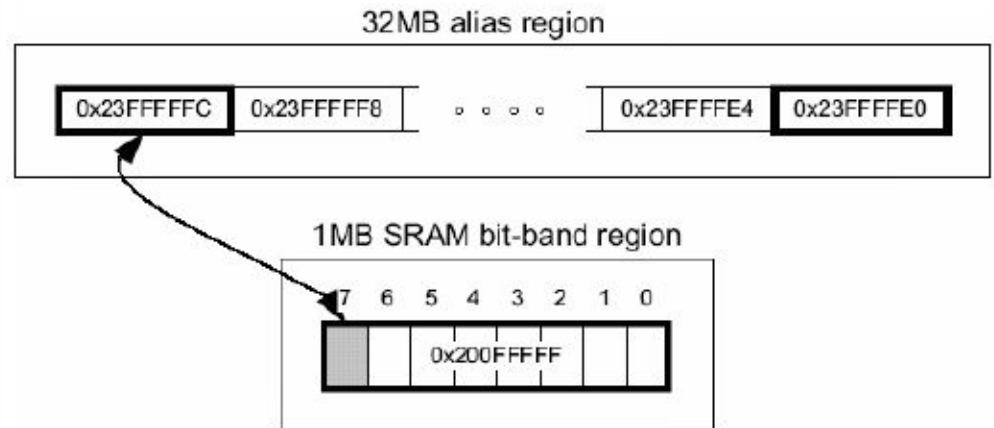


III. Harta de Memorie



```
LDR R0,=0x200FFFFF ; Setup address
MOV R2, #0x4       ; Setup data
LDR R1, [R0]       ; Read
ORR R1, R2         ; Modify bit
STR R1, [R0]       ; Write back result
```

Traditional bit manipulation method



```
LDR R0,=0x23FFFFFC ; Setup address
MOV R1, #0x1        ; Setup data
STR R1, [R0]        ; Write
```

Direct, single cycle access with bit banding

Control Subsystem

C28x™ 32-bit CPU
Up to 150 MHz

Floating-point unit

VCU

- Viterbi
- CRC
- Complex MPY
- FFT

Comms

- McBSP/SPV/PS
- UART

System

6-ch DMA

Control Modules

Up to 12× ePWM modules:
24× Outputs / 16× HR

Fault trip zones

6× 32-bit eCAP

3× 32-bit eQEP

Memory

256–512KB
ECC Flash

20KB ECC RAM

128-bit security

16KB parity RAM

64KB ROM

Shared

Analog Temp sense

12 bit, 12 ch, 2 SH, 3 MSPS
3-ch analog comparators

12 bit, 12 ch, 2 SH, 3 MSPS
3-ch analog comparators

Parity RAM

2-KB message

2-KB message

Up to 64 KB

Pwr & Clocking

- 10 MHz / 30 KHz INT OSC
- 4–20 MHz EXT
- Clock fail detect
- 3.3-V VREG
- POR/BOR

Host Subsystem

ARM® Cortex™ -M3
32-bit CPU
Up to 125 MHz

System & Clocking

32-ch DMA

4 Timers

2 Watchdogs

Memory

256–1024KB
ECC Flash

Up to 20KB ECC RAM

2× 128-bit security

16–112KB parity RAM

64-KB ROM

External interface

Communications

10/100 Ethernet MAC
1588 w/ MII

USB OTG FS PHY

4× SSI

5× UART

2× I²C

2× CAN